

DBSCAN-Based Dynamic Object Recognition and Semantic Information Entropy-Assisted Vehicle LiDAR Odometry

Shipeng Chen^{ID}, Xin Li^{ID}, Songhui Ma^{ID}, Shuo Wang^{ID}, and Xianrui Ren^{ID}

Abstract—In vehicle LiDAR odometry, a common challenge arises from the feature misalignment caused by dynamic objects. Although current popular deep-learning-based semantic segmentation methods are able to effectively identify dynamic objects, they are limited to recognizing pretrained types of dynamic objects and require substantial computational resources. Moreover, in the process of using semantic information to assist odometry feature matching, the uncertainty following point cloud semantic segmentation is not taken into account. These drawbacks lead to a lack of robustness in the backend optimization of semantic-information-assisted LiDAR odometry, especially in complex environments with various dynamic objects. Thus, we propose a semantic information-based density-based spatial clustering of applications with noise (SI-DBSCAN) for dynamic objects recognition and a semantic information entropy (SIE)-assisted vehicle LiDAR odometry to enhance the positioning accuracy of feature-based LiDAR odometry in dynamic environments. Specifically, we design an SI-DBSCAN model to recognize dynamic objects quickly and efficiently, and innovatively apply the concept of information entropy from probability theory to describe the uncertainty of the point cloud after semantic segmentation, as well as to use it as the weighting information of the point cloud feature matching to enhance the robustness of the LiDAR odometry. Experiments performed on the KITTI dataset show that our LiDAR odometry method, which integrates the SI-DBSCAN and SIE models, achieves a significant improvement in both dynamic objects' identification and odometry positioning accuracy compared with existing state-of-the-art methods.

Index Terms—Information entropy weight, LiDAR odometry, segmentation, semantic-information-based density-based spatial clustering of applications with noise (SI-DBSCAN).

I. INTRODUCTION

ODOMETRY plays a crucial role in vehicle navigation, especially in scenarios where the global navigation satellite system (GNSS) is denied. The primary objective

Received 10 March 2025; revised 21 April 2025; accepted 9 May 2025. Date of publication 19 May 2025; date of current version 3 June 2025. This work was supported in part by the Programs of the National Natural Science Foundation of China under Grant 42474026 and Grant 42127802 and in part by the Research on Key Technologies for Autonomous Drone Flight under Grant 2021-JCJQLB-071. The Associate Editor coordinating the review process was Dr. Weihua Li. (*Corresponding author: Xin Li*)

Shipeng Chen is with the Research Center of Space Control and Inertial Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: 2497410239@qq.com).

Xin Li, Shuo Wang, and Xianrui Ren are with the College of Geology Engineering and Geomatics, Chang'an University, Xi'an, Shaanxi 710054, China (e-mail: lixin2017@chd.edu.cn; 405277888@qq.com; blossom@chd.edu.cn).

Songhui Ma is with the Science and Technology on UAV Laboratory, Northwestern Polytechnical University, Xi'an, Shaanxi 710065, China (e-mail: masonghui@nwpu.edu.cn).

Digital Object Identifier 10.1109/TIM.2025.3571084

of odometry is to predict the motion of a carrier and accurately determine its current position [1]. Based on this, researchers have proposed a multitude of odometry algorithms, which are primarily categorized into visual odometry and LiDAR odometry. Compared with cameras, which are sensitive to lighting conditions and struggle to capture accurate depth information, LiDAR can obtain precise distance measurements and is less sensitive to changes in illumination. As a result, LiDAR odometry has been extensively studied and has demonstrated superior stability and accuracy.

LiDAR odometry is determined by analyzing the features of continuous LiDAR scans to locate the carrier [2], involving the application of matching between scan frames. LOAM, proposed by Zhang and Singh [3], is a classical LiDAR odometry that accomplishes scan matching by extracting edge and planar features through the calculation of points curvatures. However, relying solely on geometric features, such as curvature, to distinguish feature points can easily lead to points' mismatches, which prevent the construction of a globally consistent map and exacerbate the accumulation of odometry errors. These problems are particularly evident in complex environments with many dynamic objects, where mismatches may be caused by occlusions from these objects. LeGO-LOAM [4] is the first to propose using segmentation to extract feature points and uses clustering to reduce the interference of noisy points on feature registration. LIO-SAM [5], building on this, further enhances the accuracy and robustness of the odometry. Nevertheless, these classical SLAM methods often assume processing in static environments. In real dynamic environment, relying solely on geometric features such as smoothness cannot meet the execution of advanced commands, such as extracting edge points' information like "lamp posts/trees," significantly reducing the reliability of points matching. Evidently, there is a need for additional information to enhance the performance in such scenarios. Therefore, acquiring semantic label information from point cloud and converting it into semantic concepts of spatial features not only helps the carrier better understand the different feature information in the surrounding environment but also improves the vehicle's navigation capability in complex environments.

With the continuous development of image segmentation and deep learning, extracting semantic information from images has become increasingly mature. The application of

this semantic information in SLAM systems is also gaining more attention, such as using it to eliminate dynamic objects [6] and assist in pose estimation [7]. In recent years, the field of LiDAR point cloud semantic segmentation has achieved rapid development [8], [9], which has led to significant breakthroughs in the application of point cloud semantic information in SLAM systems. Particularly, significant progress has been made in integrating semantic information into pure LiDAR SLAM. Notably, some studies, such as SLOAM [10], focus on using semantic segmentation to design semantic-based LiDAR odometry for estimating things such as tree diameters. SUMA++ [11] uses semantic information to remove dynamic information and uses a surfel-based [12] approach to obtain a globally consistent semantic map. Moreover, SA-LOAM [13] uses semantic information to assist the odometry and loop closure modules, constructing globally consistent semantic map in large-scale environments. These studies highlight the tremendous potential of semantic information in enhancing the performance of LiDAR SLAM and have attracted numerous research efforts to incorporate it into comprehensive LiDAR SLAM solutions. In terms of identifying dynamic objects using semantic segmentation, an increasing number of studies are starting to use 3-D point cloud data from LiDAR scans as input for neural networks to generate specific detection results for identifying dynamic objects [14], [15]. Although methods based on semantic segmentation have the advantage of high accuracy through end-to-end learning, they can only recognize dynamic object types that have been pretrained, leaving other types of dynamic objects undetectable. Furthermore, the computational resource demands of deep learning methods lead to higher costs.

On the other hand, semantic information plays an important role in assisting with LiDAR odometry. Since different semantic categories carry different meanings, they are considered to have a distinctive role in accurate feature matching [13]. Therefore, it is very necessary to incorporate semantically related weights into each error term of feature registration. However, in current research, the weight information related to semantics is often set to equal values, which does not conform to the actual situation. Therefore, exploring a reasonable method for setting weights related to semantic information is particularly important.

To address the aforementioned problems, we focus on the problem of feature misalignment in complex environments with numerous dynamic objects for vehicle LiDAR odometry assisted by semantic information. Based on the density-based spatial clustering of applications with noise (DBSCAN) [16] and information entropy theory, it conducts related research, with the main tasks and contributions as follows.

- 1) A semantic information-assisted DBSCAN (SI-DBSCAN) empirical model for dynamic object recognition was proposed. It quickly identified dynamic object point cloud and effectively reduced point cloud mismatches caused by dynamic objects. Compared with current deep learning methods, this dynamic object identification model is unrestricted by the types of pretraining, offering broader applicability and flexibility.

- 2) We introduced a semantic information entropy (SIE) model that considered the uncertainty of semantic segmentation to assist in point cloud feature matching. The uncertainty of point cloud after semantic segmentation was statistically quantified from a probabilistic perspective and was used as weighting information for feature matching, effectively enhancing the robustness of LiDAR odometry. To our knowledge, this was the first scheme to describe the uncertainty of semantic segmentation using information entropy and to use it as weight information for point cloud feature matching.
- 3) We presented a vehicle LiDAR odometry aided by the SI-DBSCAN model for dynamic object identification and the SIE-weight. By conducting ablation experiments using the KITTI dataset, we validated the improvement of our proposed dynamic object recognition model and SIE weight on the accuracy of LiDAR odometry. Compared with the current state-of-the-art methods, our approach demonstrated similar or superior performance.

The remaining parts of this article are structured as follows. The related works' section presents an introduction to LiDAR odometry, dynamic object removal, and LiDAR odometry assisted by semantic information. The methodology section first outlines the overall workflow of the approach proposed in this article, followed by the semantic segmentation method, the SI-DBSCAN empirical model, and the mathematical models for point cloud registration and SIE; the experimental section focuses on the comparative analysis of the odometry accuracy under different schemes. Finally, some conclusions are drawn.

II. RELATED WORKS

A. LiDAR Odometry

To determine the position of the vehicle, the classic LOAM [3] algorithm achieves low-drift and real-time LiDAR odometry by minimizing the residual distances of point-to-line and point-to-plane during scan matching. Subsequently, researchers propose many LOAM-based improvement schemes. LeGO-LOAM [4] uses a point cloud clustering algorithm based on the breadth-first search. It raises the probability of detecting points associated with the same object between two consecutive scan frames. However, it is incapable of eliminating dynamic objects. DeLiO [17] was the first to introduce decoupled translation and rotation modules in LiDAR odometry, but it also does not deal with dynamic objects. Afterward, a weighted NDT [18] method addressed the point cloud matching problem by assigning larger or smaller weights to point with higher or lower probabilities. ISC-LOAM [19] leverages intensity information to promote effective position recognition. F-LOAM [20], compared with LOAM, determines the influence of feature matching point in nonlinear optimization based on the smoothness of feature point, thereby achieving higher accuracy than LOAM. R-LOAM [21] significantly improved the positioning accuracy of LiDAR odometry by introducing grid features. NHC-LIO [22] enhances the robustness of odometry by incorporating motion constraints. Ji et al. [23] propose a LiDAR odometry for dynamically assessing the reliability of point clouds. By

selecting key feature points that are sensitive to environmental degradation, the algorithm enhances its adaptability in diverse environments. These odometry schemes primarily enhance odometry performance using geometric feature information and additional auxiliary information.

B. Dynamic Object Removal

The presence of dynamic objects poses significant challenges to the accuracy and robustness of LiDAR SLAM algorithms, particularly in the front-end point cloud registration phase. In the map-building phase, dynamic objects can trigger the “ghost track” effect, which interferes with the recognition and extraction of static features. To address these challenges, SLAM frameworks typically adopt two strategies: filtering dynamic objects in real-time and mitigating their influence during the map construction phase. Among the online real-time filtering methods, RF-LIO [24] significantly improves the accuracy of position estimation in highly dynamic environments. This method prioritizes the removal of moving objects through adaptive multiresolution distance imagery, coupled with tightly integrated LiDAR-inertial odometry techniques. Subsequently, it performs LiDAR scanning and submap matching. In [25], a method combining the traditional dynamic point cloud filtering with deep-learning-based point cloud segmentation is proposed to enhance the accuracy of dynamic object rejection. However, this approach may fail to recognize dynamic objects of untrained categories. Although the influence of dynamic objects on point cloud alignment is relatively limited, the “ghost track” effect they produce can significantly disrupt map construction. To address this problem, [26] proposes a real-time dynamic point cloud filtering method for map construction. This method identifies and verifies dynamic points by comparing the point-to-point distance changes between two consecutive scans, thereby generating dynamic clusters. It requires only two reference scans and eliminates the need for local map construction. In the postprocessing stage of map construction, dynamic point cloud filtering methods primarily include ray tracing, visibility analysis, and segmentation techniques. In [27], a raster-based method is proposed to identify dynamic points by analyzing the hits and passes within the raster. In [28], the point-of-view visibility method is used to project query scans and submaps as depth maps, and dynamic points are identified through pixel depth comparison. In addition, ERASOR [29] effectively rejects dynamic points by dividing query scans and submaps into sector-like bins and comparing differences in point cloud distributions. Although these postprocessing strategies filter dynamic objects more accurately, they introduce a time delay because filtering occurs after SLAM map construction. In summary, processing dynamic objects is crucial in LiDAR SLAM algorithms. While existing methods provide varying tradeoffs between real-time performance and accuracy, further optimization of the efficiency and robustness of dynamic object filtering is still needed.

C. LiDAR Odometry Assisted by Semantic Information

Three-dimensional LiDAR point cloud data are mostly unstructured and unordered, yet they retain the original

geometric information of the space. Therefore, effectively extracting useful information from raw point cloud has become a hot research topic in recent years. PointNet [30], VoxelNet [15], and RangeNet++ [8] have adopted different approaches for the semantic segmentation of 3-D point cloud, which are point-based, voxel-based, and projection-based methods, respectively. With significant progress in semantic segmentation, many researchers have made substantial advancements in the application of point cloud semantic segmentation. In terms of identifying dynamic objects, Chen et al. [14] developed a dynamic object detection method based on a multiview network by using two subnetworks. Zhou and Tuzel [15] extract features from point cloud using a voxel coding method and use 3-D convolutional neural networks for end-to-end point cloud-based 3-D object detection. RI-Fusion [31] introduces an RI-Attention network designed to fuse depth image and RGB image features for the identification of dynamic small objects. Regarding assisting LiDAR odometry, S-LOAM [10] is designed for semantic LiDAR SLAM in forest scenes, using semantic information to detect point cloud belonging to trees and ground, achieving a simple and efficient LiDAR odometry. SUMA++ [11] integrates semantic information into a surfel-based map and proposes a method using semantic labels to filter dynamic objects, improving the accuracy of mapping and odometry. In PSF-LO [32], parametric semantic features are used to optimize geometric feature matching, and static and dynamic obstacles are classified at the instance level. SA-LOAM [13] introduces a method using semantic-assisted ICP, achieving significant improvements in accuracy. SSC [33] proposes an SLAM scheme that combines semantic information for effective encoding of 3-D scenes with a global descriptor, further enhancing accuracy.

At present, the research on semantic-segmentation-assisted LiDAR odometry primarily emphasizes integrating semantic information into the SLAM system by directly using semantic segmentation point information. However, it does not fully address the uncertainty of semantic segmentation results and its potential impact on feature matching accuracy. Consequently, incorporating semantically relevant weights into each error term during feature matching becomes crucial. Nonetheless, the specific definition of these semantically relevant weights is unclear, and there is a lack of an effective weight calculation method.

III. METHODOLOGY

A. Overview of the Proposed Approach

In this section, we introduce the overall process facilitated by the SI-DBSCAN empirical model and SIE together in assisting LiDAR odometry. Unlike the existing methods that integrate semantic segmentation into the SLAM system, we further explore other aspects of the SLAM system assisted by semantic information. This primarily includes the use of point cloud semantic information to aid the DBSCAN algorithm in identifying dynamic objects and using the SIE of point cloud as weight information for feature matching.

Fig. 1 shows the pipeline of our approach. Initially, the raw point cloud is input into the RangeNet++ [8] network to obtain a point cloud with semantic information. Then,

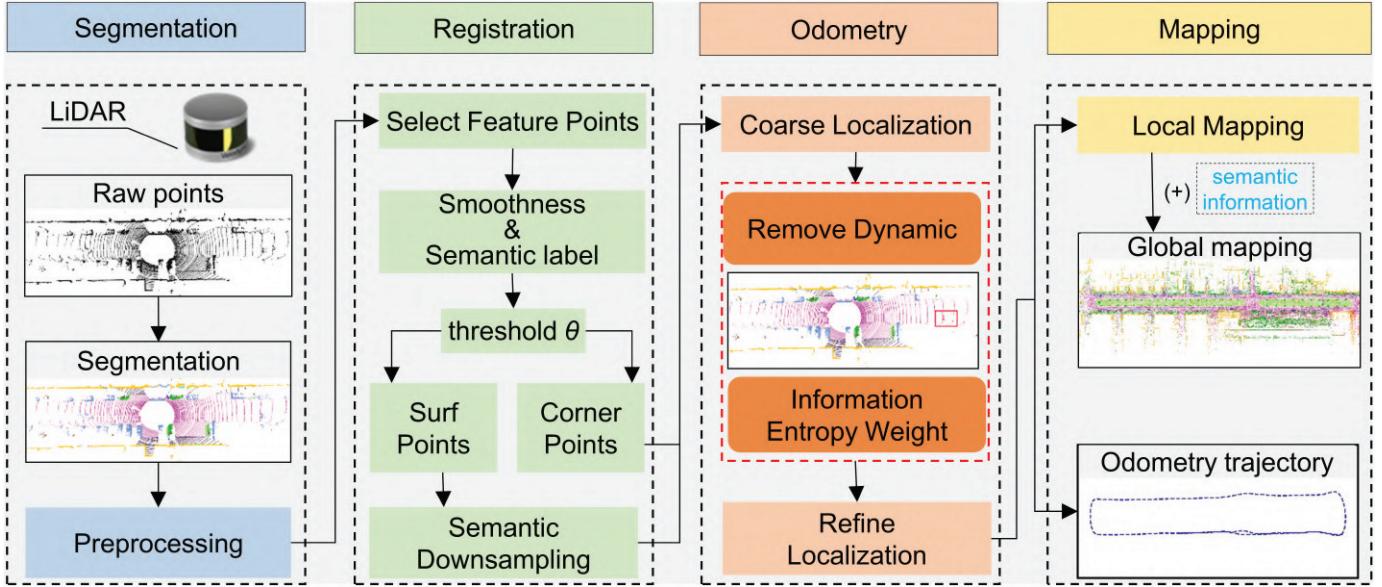


Fig. 1. Pipeline of the semantic-information-assisted LiDAR odometry.

the semantically enriched point cloud is fed into the point cloud matching module, and edge and planar feature points are extracted using semantic information and smoothness, completing the coarse localization. Subsequently, using the pose information between adjacent frames, dynamic objects are identified and eliminated using SI-DBSCAN. At the same time, the SIE of the feature point cloud is calculated as weight information for feature matching in refine localization. Finally, we can obtain a high-precision global semantic map.

B. Semantic-Information-Assisted LiDAR Odometry

Semantic segmentation is widely used to assist LiDAR odometry due to its ability to accurately identify objects, enhancing positioning and mapping capabilities. For instance, SA-LOAM [13] integrates semantic information into LiDAR odometry, effectively improving its robustness. In this section, we will introduce semantic segmentation and discuss the specific methods of using semantic segmentation information to identify dynamic objects and assist LiDAR odometry.

1) *Semantic Segmentation Network*: To achieve point cloud segmentation, we use a 2-D CNN based on projection methods to process the raw point cloud data [8], converting the point cloud from each LiDAR scan into distance images for inference.

Initially, the input point cloud is transformed into a range image representation using (1). Subsequently, a 2-D fully convolutional network is used for semantic segmentation, accomplishing the transfer of semantic information from 2-D to 3-D to reconstruct all the points of the original point cloud. Finally, an efficient 3-D postprocessing based on the range image is adopted, mainly to clean up individual discrete point in the cloud segmentation and infer artifacts

$$\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} [1 - \arctan(\mathbf{y}, \mathbf{x})\pi^{-1}] \mathbf{w} \\ [1 - (\arcsin z \mathbf{r}^{-1} + f_{\text{up}})f^{-1}] \mathbf{h} \end{pmatrix}. \quad (1)$$

The (\mathbf{u}, \mathbf{v}) represent the image coordinates, (\mathbf{h}, \mathbf{w}) are the height and width of the range image, and the values chosen in this article are (64, 2048). The $f = f_{\text{up}} + f_{\text{down}}$ represents the vertical field of view of the LiDAR. The $\mathbf{r} = \|\mathbf{P}_i\|_2$ is the distance value for each point $\mathbf{P}_i = (\mathbf{x}, \mathbf{y}, \mathbf{z})$.

The process results in a list of (\mathbf{u}, \mathbf{v}) tuples, each containing a pair of image coordinates corresponding to each point \mathbf{P}_i . Using these indices, we extract the distance, coordinates, and intensity information value for each point \mathbf{P}_i , storing them in an image to create a $[5 \times \mathbf{h} \times \mathbf{w}]$ tensor as input for the network. For the network's architecture, we have chosen an encoder-decoder hourglass structure, as shown in Fig. 2. Such deep hourglass-shaped segmentation networks are characterized by an encoder with significant downsampling, which not only allows the deep kernels to encode contextual information but also enables them to operate faster than networks without downsampling. Due to space constraints, for a comprehensive introduction to the network structure, we ask readers to refer to [8].

2) *SI-DBSCAN Model*: Traditional LiDAR SLAM algorithms are mostly based on the assumption of a static environment. However, in real-world scenarios, the presence of dynamic objects often disrupts map construction and localization accuracy. When the proportion of dynamic points in a frame is excessively high, it can significantly reduce odometry accuracy and potentially cause drift. Currently, methods for removing dynamic object point clouds mainly fall into two categories: one is based on cluster prediction for filtering dynamic obstacles [26], [34], and the other uses deep learning approaches for dynamic object recognition [25], [35]. The recognition of dynamic objects is based on the consistency of an object's location across frames. If an object remains in the same location across multiple consecutive frames, it is considered stationary; otherwise, if its position changes, it is classified as dynamic.

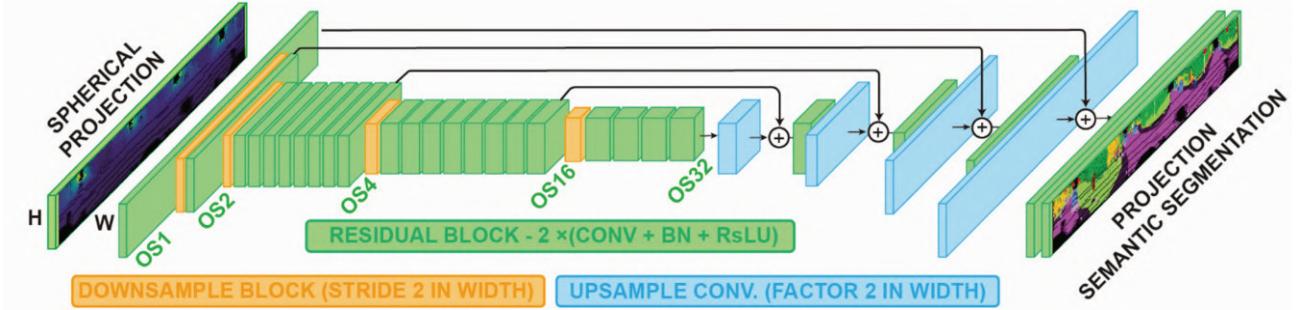


Fig. 2. Fully convolutional semantic segmentation architecture [8].

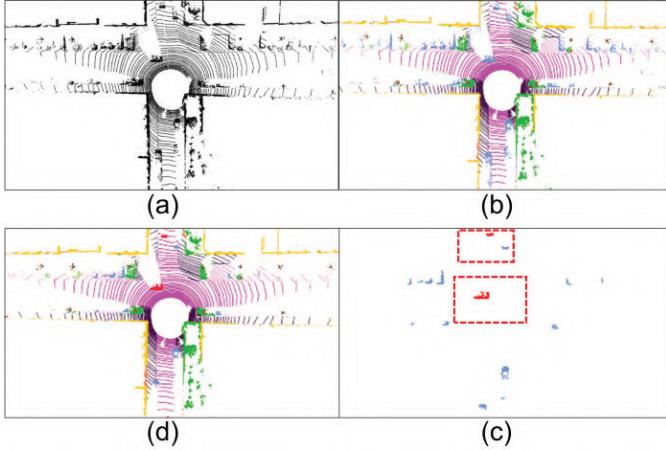


Fig. 3. Process of the SI-DBSCAN model. (a) Raw point cloud. (b) Point cloud after semantic segmentation. (c) Point cloud of dynamic objects identified by SI-DBSCAN clustering. (d) Integration of dynamic object point cloud into the point cloud after semantic segmentation.

To accurately identify and filter out dynamic objects, we propose the SI-DBSCAN model for object clustering and motion state estimation in highly dynamic environments. The method begins by performing semantic segmentation on the LiDAR point cloud, assigning labels to each point to extract all potential dynamic object point clouds, which include both static and dynamic points. Subsequently, the object point cloud is clustered using the DBSCAN algorithm, where each object is independently grouped into a cluster, and its centroid is calculated. Then, using position information from the LiDAR odometry, a velocity estimation method based on the Euclidean distance projects the vehicle centroid from the current frame into the coordinate system of the previous frame and identifies the nearest point to the projected centroid in the previous frame. If the Euclidean distance between the corresponding centroids in two consecutive frames exceeds a predefined threshold, the vehicle is classified as moving, and the results are shown in Fig. 3.

DBSCAN is a density-based unsupervised machine learning algorithm that can discover clusters of arbitrary shapes in spatial databases with noise. It achieves this by dividing regions with sufficient density, effectively identifying closely connected samples and forming meaningful clusters. We describe the density distribution of samples with parameters $(\epsilon, \text{MinPts})$, where ϵ represents the neighborhood distance

threshold, and MinPts represents the threshold for the number of samples in the neighborhood. For a frame of point cloud $P = p_1, p_2, \dots, p_m$, its density is defined as follows.

- 1) **ϵ -Neighborhood:** For $p_i \in P$, its ϵ -neighborhood contains a subset of samples from P whose distance to p_i is no greater than ϵ , i.e., $N_\epsilon(p_i) = \{p_j \in P \mid \text{distance}(p_i, p_j) \leq \epsilon\}$. The number of samples in this subset is denoted as $|N_\epsilon(p_i)|$.
- 2) **Core Object:** For any sample $p_i \in P$, if its ϵ -neighborhood $N_\epsilon(p_i)$ contains at least MinPts samples, that is, if $|N_\epsilon(p_i)| \geq \text{MinPts}$, then p_i is a core object.
- 3) **Directly Density-Reachable:** For p_i and p_j in the ϵ -neighborhood, and if p_j is a core object, then p_i is density-reachable from p_j .
- 4) **Density-Reachable:** For p_i and p_j , if there exists a sequence of samples n_1, n_2, \dots, n_m , such that $n_1 = p_i$, $n_m = p_j$, and each n_{t+1} is directly density-reachable from n_t , then p_i is density-reachable from p_j .
- 5) **Density-Connected:** For p_i and p_j , if there exists a core object sample p_k such that both p_i and p_j are density-reachable from p_k , then p_i and p_j are density-connected.

In terms of algorithm implementation, as described in Algorithm 1, the algorithm first initializes the potentially dynamic point cloud set $P = [p_1, p_2, \dots, p_m]$ after semantic segmentation, marking all the points $p_i \in P$ as unvisited ($\text{visited}(p_i) = 0$), and sets the parameters ϵ and MinPts . In this article, the thresholds for these two parameters are set to 0.55 and 50, respectively. Next, the DBSCAN clustering algorithm is applied to the point cloud: a random unvisited point p_j is selected and marked as visited ($\text{visited}(p_j) = 1$), and its ϵ -neighborhood $N_\epsilon(p_j) = \{p_i \in P \mid \text{distance}(p_j, p_i) \leq \epsilon\}$ is computed. If the number of points in the neighborhood satisfies $|N_\epsilon(p_j)| \geq \text{MinPts}$, the point is marked as a core point, and a new cluster C_k is created. The algorithm then iterates through the neighborhood points $p' \in N_\epsilon(p_j)$ to expand the cluster. If the number of points in the neighborhood is insufficient, the point is marked as noise. The clustering process terminates when all the points have been visited. Subsequently, the centroid coordinates $p_{C_k} = (1/n) \sum_{i=1}^n p_i$ (where $p_i \in C_k$) of each cluster C_k are calculated, and the distance d_{ij} between clusters in adjacent frames is computed using (2). Finally, dynamic object recognition is performed on the clusters using (3), assigning labels to the point cloud and outputting the labeled point cloud set S_{label} . If the

Algorithm 1 SI-DBSCAN Model**Input:** $P = [p_1, p_2, \dots, p_m]$, ϵ , $MinPts$ **Output:** P with static and dynamic S_{label} after clustering.**Initialization:** $visited(p_i) \leftarrow 0, \forall p_i \in P$ **1. DBSCAN performs clustering on point clouds.****while** $\exists p_i \in P, visited(p_i) = 0$ **do** $p_j \leftarrow \text{Random}\{P \mid \text{visited}(p_j) = 0\}$ $visited(p_j) \leftarrow 1$ $N_\epsilon(p_j) \leftarrow \{p_i \in P \mid \text{distance}(p_j, p_i) \leq \epsilon\}$ **if** $|N_\epsilon(p_j)| \geq MinPts$ **then** $C_k \leftarrow \{p_j\}$ where $k = 1, 2, \dots, n$ $N \leftarrow N_\epsilon(p_j)$ **for** $p' \in N$ **do** **if** $\text{visited}(p') = 0$ **then** $\text{visited}(p') \leftarrow 1$ $N' \leftarrow N_\epsilon(p')$ **if** $|N'| \geq MinPts$ **then** $N \leftarrow N \cup N'$ **end if** **if** $p' \notin C_k$ **then** $C_k \leftarrow C_k \cup \{p'\}$ **end if** **end if** **end for** **end if****end while****2. Calculate the centroid coordinates of each cluster.**

$$p_{C_k} = \frac{1}{n} \sum_{i=1}^n p_i, p_i \in C_k$$

3. Calculate adjacent-frame cluster distances.

$$d_{ij} \leftarrow (2)$$

4. Perform dynamic object recognition.

$$label \leftarrow (3)$$

$$P \leftarrow label$$

Output: S_{label}

corresponding distance difference d_{ij} is less than the threshold α , it is classified as a static object and the labels of all the point clouds in its corresponding cluster are set to 1; conversely, if the d_{ij} is greater than the threshold α , it is identified as a dynamic object and the labels of all the point clouds in its corresponding cluster are set to -1. In the subsequent feature extraction phase, all the dynamic objects labeled as -1 will be eliminated to reduce interference with localization and mapping. Furthermore, the setting of the threshold α is not fixed, and we believe that fine-tuning according to the actual situation will bring certain improvements. In addition, for specific situations such as when background point are obscured by foreground point, or vice versa, we use (4) to address these problem

$$d_{ij} = \sqrt{(p_j - p_i)_x^2 + (p_j - p_i)_y^2 + (p_j - p_i)_z^2} \quad (2)$$

where d_{ij} represents the difference in the distance between the center positions of corresponding objects across two adjacent

frames

$$S_{label} = \begin{cases} 1, & \text{if } d_{ij} \leq \alpha \\ -1, & \text{if } d_{ij} > \alpha \end{cases} \quad (3)$$

where S_{label} represents the label information of the point cloud, where 1 indicates the label for static objects, and -1 is the label for dynamic objects; α represents the distance threshold

$$\begin{cases} d = |p_x|^2 + |p_y|^2 + |p_z|^2 \\ \phi = \frac{(\arctan \theta_{(x,y,z)} + \pi)}{2\pi} \end{cases} \quad (4)$$

where d represents the distance to the LiDAR, and x, y, z represent the coordinates of the scanned point cloud. ϕ represents the angle in the vertical direction. If the distance corresponding to the same ϕ is less than d , then the current point's label is marked as the label of a background point, with the label of background point being determined by the labels of the surrounding point cloud.

3) *SIE-Assisted Feature Registration For Lidar Odometry:* We adopt the feature extraction strategy from LOAM [3], which involves extracting edge and planar features from the LiDAR points by calculating the curvature of each point. The specific formula for calculating curvature is shown in the following equation:

$$c = \frac{1}{n \cdot \|p_i\|} \left\| \sum_{j=1, j \neq i}^n (p_j - p_i) \right\| \quad (5)$$

where p_i represents the object point. p_j represents the point within the same beam. Then, by setting a threshold β , these points are classified into edge points and planar point. Typically, points with curvature greater than β are classified as edge points, and those with curvature less than β are classified as planar points.

After acquiring edge points and planar points, to determine the relationship between consecutive LiDAR frames, we first define two consecutive LiDAR point clouds as P_k and P_{k+1} , with their corresponding timestamps being t_k and t_{k+1} . Given that points are not acquired instantaneously, it is necessary to project the P_k point to the time of t_{k+1} to explore the connection between these two frames. This results in the projected point cloud P_k . Next, we search for the corresponding edge lines and planes in P_k for the edge points and planar points of the P_{k+1} frame. For a detailed description of this process, refer [3].

To find the corresponding relationship between two consecutive LiDAR points, we use the distance from point to line and the distance from point to plane for matching. Here, we only present the formulas for the distance from point to line and from point to plane, as shown in the following equation and (7)

$$d_{e_{k+1}} = \frac{|(P_{i+1,k+1}^e - P_{i,u}^e) \times (P_{i+1,k+1}^e - P_{i,v}^e)|}{|P_{i,u}^e - P_{i,v}^e|} \quad (6)$$

where $P_{i+1,k+1}^e$ denotes the $(i+1)$ th edge point in the $(k+1)$ th frame; $P_{i,u}^e$ and $P_{i,v}^e$ represent two distinct points on the same

line found in the k th frame. The term $d_{e_{k+1}}$ is the distance from the object point to the line

$$d_{p_{k+1}} = \frac{\left| (\mathbf{P}_{i+1,k+1}^p - \mathbf{P}_{i,u}^p) \right|}{\left| (\mathbf{P}_{i,u}^p - \mathbf{P}_{i,v}^p) \times (\mathbf{P}_{i,u}^p - \mathbf{P}_{i,w}^p) \right|} \quad (7)$$

where $\mathbf{P}_{i+1,k+1}^p$ denotes the $(i+1)$ th edge point in the $(k+1)$ th frame. $\mathbf{P}_{i,u}^p$, $\mathbf{P}_{i,v}^p$, and $\mathbf{P}_{i,w}^p$ represent three different points on the same plane found in the k th frame. $d_{p_{k+1}}$ is the distance from the object point to the planar.

To estimate the motion relationship between two frames of LiDAR points, we minimize the distance from edge points to their corresponding lines and the distance from planar point to their corresponding planes. Since we use semantic information to assist in the selection of feature point, and different point categories contribute differently to feature matching, we assign a semantic-related weight w^l to each error function. The overall optimization objective function is presented in the following equation:

$$r = \sum_l \left\{ \sum_i (w^l d_{ei}^l) + \sum_j (w^l d_{pj}^l) \right\} \quad (8)$$

where d_{ei}^l represents the distance from a point to line, and d_{pj}^l is the distance from a point to plane.

The point cloud features extracted based on semantic information are inherently obtained through network prediction and naturally carry a certain degree of uncertainty. To describe the uncertainty of point after semantic segmentation, we innovatively introduce the concept of information entropy from probability theory. Information entropy is a common metric to measure the uncertainty or complexity of information. It can assess the impact of variables or features on the entire information system. Typically, the higher the uncertainty of information, the greater its information entropy, and vice versa. In point feature matching, the geometric structural feature point usually contributes more significantly, even though their information entropy is lower. Therefore, we have performed a certain transformation on information entropy, that is, the greater the uncertainty, the lower the numerical value of information entropy; the smaller the uncertainty, the higher the numerical value of information entropy. Subsequently, we normalized the numerical values of information entropy to serve as the weight information for point feature matching. The specific formula for information entropy is as shown in the following equation:

$$w^l = - \sum_{p_i \in P} P(p_i) \log P(p_i) \quad (9)$$

where w^l represents the information entropy, and $P(p_i)$ is the probability that the feature point belongs to the i th category. The probability value p_i represents the likelihood that a point in the point cloud belongs to the i th category following semantic segmentation.

Finally, we use the following equation to obtain the relative transformation $\Delta T_{i,i+1}$ between the states X_i and X_{i+1} , thus acquiring the LiDAR odometry for these two poses

$$\Delta T = T_i^T T_{i+1}. \quad (10)$$

IV. EXPERIMENTAL STUDIES

In this section, we designed a series of ablation experiments aimed at qualitatively and quantitatively demonstrating the effectiveness of the SI-DBSCAN model in eliminating dynamic objects, such as vehicles, and the enhancement of LiDAR odometry performance using SIE as weight information for feature matching. First, we verified and analyzed the dynamic object segmentation capability of the SI-DBSCAN model. Subsequently, we presented the results of using SIE weight to assist in odometry estimation. Finally, the effectiveness of the SI-DBSCAN model and SIE-assisted feature matching in improving LiDAR odometry accuracy was verified.

A. Dataset and Implementation Details

We evaluate the proposed method using the SemanticKITTI dataset and the KITTI Odometry Benchmark [36], [37]. The SemanticKITTI dataset provides pointwise annotation information, where points annotated with selected classes (252, 253, 254, 255, 256, 257, and 259) are marked as ground-truth dynamic points for removal. The KITTI Odometry Benchmark includes 11 sequences with ground-truth trajectories, providing a reliable reference for evaluation. We selected the open-source A-LOAM project as our baseline method. For semantic segmentation, we used a pretrained model of RangeNet++[8], trained on the SemanticKITTI dataset and implemented in PyTorch [38], resulting in 19 semantic categories. Experiments were conducted on a computer equipped with an Intel i7 CPU, an RTX 3070 Ti GPU, and 16 GB of memory.

B. SI-DBSCAN Model Identifying Dynamic Objects

To evaluate the effectiveness of the SI-DBSCAN model in removing dynamic objects, we selected five sequences with the highest number of dynamic object occurrences from the Semantic KITTI dataset (00, 01, 02, 05, and 07) and assessed them both qualitatively and quantitatively. First, we qualitatively evaluated the SI-DBSCAN model's accuracy in identifying dynamic objects by comparing its performance with other state-of-the-art methods, such as ERASOR and Removert, through visualization. Fig. 4 presents the results of using the SI-DBSCAN method to generate static maps compared with other state-of-the-art methods in sequences 05 and 07. The red point clouds in the figure indicate dynamic objects, with fewer red points corresponding to better removal of dynamic objects. The results show that the existing state-of-the-art methods can filter out most dynamic point clouds while preserving static point clouds. However, our proposed SI-DBSCAN method demonstrates significant advantages over ray-tracing and visibility-based methods. It not only completely preserves the static point cloud but also nearly completely filters out the dynamic

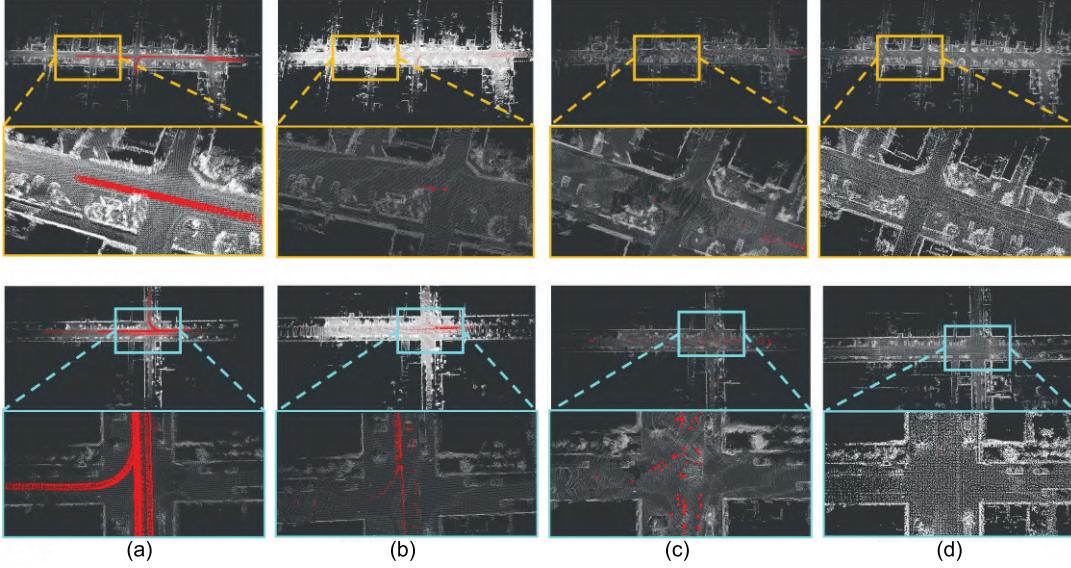


Fig. 4. Comparison of static map generation results between the proposed method and state-of-the-art techniques on SemanticKITTI: sequences 05 and 07. Fewer red points (dynamic objects) indicate better performance. (a) Original map. (b) Removert. (c) ERSOR. (d) SI-DBSCAN.

TABLE I
COMPARISON TO STATE-OF-THE-ART METHODS ON THE SEMANTICKITTI DATASET. PR: PRESERVATION RATE, RR: REJECTION RATE, F1 SCORE

Seq	Method	PR[%]	RR[%]	F1 score
00	Removert [29]	86.829	90.617	0.887
	ERSOR [29]	93.980	97.081	0.955
	SI-DBSCAN	97.379	99.594	0.984
01	Removert [29]	95.815	57.077	0.715
	ERSOR [29]	91.487	95.383	0.934
	SI-DBSCAN	99.926	96.444	0.981
02	Removert [29]	83.293	88.371	0.858
	ERSOR [29]	87.731	97.008	0.921
	SI-DBSCAN	99.243	99.776	0.995
05	Removert [29]	88.170	79.981	0.839
	ERSOR [29]	88.730	98.262	0.933
	SI-DBSCAN	99.127	98.907	0.990
07	Removert [29]	82.038	95.504	0.883
	ERSOR [29]	90.624	99.271	0.948
	SI-DBSCAN	97.958	97.613	0.977

point cloud. This is attributed to the SI-DBSCAN method's reliance on semantic segmentation, which reclusters unclassified and potentially dynamic objects for identification, effectively removing dynamic objects even in areas where visibility is compromised. Subsequently, the SI-DBSCAN method was quantitatively compared with other open-source state-of-the-art methods using three metrics: rejection rate (PR), preservation rate (RR), and *F1* score, as shown in (11)–(13). Table I lists the PR, RR, and *F1* score for sequences 00, 01, 02, 05, and 07. The results indicate that our method achieves optimal performance on most sequences. This is primarily attributed to the semantic segmentation preprocessing step, which restricts dynamic object recognition to point clouds of potential dynamic objects. This approach reduces the number of point cloud clusters and avoids processing large quantities

TABLE II
DATA PROCESSING STRATEGY OF THREE DIFFERENT SCHEMES

	SI-DBSCAN	the weight of SIE
Ours-SI-DBSCAN	✓	✗
Ours-SIE	✗	✓
Ours-ODOM	✓	✓

of irrelevant points, thereby enhancing the accuracy of dynamic object recognition

$$PR = \frac{\text{of preserved static points}}{\text{of total static points on the raw map}} \quad (11)$$

$$RR = 1 - \frac{\text{of preserved dynamic points}}{\text{of total dynamic points on the raw map}} \quad (12)$$

$$F1 \text{ score} = \frac{2 * PR * RR}{PR + RR}. \quad (13)$$

C. LiDAR Odometry Ablation Experiment

To validate the impact of the SI-DBSCAN model and SIE weight on LiDAR odometry, we designed ablation experiments to be tested on the KITTI dataset. For semantic segmentation of point clouds, we used the RangeNet++ network model, adopting an offline training-online inference mode to ensure the timeliness of segmentation. Since our aim is to verify the performance of the proposed method, we only selected the KITTI 00–10 sequences as the test dataset. The specific scheme is illustrated in Table II, where the approach of removing dynamic objects and assigning equal weights is referred to as Ours-SI-DBSCAN; the approach using SIE as the weight information is referred to as Ours-SIE; and the approach of removing dynamic objects and using information entropy as weight information is called Ours-ODOM. In addition, our designed scheme is compared with several advanced pure

TABLE III
RPE ON KITTI ODOMETRY

Seq	A-LOAM	FLOAM	ISC-LOAM	SA-LOAM*	TRLO*	Ours-SI-DBSCAN	Ours-SIE	Ours-ODOM
00	0.29/0.76	0.28/0.65	0.53/1.33	0.25/0.59	-/1.15	0.26/0.59	0.24/0.56	0.25/0.56
01	0.49/1.97	0.50/1.97	0.57/2.17	0.48/1.89	-/-	0.50/2.00	0.49/1.90	0.49/1.96
02	0.35/0.94	0.36/1.05	0.54/1.55	0.28/0.77	-/-	0.37/1.06	0.34/0.93	0.37/0.97
03	0.48/0.96	0.47/0.96	0.59/1.29	0.46/0.87	-/-	0.47/0.92	0.45/0.85	0.45/0.88
04	0.39/0.72	0.39/0.69	0.39/0.67	0.35/0.59	-/-	0.39/0.72	0.38/0.67	0.38/0.68
05	0.24/0.54	0.29/0.52	0.43/0.96	0.24/0.45	-/1.15	0.23/0.44	0.24/0.46	0.24/0.44
06	0.28/0.60	0.27/0.55	0.46/1.23	0.25/0.52	-/-	0.27/0.60	0.27/0.52	0.26/0.56
07	0.25/0.43	0.23/0.44	0.56/1.23	0.22/0.41	-/-	0.21/0.39	0.22/0.40	0.21/0.39
08	0.32/1.04	0.32/0.95	0.48/1.35	0.27/0.85	-/1.10	0.26/0.83	0.28/0.85	0.26/0.85
09	0.30/0.73	0.29/0.68	0.51/1.25	0.28/0.68	-/-	0.28/0.67	0.27/0.66	0.27/0.66
10	0.39/0.99	0.40/1.00	0.60/1.52	0.35/0.76	-/-	0.42/1.02	0.37/0.73	0.39/0.86

Mean relative pose error over trajectories of 100 to 800 m: relative rotational error in degrees per 100 m / relative translational error in%. The results of SA-LOAM and TRLO marked with * is from the original paper [13] [39], Specifically, within the SA-LOAM results, we have chosen the Ours-ODOM data as reported in the original document. - is due to the method's failure to process the trajectory correctly.

TABLE IV
ATE ON KITTI ODOMETRY: MEAN(M)/RMSE(M)

Seq	A-LOAM	FLOAM	ISC-LOAM	SA-LOAM*	TRLO*	Ours-SI-DBSCAN	Ours-SIE	Ours-ODOM
00	2.6/3.13	4.04/4.76	4.06/4.88	-/5.14	-/3.29	1.89/2.30	2.10/2.54	2.09/2.56
01	16.89/18.78	17.12/19.04	17.01/18.92	-	-	17.04/18.98	16.63/18.52	16.86/18.77
02	3.09/3.61	2.94/3.30	3.03/3.56	-	-	2.88/3.27	2.90/3.24	2.91/3.29
03	0.80/0.86	0.85/0.92	0.90/0.96	-	-	0.78/0.85	0.80/0.86	0.79/0.85
04	0.36/0.40	0.31/0.34	0.19/0.21	-	-	0.38/0.42	0.35/0.39	0.37/0.41
05	2.28/2.75	2.92/3.41	2.54/2.96	-/3.04	-/2.36	1.91/2.23	1.79/2.09	1.49/1.71
06	0.66/0.72	0.70/0.75	0.69/0.74	-/0.69	-	0.64/0.70	0.64/0.69	0.62/0.68
07	0.63/0.65	0.54/0.58	0.57/0.62	-/0.53	-/1.10	0.48/0.51	0.49/0.53	0.46/0.50
08	3.27/3.72	3.58/4.08	4.25/4.88	-/3.61	-	3.16/3.79	3.15/3.58	3.14/3.58
09	1.32/1.49	1.24/1.46	1.53/1.74	-/1.74	-	1.26/1.43	1.19/1.38	1.10/1.24
10	1.19/1.36	1.20/1.39	1.21/1.40	-	-	1.24/1.42	1.10/1.26	1.17/1.34

The results of SA-LOAM and TRLO marked with * is from the original paper [13] [39]. Specifically, within the SA-LOAM results, we have chosen the Ours-ODOM data as reported in the original document.- means there are no results in the original paper.

LiDAR-based SLAM methods, such as LOAM, FLOAM, ISC-LOAM, SA-LOAM, and TRLO [39].

1) *Validate the Impact of Dynamic Objects on LiDAR Odometry:* In point cloud mapping that contains dynamic objects, the “ghosts” effect caused by these objects can degrade localization accuracy, particularly in scenes with a high density of dynamic objects. In the SI-DBSCAN model identifying dynamic objects’ section, we demonstrated the reliability of the SI-DBSCAN model for dynamic object detection. To further evaluate the robustness of LiDAR odometry after dynamic object removal, we selected sequence 07 from the KITTI dataset, which contains numerous dynamic objects, and performed both qualitative and quantitative analyses.

Fig. 5 shows the local details of mapping in the KITTI dataset sequence 07 with and without dynamic objects removed. Here, we present the local mapping for Our-SI-DBSCAN and A-LOAM. The local mapping for FLOAM and ISC-LOAM is not shown, as they are similar to A-LOAM and do not eliminate dynamic objects, resulting in “ghosting” effects. Through comparison, using the SI-DBSCAN model on semantically segmented point cloud can effectively remove dynamic objects, reducing the “ghosts” formed by dynamic objects in the point cloud maps. Tables III and IV list the RPE and absolute trajectory error (ATE) information after removing dynamic objects from the first ten sequences in the KITTI dataset, respectively. It can be seen that in sequences

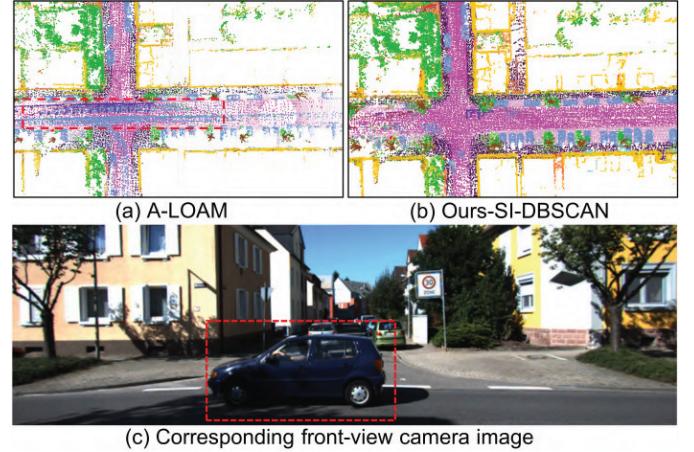
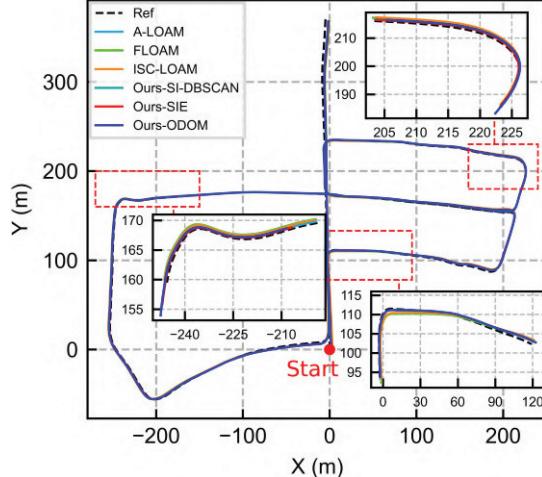
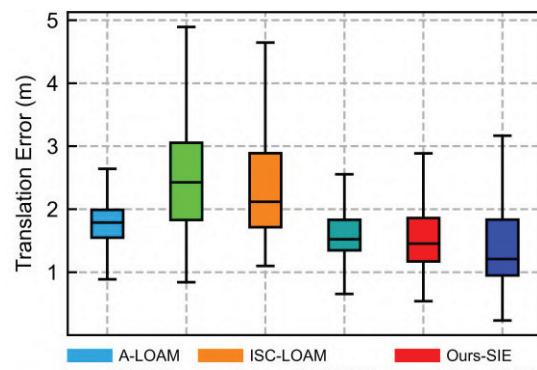


Fig. 5. Comparison of local mapping between the Ours-SI-DBSCAN (a) for eliminating dynamic objects and the baseline method A-LOAM (b). (c) Dynamic objects appearing in the local mapping.

with many dynamic objects, such as sequences 05 and 07, our proposed Ours-SI-DBSCAN can effectively improve the reliability and robustness of the odometry. Specifically, compared with A-LOAM, which does not remove dynamic objects, Ours-SI-DBSCAN achieves a 20% reduction in the average RMSE of the ATE. Furthermore, Ours-SI-DBSCAN demonstrates superior performance in ATE accuracy over state-of-the-art



(a)



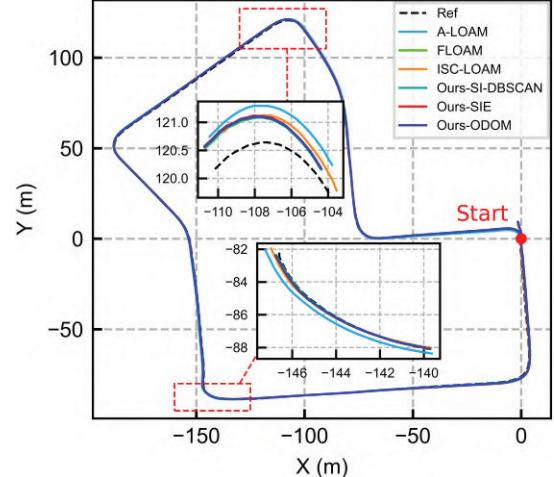
(b)

Fig. 6. (a) Trajectory plot and (b) box plot of absolute translation error for the KITTI 05 sequence.

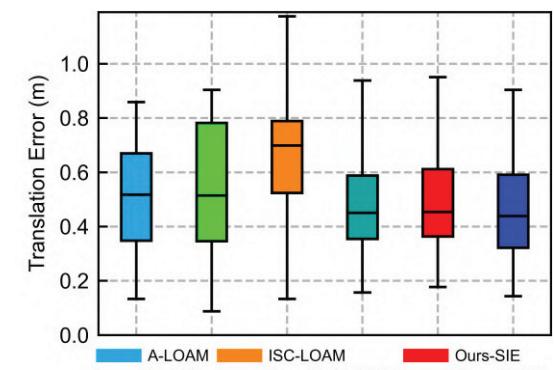
dynamic object removal methods, including SA-LOAM and TRLO.

2) *Validate SIE-Assisted LiDAR Odometry:* Accurate semantic information can enhance the robustness of the LiDAR SLAM. By performing semantic segmentation on LiDAR point, the SLAM system can recognize and understand various objects and scenes in the environment. Traditional LiDAR SLAM systems usually construct maps of geometric information, lacking semantic understanding of the environment. With the introduction of point cloud semantic segmentation, the map contains not only geometric information but also semantic information about objects, making the map more enriched. Furthermore, semantic segmentation can directly filter out irrelevant point cloud data in mapping, improving the quality of feature point, reducing problem of point misalignment, and thus enhancing positioning accuracy. However, semantic segmentation is not always perfect. It provides a richer understanding of the environment, but it also introduces uncertainty into the point. Therefore, we introduce the concept of information entropy to estimate the uncertainty of point cloud after semantic segmentation.

To verify the impact of SIE on the performance of LiDAR odometry, we associate data with semantically labeled objects and establish constraints combined with geometric



(a)



(b)

Fig. 7. (a) Trajectory plot and (b) box plot of absolute translation error for the KITTI 07 sequence.

information, whereas using SIE as the weight information for point cloud matching. We use RPE and ATE as accuracy evaluation metrics. Tables III and IV, respectively, show the accuracy evaluation results of our proposed method Ours-SIE and other approaches. We can find that compared with the baseline method A-LOAM, our method Ours-SIE significantly improves the accuracy of the odometry. SA-LOAM is an excellent scheme among the existing LOAM-based methods that use semantic information to assist LiDAR odometry, and our method exceeds SA-LOAM on most sequences. This means that adopting SIE as the weight information for feature matching can effectively enhance the accuracy of LiDAR odometry and simultaneously estimate the uncertainty of the feature point cloud.

3) *Validate SI-DBSCAN and SIE Jointly Assisting LiDAR Odometry:* To verify the combined effect of SI-DBSCAN and SIE-weight on the performance of LiDAR odometry, we select the first ten sequences from the KITTI dataset for extensive analysis. Initially, we downsample the LiDAR scan frames based on semantic information, unlike voxel downsampling, which retains more useful point information. Subsequently, we incorporate the SI-DBSCAN module and SIE-weight to assist in feature matching for LiDAR odometry. Table III presents the numerical values of relative pose accuracy evaluations for

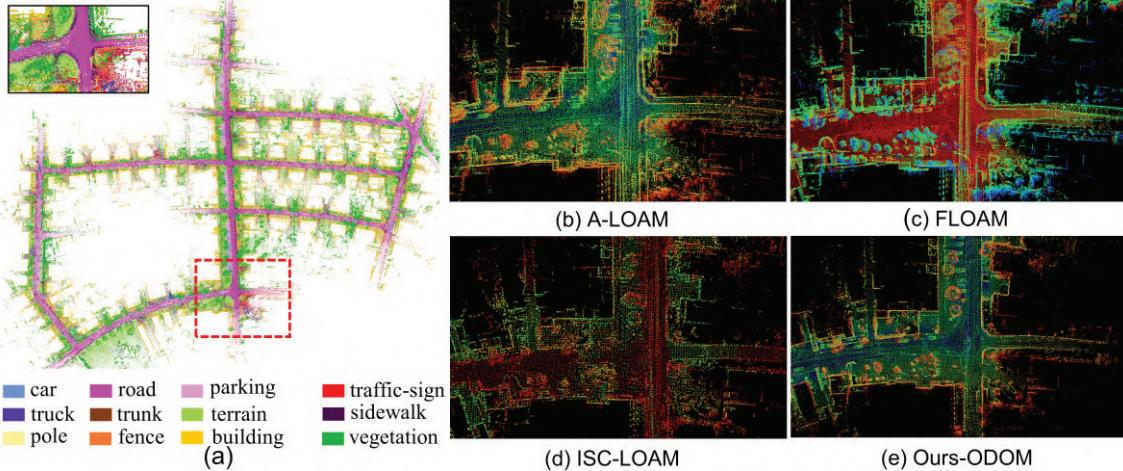


Fig. 8. Semantic map of the KITTI dataset sequence 05, where the enlarged area in (a) shows mapping results after removing dynamic object interference and (b)–(e) present local maps from various methods.

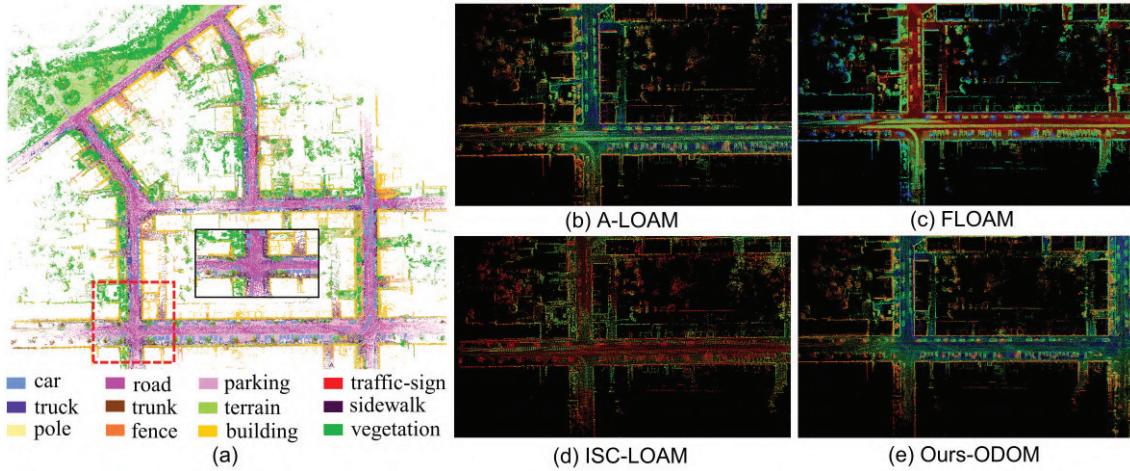


Fig. 9. Semantic map of the KITTI dataset sequence 07, where the enlarged area in (a) shows the mapping results after removing dynamic object interference and (b)–(e) present local maps from various methods.

different schemes, revealing that compared with the benchmark method A-LOAM, our proposed Ours-ODOM significantly improves RPE in each sequence, particularly in sequences 01 and 08, where Ours-ODOM reduces RPE by 26% and 18%, respectively. Compared with SA-LOAM, a method that assists LiDAR with semantic information, Ours-ODOM maintains the same or higher RPE in most sequences. However, for sequences with fewer dynamic objects, such as 01 and 10, the accuracy of feature odometry deteriorates due to the incorrect exclusion of some static point cloud. Table IV shows the numerical values of absolute trajectory accuracy evaluations for different schemes, indicating that in assessing global trajectory accuracy, our proposed Ours-ODOM performs better on some sequences with more dynamic objects and longer distances compared with other methods. In sequence 04, where the trajectory is only 393 m and no dynamic objects are present, the ISC-LOAM method achieves the highest accuracy by leveraging its integration of geometric and intensity features. To qualitatively assess the reliability of the method proposed in this article, we selected

sequences 05 and 07, which have more dynamic objects and complex environments, for evaluation.

Figs. 6(a) and 7(a) show the trajectories for the KITTI dataset test sequences 05 and 07. Since SA-LOAM is not open-sourced, this scheme was not included in the trajectory comparison. The local zoomed areas of the trajectory plots contain many dynamic objects, and due to the cumulative error of the odometry and the interference of dynamic objects, it can be observed that Ours-ODOM is closer to the true trajectory, minimizing the impact of dynamic objects on localization and mapping as much as possible. Figs. 6(b) and 7(b) show the box plots of translational error for absolute trajectory accuracy of the two sequences, respectively. Even after removing several outliers from the box plot, the visual representation still clearly reflects the significant improvement in LiDAR odometry performance brought by our proposed solution. Figs. 8 and 9 show the global semantic mapping generated by our Ours-ODOM method for the KITTI dataset sequences 05 and 07, along with local mapping from other comparison methods. The highlighted local areas in the maps are situations where

other dynamic objects appear simultaneously as the vehicle moves. It can be observed that using the SI-DBSCAN model effectively reduces the interference of dynamic objects and establishes a more accurate semantic map.

V. CONCLUSION

In this article, we present a fast and robust vehicle LiDAR odometry assisted by the SI-DBSCAN empirical model for dynamic object identification and SIE weighting for feature registration, which is suitable for complex environmental scenes with numerous dynamic objects. The proposed SI-DBSCAN model performs cluster analysis on the potential dynamic point clouds after semantic segmentation and rapidly identifies dynamic objects based on the distance differences between clustering centroids in adjacent frames. Furthermore, by innovatively using SIE to consider the uncertainty of semantic segmentation results and using it as weight information, the reliability of point feature matching is enhanced. The evaluation results on the KITTI dataset show that our proposed SI-DBSCAN model can quickly and effectively identify dynamic objects, which operates independently of the dynamic object types. In addition, the SIE provides accurate weight information for each error term in feature matching, thereby further enhancing the robustness of the LiDAR odometry. The new method proposed in this research has practical value in the field of vehicle LiDAR odometry and serves as an important reference for future related research. We will continue to focus on more effective dynamic object identification research to further improve the consistency and reliability of mapping.

REFERENCES

- [1] S. A. S. Mohamed, M. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A survey on odometry for autonomous navigation systems," *IEEE Access*, vol. 7, pp. 97466–97486, 2019.
- [2] D. Lee, M. Jung, W. Yang, and A. Kim, "LiDAR odometry survey: Recent advancements and remaining challenges," *Intell. Service Robot.*, vol. 17, no. 2, pp. 95–118, Mar. 2024.
- [3] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot., Sci. Syst. Conf.*, Berkeley, CA, USA, Jul. 2014, pp. 1–9.
- [4] 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.
- [5] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5135–5142.
- [6] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.
- [7] K.-N. Lianos, J. L. Schonberger, M. Pollefeys, and T. Sattler, "VSO: Visual semantic odometry," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 234–250.
- [8] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and accurate LiDAR semantic segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4213–4220.
- [9] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds for autonomous driving," 2020, *arXiv:2003.03653*.
- [10] S. W. Chen et al., "SLOAM: Semantic LiDAR odometry and mapping for forest inventory," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 612–619, Apr. 2020.
- [11] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4530–4537.
- [12] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Proc. Robot., Sci. Syst.*, Jun. 2018, p. 59.
- [13] L. Li et al., "SA-LOAM: Semantic-aided LiDAR SLAM with loop closure," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 7627–7634.
- [14] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1907–1915.
- [15] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.
- [16] O. Kramer and H. Danielsiek, "DBSCAN-based multi-objective niching to approximate equivalent Pareto-subsets," in *Proc. 12th Annu. Conf. Genetic Evol. Comput.*, Jul. 2010, pp. 503–510.
- [17] Q. M. Thomas, O. Wasenmüller, and D. Stricker, "DeLiO: Decoupled LiDAR odometry," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1549–1556.
- [18] S. Lee, C. Kim, S. Cho, S. Myoungho, and K. Jo, "Robust 3-dimension point cloud mapping in dynamic environment using point-wise static probability-based NDT scan-matching," *IEEE Access*, vol. 8, pp. 175563–175575, 2020.
- [19] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2095–2101.
- [20] 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.
- [21] M. Oelsch, M. Karimi, and E. Steinbach, "R-LOAM: Improving LiDAR odometry and mapping with point-to-mesh features of a known 3D reference object," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2068–2075, Apr. 2021.
- [22] S. Chen, X. Li, G. Huang, Q. Zhang, and S. Wang, "NHC-LIO: A novel vehicle LiDAR-inertial odometry (LIO) with reliable nonholonomic constraint (NHC) factor," *IEEE Sensors J.*, vol. 23, no. 21, pp. 26513–26523, Nov. 2023.
- [23] M. Ji, W. Shi, Y. Cui, C. Liu, and Q. Chen, "Adaptive denoising-enhanced LiDAR odometry for degeneration resilience in diverse terrains," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–15, 2024.
- [24] C. Qian, Z. Xiang, Z. Wu, and H. Sun, "RF-LIO: Removal-first tightly-coupled LiDAR inertial odometry in high dynamic environments," 2022, *arXiv:2206.09463*.
- [25] P. Pfreundschuh, H. F. C. Hendrikx, V. Reijgwart, R. Dubé, R. Siegwart, and A. Cramariuc, "Dynamic object aware LiDAR SLAM based on automatic generation of training data," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11641–11647.
- [26] D. Yoon, T. Tang, and T. Barfoot, "Mapless online detection of dynamic objects in 3D LiDAR," in *Proc. 16th Conf. Comput. Robot Vis. (CRV)*, May 2019, pp. 113–120.
- [27] J. Schauer and A. Nüchter, "The peopleRemover—Removing dynamic objects from 3-D point cloud data by traversing a voxel occupancy grid," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1679–1686, Jul. 2018.
- [28] 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.
- [29] 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.
- [30] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.
- [31] X. Zhang et al., "RI-fusion: 3D object detection using enhanced point features with range-image fusion for autonomous driving," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.
- [32] G. Chen, B. Wang, X. Wang, H. Deng, B. Wang, and S. Zhang, "PSF-LO: Parameterized semantic features based LiDAR odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 5056–5062.
- [33] L. Li et al., "SSC: Semantic scan context for large-scale place recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 2092–2099.
- [34] K. Litomiský and B. Bhanu, "Removing moving objects from point cloud scenes," in *Proc. Adv. Depth Image Anal. Appl., Int. Workshop*, Tsukuba, Japan. Cham, Switzerland: Springer, Jan. 2013, pp. 50–58.

- [35] X. Chen et al., "Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6529–6536, Oct. 2021.
- [36] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9296–9306.
- [37] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [38] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2019, pp. 8024–8035.
- [39] Y. Jia, T. Wang, X. Chen, and S. Shao, "TRLO: An efficient LiDAR odometry with 3D dynamic object tracking and removal," 2024, *arXiv:2410.13240*.



Shipeng Chen received the bachelor's and M.Sc. degrees from Chang'an University, Xi'an, China, in 2021 and 2024, respectively. He is currently pursuing the Ph.D. degree in control science and engineering with Harbin Institute of Technology, Harbin, China.

His research activities include LiDAR SLAM/IMU-integrated navigation and deep learning.



Xin Li received the bachelor's degree from Southwest Jiaotong University, Chengdu, China, in 2011, and the M.Sc. and Ph.D. degrees in surveying engineering from Wuhan University, Wuhan, China, in 2013 and 2017, respectively.

He is currently an Associate Professor with Chang'an University, Xi'an, China. His research activities mainly include GNSS/SINS-integrated navigation, multisensors' fusion theory, and application.



Songhui Ma received the B.Sc., M.Sc., and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 1999, 2002, and 2007, respectively.

He is currently an Associate Professor with Northwestern Polytechnical University. His main research interests include flight control, vision navigation, and obstacle avoidance of UAV.



Shuo Wang received the bachelor's and M.Sc. degrees from Chang'an University, Xi'an, China, in 2020 and 2024, respectively.

His research activities include visual SLAM/IMU-integrated navigation and deep learning.



Xianrui Ren received the bachelor's degree from Shenyang Agricultural University, Shenyang, China, in 2022. She is currently pursuing the Ph.D. degree with Chang'an University, Xi'an, China.

Her research activities include visual SLAM/IMU-integrated navigation.