Segmented Curved-Voxel Occupancy Descriptor for Dynamic-Aware LiDAR Odometry and Mapping

Yixin Fang, Kun Qian, *Member, IEEE*, Yun Zhang, Tong Shi, and Hai Yu

***Abstract*— Simultaneous localization and mapping (SLAM) employing 3-D light detection and ranging (LiDAR) data constitutes an indispensable perception technology for geospatial sensing of the surrounding environments. Nevertheless, the existence of dynamic objects can substantially impair sensing performance. This article proposes a novel egocentric descriptor named *segmented curved-voxel occupancy descriptor* (SCV-OD), which serves as the backbone for constructing a dynamic-aware and LiDAR-only SLAM in a tight-coupled and consistent manner. Assisted with LiDAR intensity and geometric features, the object segmentation module clusters curved voxels into objects and recognizes potential dynamic (PD) objects as prior knowledge. Then, in the object tracking module, PD objects are tracked through curved-voxel overlay, and high dynamic (HD) objects are removed according to the object overlap ratio. The aforementioned two modules are closely coupled to mutually compensate for accuracy loss by sharing the same SCV-OD. Finally, the voxelized generalized iterative closest point (VGICP)-based LiDAR mapping module optimizes LiDAR poses by considering dynamic objects and results in a global static instance map. We validated the proposed method on the public dataset (KITTI) and our custom dataset. The evaluation results illustrate that our method outperforms the state-of-the-art (SOTA) LiDAR-only methods in pose estimation and dynamic removal.**

***Index Terms*— Dynamic removal, geospatial sensing, light detection and ranging (LiDAR) data, pose estimation, simultaneous localization and mapping (SLAM).**

NOMENCLATURE

$\mathcal{F}_t$, T_t	LiDAR frame and pose at timestamp t.
p_n, ϵ_n	nth Cartesian and spherical coordinate.
δ_n	Intensity value of the nth point.
$T_{j,i}$	Transformation matrix from $\mathcal{F}_i$ to $\mathcal{F}_j$.

I. INTRODUCTION

RECENTLY, light detection and ranging (LiDAR) sensors have been widely employed to sample 3-D geospatial information of the environment, owing to their ability to provide lighting-invariant and precise measurements. However, the scan data provides a transient snapshot that captures dynamic objects like cars and pedestrians, resulting in a notable reduction in the accuracy of LiDAR-based odometry [1], [2], [3], [4], [5], [6], [7], [8], localization [9], [10], and mapping [11].

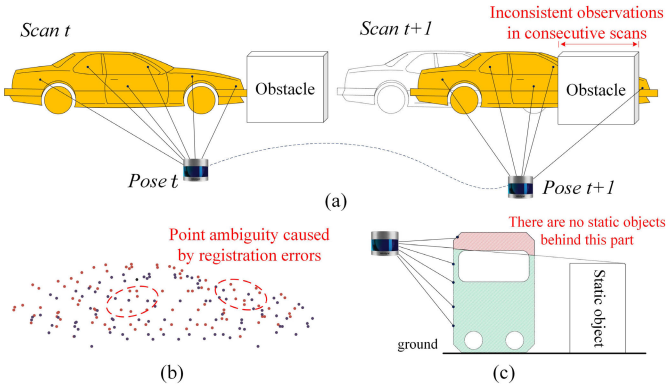


Fig. 1. Potential limitations in LiDAR-only dynamic removal methods include (a) inconsistent observations stemming from occlusion, (b) point ambiguity resulting from registration errors, and (c) irremovable dynamic parts arising from restricted viewpoints. The highlighted parts in green are disabled to be removed, while those in red are not.

To achieve the effectiveness of measurements in dynamic environments, many studies [12], [13], [14] mainly focus on fusing the information from multiple sensors [e.g., camera, inertial measurement unit (IMU), and global navigation satellite system (GNSS)] to compensate for the drift in LiDAR odometry, but they struggle with calibration accuracy covering and signal reception constraining. For the detection of dynamic objects, cameras can provide plentiful visual information for object recognition with the deep-learning methods [15], [16]. Unlike cameras, 3-D LiDAR generates point clouds with disequilibrium and inhomogeneity, simply providing basic geospatial information about spatial environments. Many previous LiDAR-only dynamic removal methods [17], [18], [19], [20], [21], [22] can erase most dynamic objects by associating dynamic information with the occupancy changes based on range images, grids, or voxels. However, because these methods overlook the object-level concept, they frequently generate false detections when it comes to accurately representing dynamic elements in the presence of challenges illustrated in Fig. 1(a)–(c), including inconsistent observations, point ambiguity, and restricted viewpoints. To address that object-based methods [23], [24], [25] are introduced to remove dynamic

Manuscript received 31 May 2023; revised 18 October 2023 and 21 December 2023; accepted 24 January 2024. Date of publication 2 February 2024; date of current version 13 February 2024. This work was supported in part by the Jiangsu Province Natural Science Foundation under Grant BK20201264, in part by the Zhejiang Laboratory under Grant 2022NB0AB02, and in part by the National Natural Science Foundation of China under Grant 61573101. *(Corresponding author: Kun Qian.)*

Yixin Fang, Kun Qian, Yun Zhang, and Tong Shi are with the School of Automation and the Key Laboratory of Measurement and Control of CSE, Ministry of Education, Southeast University, Nanjing 210096, China (e-mail: kqian@seu.edu.cn).

Hai Yu is with the Power Grid Digitization Technology Research Institute, State Grid Smart Grid Research Institute, Nanjing 210003, China.

Digital Object Identifier 10.1109/TGRS.2024.3361868

1558-0644 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

elements at the object level, but their performances are heavily affected by the accuracy loss in object segmentation and they cannot robustly track the positions offset of objects between consecutive scans. Therefore, a crucial issue in LiDAR-only methods is to construct a more rational expression for point cloud and dynamic information.

This article proposes a novel egocentric descriptor termed *segmented curved-voxel occupancy descriptor* (SCV-OD), which considers curved-voxel occupancy detection at the object level. The SCV-OD serves as the backbone to build a dynamic-aware and LiDAR-only Simultaneous localization and mapping (SLAM), where we implement object segmentation, object tracking, and LiDAR mapping in a tight-coupled and consistent manner. Fig. 2 presents the application result of our method on sequence 05 of the KITTI dataset [26]. Object segmentation provides prior knowledge for object tracking which removes dynamic objects. The LiDAR mapping estimates initial motion and updates local maps with considering dynamic objects. Compared with existing methods, our approach demonstrates its uniqueness by incorporating curved-voxel occupancy detection at the object level to achieve pose estimation and dynamic removal in real-time. The main contributions are fourfold as follows.

1) We proposed an egocentric descriptor termed SCV-OD, based on which a dynamic-aware and LiDAR-only SLAM is built for mobile sensing in dynamic environments.
2) We proposed a multistage object segmentation method where we proposed *restriction then promotion criterion* (RPC) for object clustering and *prominent geometric features selection* (PGS) for object classification. This method is closely coupled with a lightweight object tracking strategy through *tight coupling* (TC), allowing for dynamic object removal and imperfect object refinement by curved-voxel occupancy detection.
3) We developed a voxelized generalized iterative closest point (VGICP)-based LiDAR mapping module for the SCV-OD, which utilizes a passive iteration process to match adjacent voxels and constructs a hash table to manage motion labels. This module estimates motions at 10 Hz and updates local maps at 20 Hz, meeting the real-time requirements of a single CPU.
4) Using the SCV-OD as the backbone, we developed a tight-coupled and consistent framework including object segmentation, object tracking, and LiDAR mapping. The evaluation results demonstrated the robustness of our method in pose estimation and dynamic removal.

The rest of this article is organized as follows. Related works are introduced in Section II. Section III describes the proposed SCV-OD. Sections IV–VI present the proposed object segmentation, object tracking, and LiDAR mapping modules, respectively. Section VII discusses the experimental results in detail. Conclusions are presented in Section VIII.

Fig. 2. (Middle) Application result of the proposed method on sequence 05 of KITTI [26]. The left column at the top and the bottom shows the original map containing moving objects in red and the right column represents our mapping result where ground, buildings, vegetation, and others (e.g., parked cars) are assigned yellow, brown, green, and random colors, respectively.

II. Related Works

A. LiDAR-Based SLAM

The development of 3-D LiDAR SLAM for mobile sensing has been extensively pursued using various processing and optimization strategies. The basic approach directly utilizes raw point clouds with the iterated closest point (ICP) algorithm [1] without any preprocessing. To reduce the impact of subtle environmental changes, the normal distributions transform (NDT) algorithm [2] is tailored to scan matching based on the Gaussian model. The 3-D point clouds are first subdivided into each cell and the probability of each measuring point is assigned a normal distribution. By extent, Koide et al. [4] creatively proposed VGICP algorithm [4], which calculates voxel distributions from point positions to avoid costly nearest neighbor search while retaining its accuracy.

The point clouds generated from 3-D LiDAR typically consist of large streaming volumes. However, there is a lack of sufficient computing resources to process these data promptly. Feature-based methods [5], [6], [7] extract edge and planar features from raw point cloud according to the smoothness of points in diverse laser beams, to perform point-to-line and point-to-plane scan matching tactics. Additionally, Li et al. [27] proposed a novel feature extraction based on geometry and intensity to address degeneracy in indoor environments. Additionally, Zhou et al. [8] introduced T-LOAM using truncated least squares (TLS), which applies dynamic curved-voxel clustering to extract stable characteristics and recover robust LiDAR poses. However, these methods may fail to obtain sufficient valid features from degraded scenes or point cloud

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

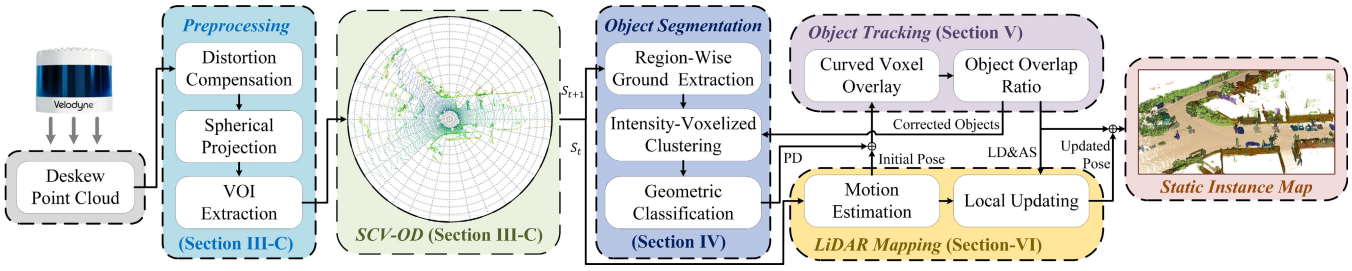


Fig. 3. Overview of our proposed method. The main algorithms comprise three modules: object segmentation, object tracking, and LiDAR mapping, which are integrated in a tight-coupled and consistent manner by using the SCV-OD as the backbone. The details are described in Sections IV–VI.

inputs containing dynamic outliers, leading to a decrease in the accuracy of motion estimation.

B. Dynamic Removal

Visibility-based methods convert 3-D point clouds into 2-D range images and detect pixelwise differences by comparing consecutive range images. Kim and Kim [17] presented a multiresolution range image-based false prediction reverting algorithm utilizing a pixel-to-window comparison method to compensate for the point ambiguity implicitly. By extension, Liu et al. [18] extended the detection range by using a low-channel roadside light detection, to identify the near-range traffic objects with the density-based spatial clustering of applications (DBSCAN). Park et al. [19] applied a nonparametric model to estimate the static background and addressed the point ambiguity through false detection suppression. Though various models are utilized to tackle the false detection caused by point ambiguity shown in Fig. 1(b) and restricted viewpoint shown in Fig. 1(c), they still struggle with resolution settings.

Ray-tracing-based methods encode LiDAR scans into unitary representations such as grids and voxels, then erase dynamic objects by detecting unit occupancy changes. Hornung et al. [20] proposed an Octomap that counts the hits and misses of scans in the grid map to fetch dynamic units. Schauer and Nüchter [21] traversed regular occupancy voxels along the sight lines between the sensor and the measured points to find the volumetric occupancy differences in consecutive scans. By extension, Lim et al. [22] proposed the representation of points organized in vertical columns to obtain bins containing potential dynamic (PD) objects and remove overground points. This category converts dynamic object removal into regional dynamic detection without considering the object-level concept, causing excessive removal of actual static (AS) points when there are inconsistent observations shown in Fig. 1(a).

Learning-based methods generally perform as postprocessing steps for object detection. Chen et al. [23] designed an automatic data labeling pipeline for 3-D LiDAR data to save the extensive manual labeling effort and to improve the performance of existing learning-based moving object segmentation (MOS) systems by automatically annotation training data. Sun et al. [24] utilized a range image-based dual-branch structure to separately deal with spatial–temporal information and designed a point refinement module via 3-D sparse convolution to fuse range and point information. Wang et al. [25] utilized 4-D sparse convolutions to extract spatiotemporal features and an upsample fusion module to output pointwise labels by fusing features and predicted instance information. These methods improve the dynamic removal performance significantly, but misclassification and mislabeling might cause irreparable results.

III. SCV-OD for LiDAR Sensing

A. Method Overview

Fig. 3 shows a schematic of our proposed SLAM which works on the backbone of SCV-OD (see Section III-C) comprises three main modules: object segmentation (see Section IV), object tracking (see Section V), and LiDAR mapping (see Section VI).

According to the motion properties, objects can be primarily divided into four categories: high dynamic (HD) objects (e.g., traveling vehicles and walking pedestrians), low dynamic (LD) objects (e.g., parked vehicles and stopped pedestrians), PD objects (i.e., HD and LD), and AS objects (i.e., ground, buildings, and vegetation).

First, the target SCV-OD ($\mathcal{S}_{t+1}$) is initialized through preprocessing steps that include distortion compensation, spherical projection, and volume of interest (VOI) extraction, and the initial motion is estimated with the source SCV-OD ($\mathcal{S}_t$) through motion estimation. Then, the object segmentation method including regionwise ground extraction, intensity-voxelized clustering, and geometric classification, is employed to distinguish PD objects as prior knowledge for the object tracking module. Only PD objects are tracked using a curved-voxel overlay, while HD objects are removed based on the object overlap ratio. Additionally, a tightly coupled approach is employed to refine imperfect objects by correcting their shapes and labels, compensating for the accuracy loss in segmentation. The local updating module utilizes the LD objects and AS objects as the base inputs for local updating, which optimizes the LiDAR poses. Finally, LiDAR poses according to timestamps are optimized and the global static instance map is generated.

Hence, our primary objective is to preserve the AS and LD objects for pose optimization while removing the HD objects in consecutive frames. The resulting static instance map $\hat{\mathcal{M}}$ with the optimized poses $\mathbf{T}_t^*$ can be estimated as

$$\hat{\mathcal{M}} = \bigcup_{t\in[T]} \mathbf{T}_t^* \cdot (\mathrm{AS}_t \cup \mathrm{LD}_t) \tag{1}$$

where $[T]$ equals the total frame timestamps.

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

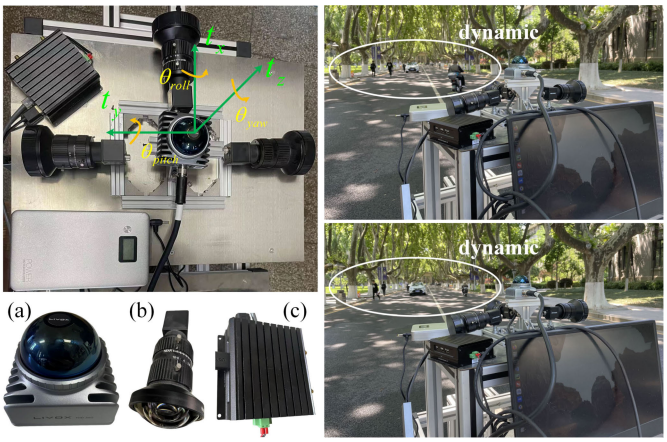


Fig. 4. Mobile sensing platform in the experiments, equipped with (a) Livox Mid360, (b) industrial cameras, and (c) Nvidia Jetson Nano with Maxwell 128-Core GPU and ARM Cortex-A57 CPU.

B. Sensing Platform

The 3-D LiDAR sensors utilized in our experimental validation include Velodyne HDL-64E and Livox Mid360 and both of them enable a 360° field of view (FoV) from a perspective view. The primary distinction between them is the scanning mechanisms. The Velodyne HDL-64E employs mechanical spinning to generate LiDAR scans, while the Livox Mid360 is a solid-state sensor that receives point clouds through time integration.

Fig. 4 illustrates our mobile sensing platform equipped with the relevant sensors. Livox Mid360 sensor offers a random angular error of 0.15°, providing a 59° (−7°–52°) vertical FoV. It is capable of distance measurements exceeding 80 m with a precision of 2 cm. We set the integrated time as 0.1 s and ensure that the valid reflectance of the LiDAR intensity channel is above 80%. The Velodyne HDL-64E with the high vertical resolution (0.4°) is provided by the KITTI odometry benchmark [26] which possesses a 26.9° vertical FoV.

An alloy vehicle with a height of 1.5 m and an average movement speed of 0.5 m/s is used as the data-recording platform in the experiment, The ego-motion coordinate system of position and orientation is employed in our framework as shown in Fig. 3. Two computers are computationally adopted for the proposed framework, including an Nvidia Jetson Nano RTSS-X506/Z506 and a consumer-level laptop with an Intel Core i7-9750Q and 16 GB of RAM. The Nvidia Jetson Nano is an embedded computing device equipped with a Maxwell 128-Core GPU and an ARM Cortex-A57 CPU.

C. Segmented Curved-Voxel Occupancy Descriptor

The novel egocentric occupancy descriptor SCV-OD utilizes curved voxels to describe the occupancy of objects. Fig. 5 depicts the spatial structure of curved voxel [28].

The LiDAR motion is modeled with constant angular and linear velocities during a sweep, which allows us to implement point distortion by linearly interpolating the pose transform within a sweep for the points according to timestamps [5].

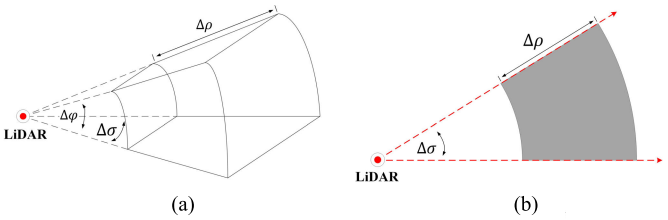


Fig. 5. Spatial structure of the curved voxel. (a) Perspective view of a curved voxel. (b) Top view of a curved voxel (gray) and two laser beams emitted from the sensor (red).

Let LiDAR poses $\mathbf{T}_t$ be the initial pose resulted from *motion estimation* (see Section VI) at the starting sweep of frame $\mathcal{F}_t$ in six-degree of freedom (DoF), $\mathbf{T}_t = [l_x, l_y, l_z, \theta_x, \theta_y, \theta_z]$, where are translations along the x-, y-, and z-axes and θ_x, θ_y, and θ_z are rotation angles, following the right-hand rule. Let $T_{t,\tau}$ be the lidar pose transform between $[t, \tau]$. We recall τ be the current timestamp and τ_n $(t < \tau_n < t + 1)$ be the timestamp of the nth point in frame $\mathcal{F}_t$. Then, the interpolated pose T_{t,τ_n} for the nth point can be calculated as

$$T_{t,\tau_n} = \frac{\tau_n - t}{\tau - t} T_t. \tag{2}$$

After point distortion, we convert Cartesian coordinates into spherical ones $\epsilon_n = [\rho_n, \sigma_n, \varphi_n]^T$ via a spherical mapping $\Pi : \mathbb{R}^3 \to \mathbb{R}^3$, which is denoted as

$$\epsilon_n = \Pi(p_n) = \begin{bmatrix} \sqrt{x_n^2 + y_n^2} \\ \arctan(y_n / x_n) \\ \arctan(z_n / \rho_n) \end{bmatrix} \tag{3}$$

where ρ_n, σ_n, and φ_n are the radial distance, polar angle, and azimuth angle, respectively. Then, the VOI named $\mathcal{V}_t$ for valid point space is defined as

$$\mathcal{V}_t = \{\{p_n, \epsilon_n\} | \rho_n < R_{\max}, A_{\min} < \varphi_n < A_{\max}\} \tag{4}$$

where $R_{\max}$ and $A_{\min,\max}$ represent the boundary of radial distance and azimuth angle, respectively. Dynamic objects of our interest, such as vehicles or pedestrians, typically remain within a reasonable range.

With the initialization of the LiDAR frame, $\mathcal{V}_t$ is divided into curved voxels over the regular interval of radial, polar, and azimuth directions. Let N_ρ, N_σ, and N_φ be the number of indexes in the three directions. Then, the SCV-OD, which is denoted as $\mathcal{S}_t$, can be represented as

$$\mathcal{S}_t = \bigcup_{i \in [N_\rho], j \in [N_\sigma], k \in [N_\varphi]} \mathrm{SCV}_{ijk,t} \tag{5}$$

where $\mathrm{SCV}_{ijk,t}$ denotes the (i, j, k)th segmented curved voxel in $\mathcal{S}_t$. Note that due to the sparsity of the point cloud, only valid units containing one point at least are maintained as

$$\begin{aligned} \mathrm{SCV}_{ijk,t} = \big\{\{p_n, \epsilon_n\}, \mathcal{L}_{ijk} | &\Delta\rho \cdot (i-1) \le \rho_n < \Delta\rho \cdot i \\ &\Delta\sigma \cdot (j-1) \le \sigma_n < \Delta\sigma \cdot j \\ &\Delta\varphi \cdot (k-1) \le \varphi_n < \Delta\varphi \cdot k\big\} \end{aligned} \tag{6}$$

where $\Delta\rho = R_{\max}/N_\rho$, $\Delta\sigma = 2\pi/N_\sigma$, and $\Delta\varphi = (A_{\max} - A_{\min})/N_\varphi$ are unit sizes for each spherical direction. $\mathcal{L}_{ijk}$ denotes the 16-bit segmented label meaning that $\mathrm{SCV}_{ijk,t}$ is

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

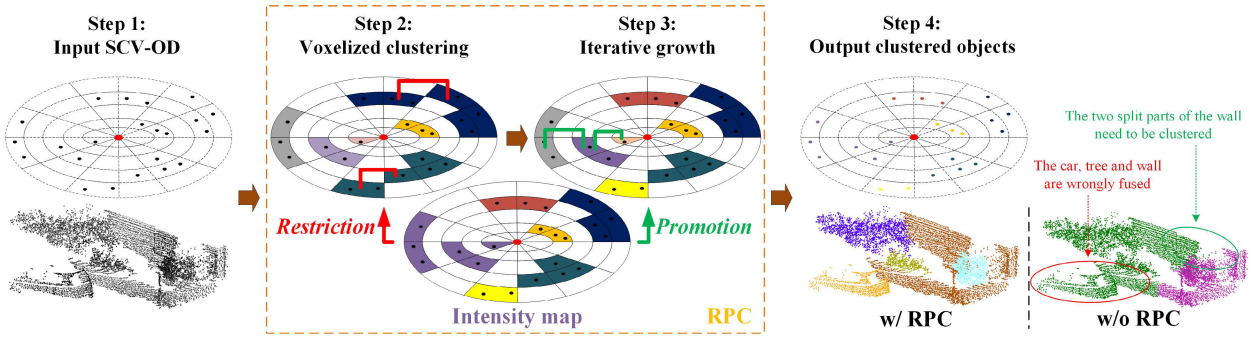


Fig. 6. Steps of intensity-voxelized clustering (top view) using RPC (orange box). (Bottom) In the intensity map, curved voxels in the same color depict the continuous intensity. The red and green lines represent the restriction and promotion. The output on the left is the result of our method with RPC, and the other is the result of the method without RPC.

occupied by a single instance, which can be assigned by the related curved voxels as

$$\mathcal{B}_{\mathrm{id},t} = \left\{ \mathrm{SCV}_{ijk,t} | \mathcal{L}_{ijk} = id \right\} \tag{7}$$

where $\mathcal{B}_{\mathrm{id},t}$ is the instance using the label id to fill $\mathcal{L}_{ijk}$.

IV. Multistage Object Segmentation

To tackle the challenges of limited information from LiDAR data and the computational burden associated with object identification, the proposed multistage object segmentation method incorporates intensity-voxelized clustering using RPC and geometric classification using PGS.

A. Regionwise Ground Extraction

Since the ground points affect the separation of terrestrial vehicles and pedestrians, it is necessary to extract them previously. Assuming that ground points only exist in the lowest curved voxels with the smallest index in the azimuth direction, the (u, v)th ground region $\mathcal{R}_{uv,t}$ can be defined as

$$\mathcal{R}_{uv,t} = \left\{ p_n | p_n \in \arg \min_{\mathrm{SCV}_{uvk,t}} k, z_n < \bar{z}_{uv,t} + \gamma_{\mathrm{seed}} \right\} \tag{8}$$

where $\bar{z}_{uv,t}$ is the mean value of z and γ_{seed} is the height margin. Then, the covariance matrix $C_{uv,t}$ is calculated as

$$C_{uv,t} = \sum_{n \in N} \left(p_n - \bar{p}_{uv,t} \right) \left(p_n - \bar{p}_{uv,t} \right)^T \tag{9}$$

where N and $\bar{p}_{uv,t}$ denote the number of points and the mean position of $\mathcal{R}_{uv,t}$. Let the eigenvector with the smallest eigenvalue be $n_{uv,t} = [a_{uv,t}, b_{uv,t}, c_{uv,t}]^T$. Then, the plane equation can be calculated as $d_{uv,t} = -n_{uv,t}^T \cdot \bar{p}_{uv,t}$. Finally, the estimated ground points are extracted as follows:

$$\mathcal{G}_{uv,t} = \left\{ p_n | p_n \in \mathcal{R}_{uv,t}, d_{uv,t} - d_{uv,t}^n < g_{\mathrm{seed}} \right\} \tag{10}$$

where $d_{uv,t}^n = -n_{uv,t}^T \cdot p_n$ and g_{seed} represents the distance margin of the ground plane.

Algorithm 1 RPC

Data: initial egocentric occupancy descriptor $\mathcal{S}_t$
Result: a set of objects $\{\mathcal{B}_{id,t}\}$
Let $m \leftarrow 0$ be the clustered object index;
Let $N_{ijk,t}^v$ be the neighbor curved voxels of $\mathrm{SCV}_{ijk,t}$;
Create intensity map ζ_t through (12);
——————**-Voxelized Clustering**——————
for $SCV_{ijk,t}$ **do**
  $N_{ijk,t}^v = E_1(\mathrm{SCV}_{ijk,t})$;
  if $N_{ijk,t}^v$ *are not clustered* **then**
    Assign $N_{ijk,t}^v$ as a new object $\mathcal{B}_{id=m,t}$;
    $m = m \oplus 1$; // Bitwise operation
  else
    Add $\mathrm{SCV}_{ijk,t}$ into the related object;
  end
end
Obtain initial object set $\{\mathcal{B}_{id,t} | id = 0, 1, \ldots, m\}$;
Let $N_{id,t}^{\mathcal{B}}$ be the neighbor objects of $\mathcal{B}_{id,t}$;
——————**-Iterative Growth**——————
for $\mathcal{B}_{id,t}$ **do**
  $N_{id,t}^{\mathcal{B}} = G_{3,1}(\mathcal{B}_{id,t})$;
  Combine $N_{id,t}^{\mathcal{B}}$ and update $\{\mathcal{B}_{id,t}\}$;
end
Return $\{\mathcal{B}_{id,t}\}$;

B. Intensity-Voxelized Clustering

LiDAR intensity provides valuable information about surface properties (e.g., roughness and reflectance) [29]. However, the LiDAR intensity channel is often filled with noise, making it challenging to accurately recover the true values of intensity through mathematical modeling. Therefore, we adapt a local correction approach to smooth the intensity values within a neighborhood. The correction function $\omega(\cdot)$ is defined as follows:

$$\omega(\delta_n) = \frac{\delta_n \cdot (\rho_n / \cos(\varphi_n))}{\cos < (x_n, y_n, z_n), \vec{\mu}_n >} \tag{11}$$

where $\vec{\mu}_n$ denotes the normal vector of p_n in the neighborhood.

To utilize the smoothed intensity effectively, an intensity map ζ_t is generated to capture the local continuity of object properties. Fig. 6 shows the steps of the intensity-voxelized

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

clustering method, where the intensity map ζ_t projected to $\mathcal{S}_t$ is defined as

$$\zeta_t = \bigcup_{i\in[N_\rho],j\in[N_\sigma],k\in[N_\varphi]} \{av_{ijk}, \mathrm{var}_{ijk}\} \tag{12}$$

where av_{ijk} and var_{ijk} denote the average and variance of the intensity value in $\mathrm{SCV}_{ijk,t}$, respectively.

As a voxelized clustering method using Euclidean distance, the RPC based on the intensity map ζ_t is proposed to specify the voxel searching process. First, the restriction function $E_r(\cdot)$ with the search radius r is formulated as

$$\begin{aligned} E_r\left(\mathrm{SCV}_{ijk,t}\right) = \{&\mathrm{SCV}_{uvw,t} | i-r \le u \le i+r, \\ &j-r \le v \le j+r, k-r \le w \le k+r \\ &|av_{ijk} - av_{uvw}| \le h_{av}, \mathrm{var}_{uvw} \le h_{\mathrm{var}}\} \end{aligned} \tag{13}$$

where $\mathrm{SCV}_{uvw,t}$ denotes the neighbor curved voxel of $\mathrm{SCV}_{ijk,t}$. h_{av} and h_{var} represent the thresholds. Then, the promotion function $G_{\tau,r}(\cdot)$ for the iterative growth of $\mathcal{B}_{\mathrm{id},t}$ is defined as

$$G_{\tau,r}\left(\mathcal{B}_{\mathrm{id},t}\right) = \left\{\mathcal{B}_{k,t} | \mathcal{B}_{k,t} \cap E_r\left(\mathcal{B}_{\mathrm{id},t}\right) \neq \emptyset\right\} \tag{14}$$

where $\mathcal{B}_{k,t}$ represents the neighbor object being engulfed by $\mathcal{B}_{\mathrm{id},t}$ and τ is the iteration time.

We discuss the RPC algorithm in Algorithm 1 in detail. First, the intensity map ζ_t is created (lines 3). Then, the voxelized clustering constrained by restriction function $E_1(\cdot)$ is employed to obtain initial object set $\{\mathcal{B}_{\mathrm{id},t} | id = 0, 1, \ldots, m\}$ (lines 4–14). The iterative growth using promotion function $G_{3,1}(\cdot)$ combines the neighbor objects (lines 16–20). Finally, the updated object set $\{\mathcal{B}_{\mathrm{id},t}\}$ is returned.

C. Geometric Classification

Once obtaining clustered objects, geometric features [9], [30] are utilized to recognize AS objects (e.g., buildings and vegetation) and PD objects (e.g., vehicles and people) by assigning a 7-D vector $f_{\mathrm{id}} = \{f_{\mathrm{id}}^1, f_{\mathrm{id}}^2\}$, which consists of two parts.

1) f_{id}^1 *Eigenvalue:* This feature is merged into a 4-D vector with linearity, planarity, scattering, and primary orientation, to describe the shape characteristics. The object point cloud is first normalized by centering, and the eigenvalues of its covariance matrix are computed as λ_1, λ_2, and λ_3 in descending order. Then, the linearity f_l, the planarity f_p and the scattering f_s are calculated as

$$f_l = \frac{\lambda_1 - \lambda_2}{\lambda_1}, \quad f_p = \frac{\lambda_1 - \lambda_3}{\lambda_1}, \quad f_s = \frac{\lambda_1}{\lambda_3}. \tag{15}$$

2) f_{id}^2 *Spatial Distribution:* This part consists of minimum height, maximum height, and scale in a 3-D vector, to represent the spatial information. Particularly, the scale represents the projected area on the ground.

Since geometric features can be coarse for object classification, we filter specific object categories by comparing the most prominent features. While this process may inevitably introduce classification errors, we compensate for the loss in classification accuracy through the TC in object tracking (see Section V) by fusing information from multiple segmented frames. Fig. 7 shows the process of PGS and the distribution of geometric features of different categories. The AS objects (e.g., buildings and trees) are first identified according to the linearity, planarity, and maximum height. Then, the rest of the clusters are considered PD objects (e.g., cars, cyclists and pedestrians), which are stated as

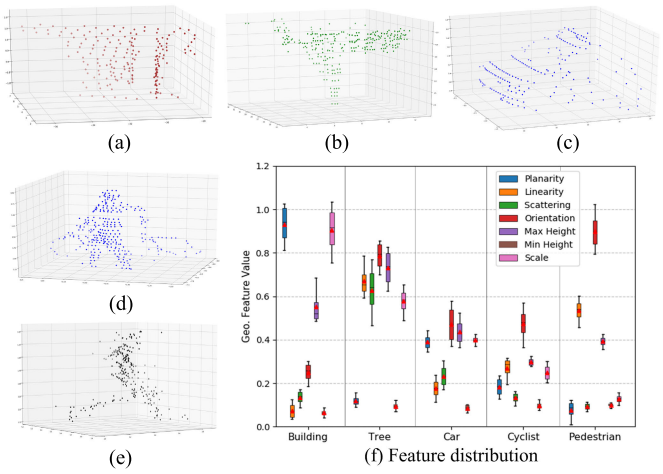


Fig. 7. (a)–(e) Various classes of segmented objects. (f) Value distribution of geometric features. (a) Building. (b) Tree. (c) Car. (d) Cyclist. (e) Pedestrian. (f) Feature distribution.

$$\mathcal{B}_{\mathrm{id},t}^{\mathrm{PD}} = \left\{\mathcal{B}_{\mathrm{id},t} | \bigtriangledown\left(f_{\mathrm{id}}\right)\right\} \tag{16}$$

where $\mathcal{B}_{\mathrm{id},t}^{\mathrm{PD}}$ denotes a PD object determined through the Boolean function $\bigtriangledown(\cdot)$.

V. Object Tracking via Occupancy Detection

Fig. 8 presents the object tracking model, which employs curved-voxel occupancy detection at the object level to remove HD objects. Furthermore, a TC is proposed by sharing the same SCV-OD between consecutive scans, which refines imperfect objects to compensate for the accuracy loss in object segmentation.

A. Curved-Voxel Overlay

For the object tracking in source $\mathcal{S}_t$ and target $\mathcal{S}_{t+1}$, the registration should be first implemented to unify the coordinate system. A lightweight registration method named curved-voxel overlay is proposed to create the unified viewpoint in target $\mathcal{S}_{t+1}$, which is denoted as

$$\mathrm{SCV}_{ijk,t}^{t+1} = \Pi\left(\mathbf{T}_{t+1,t}\left[p_{ijk,t}^{\mathrm{near}}, p_{ijk,t}^{\mathrm{cent}}, p_{ijk,t}^{\mathrm{far}}\right]\right) \tag{17}$$

where the point set represents the nearest, central, and farthest vertices of $\mathrm{SCV}_{ijk,t}$ is projected to match a single overlaid curved voxel $\mathrm{SCV}_{ijk,t}^{t+1}$ with a selection using RANSAC [31].

Therefore, each PD object $\mathcal{B}_{\mathrm{id},t}^{\mathrm{PD}}$ can be registered into $\mathcal{S}_{t+1}$ as

$$\mathcal{B}_{\mathrm{id},t}^{\mathrm{PD},t+1} = T_{t+1,t} \odot \mathcal{B}_{\mathrm{id},t}^{\mathrm{PD}} \tag{18}$$

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

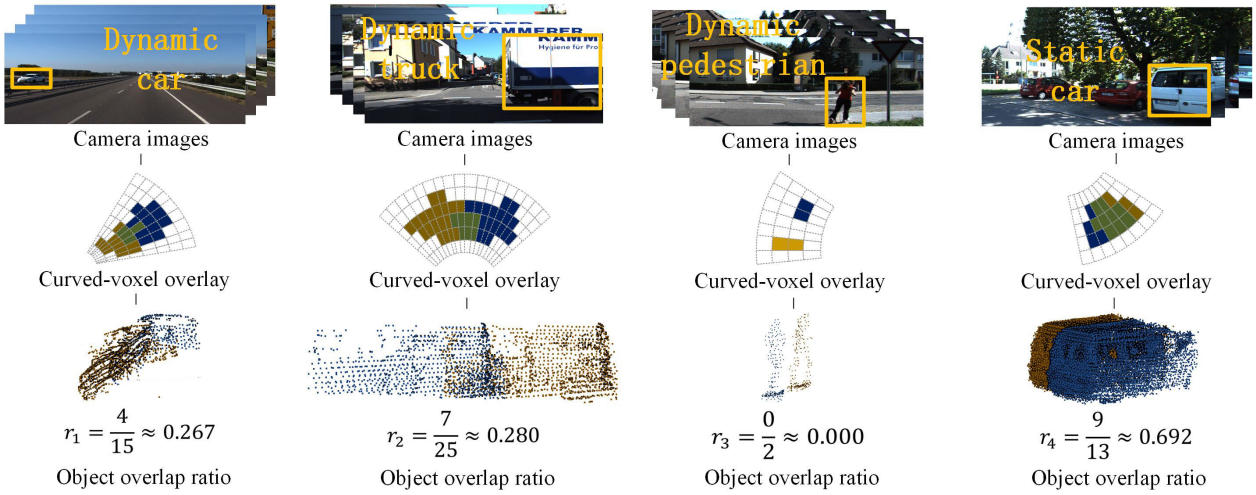


Fig. 8. Object tacking model in source $\mathcal{S}_t$ and target $\mathcal{S}_{t+1}$. The first row presents the related objects. The tracking object $\mathcal{B}^{\mathrm{PD}}_{\mathrm{id},t}$ (brown) is tracked in $\mathcal{S}_{t+1}$ using curved-voxel overlay to match the responding objects $\mathcal{B}^{\mathrm{ol}}_{\{k\},t+1}$ (blue). The green area corresponds to the overlaid curved voxels and the object overlap ratio $r^{\mathrm{PD}}_{\mathrm{id},t}$ is calculated by counting the overlaid curved-voxel. Finally, the motion property of $\mathcal{B}^{\mathrm{PD}}_{\mathrm{id},t}$ is determined with a ratio threshold $h_r = 0.5$. If $r^{\mathrm{PD}}_{\mathrm{id},t}$ is smaller than h_r, $\mathcal{B}^{\mathrm{PD}}_{\mathrm{id},t}$ is HD, and vice versa is LD.

where $\mathcal{B}^{\mathrm{PD},t+1}_{\mathrm{id},t}$ is the expression of tracking object $\mathcal{B}^{\mathrm{PD}}_{\mathrm{id},t}$ in $\mathcal{S}_{t+1}$ and $\odot$ denotes the curved-voxel overlay. The objects responding to the tracking in $\mathcal{S}_{t+1}$ enable to be stated as

$$\mathcal{B}^{\mathrm{ol}}_{\{k\},t+1} = \left\{ \mathcal{B}_{k,t+1} | k \in \mathcal{L}\left[v\left(\mathcal{B}^{\mathrm{PD},t+1}_{\mathrm{id},t} \right) \right] \right\} \tag{19}$$

where $v(\cdot)$ denotes the occupied curved voxels and $\mathcal{L}[\cdot]$ extracts the related segmented labels. Due to the voxel occupancy difference in $\mathcal{S}_t$ and $\mathcal{S}_{t+1}$, $\mathcal{B}^{\mathrm{ol}}_{\{k\},t+1}$ might contains more than one responding object.

B. Object Overlap Ratio

According to the overlaid curved voxels, the object overlap ratio quantitatively describes coverage degree by detecting curved-voxel occupancy changes of the tracking object $\mathcal{B}^{\mathrm{PD},t+1}_{\mathrm{id},t}$, which is defined as

$$r^{\mathrm{PD}}_{\mathrm{id},t} = \frac{v\left(\mathcal{B}^{\mathrm{PD},t+1}_{\mathrm{id},t} \right) \cap v\left(\mathcal{B}^{\mathrm{ol}}_{\{k\},t+1} \right)}{v\left(\mathcal{B}^{\mathrm{PD},t+1}_{\mathrm{id},t} \right)}. \tag{20}$$

Based on this definition, two cases with the ratio threshold h_r are considered as: 1) $r^{\mathrm{PD}}_{\mathrm{id},t} \leq h_r$ and 2) $r^{\mathrm{PD}}_{\mathrm{id},t} > h_r$. Of these cases, each $\mathcal{B}^{\mathrm{PD}}_{\mathrm{id},t}$ is categorized as an LD or HD object. Case 1) represents that $\mathcal{B}^{\mathrm{PD}}_{\mathrm{id},t}$ cannot be established a good connection in $\mathcal{S}_{t+1}$. Thus, $\mathcal{B}^{\mathrm{PD}}_{\mathrm{id},t}$ is an HD object required to be removed. Case 2) denotes that $\mathcal{B}^{\mathrm{PD}}_{\mathrm{id},t}$ can be tracked well in $\mathcal{S}_{t+1}$. Therefore, $\mathcal{B}^{\mathrm{PD}}_{\mathrm{id},t}$ is retained as an LD object to construct the static instance map.

C. Tight Coupling

The responding objects $\mathcal{B}^{\mathrm{ol}}_{\{k\},t+1}$ might contain objects that belong to different categories from $\mathcal{B}^{\mathrm{PD}}_{\mathrm{id},t}$, which can be caused by imperfect instances with incorrect shapes or labels in $\mathcal{S}_t$ and $\mathcal{S}_{t+1}$. To tackle this problem, the object responding score is defined as

$$s^{\mathrm{ol}}_{k,t+1} = \frac{v\left(\mathcal{B}^{\mathrm{PD},t+1}_{\mathrm{id},t} \right) \cap v\left(\mathcal{B}^{\mathrm{ol}}_{k,t+1} \right)}{v\left(\mathcal{B}^{\mathrm{ol}}_{k,t+1} \right)} \tag{21}$$

where $\mathcal{B}^{\mathrm{ol}}_{k,t+1}$ represents one of the responding objects.

According to that, the proposed method refines imperfect objects through shape and label correction. The incorrect shapes are rectified by combining objects with high responding scores, while the misclassified labels are corrected by updating the segmented labels in related curved voxels. The details are illustrated in Algorithm 2 (lines 11–24).

VI. Dynamic-Aware LiDAR Mapping

We develop an efficient LiDAR mapping module, which modifies the VGICP for SCV-OD to estimate initial LiDAR poses for object tracking and update local maps by considering dynamic objects in real-time on a single CPU.

A. Motion Estimation

Compared to the original VGICP [4], we modify the active nearest neighbor search method to a passive iteration process, reducing the computational burden in the registration process. Additionally, we utilize a hash table to manage the segmented label and motion property of each curved voxel.

Let the estimation of the transformation be $\mathbf{T}_{t+1,t}$, which aligns source $\mathcal{S}_t$ to target $\mathcal{S}_{t+1}$. We assume that the correspondences $\{SCV^t_m, SCV^{t+1}_n\}$ between $\mathcal{S}_t$ and $\mathcal{S}_{t+1}$ are given by kd-tree [32] searching for the central vertex of each SCV. Then, we model the correspondences from which a point was sampled as a Gaussian distribution: $a_m \sim \mathcal{N}(\hat{a_m}, C^t_m)$ and $b_n \sim \mathcal{N}(\hat{b_n}, C^{t+1}_n)$. Thus, the transformation error is defined as follows:

$$\hat{d_m} = \sum_n \left(\hat{b_n} - \mathbf{T}_{t+1,t} \hat{a_m} \right) \tag{22}$$

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2 Object Tracking With TC

Data: Source $\mathcal{S}_t$ and target $\mathcal{S}_{t+1}$
Result: low dynamic and actual static objects $\{\mathcal{B}_{id,t}^{LD}, \mathcal{B}_{id,t}^{AS}\}$

```
Let h_r be the overlapped ratio threshold;
Let h_s be the responding score threshold;
for B_{id,t}^{PD} do
    Get registered object B_{id,t}^{PD,t+1} through (18);
    Get responding objects B_{{k},t+1}^{ol} through (19);
    Calculate ratio r_{id,t}^{PD} through (20);
    if r_{id,t}^{PD} ≤ h_r then
        Remove B_{id,t}^{PD} as HD;
    else
        Retain B_{id,t}^{PD} as LD;
        ————-Tight Coupling————
        Initialize a new object B_{id⊕1,t+1}^{PD};
        for B_{k,t+1}^{ol} do
            Calculate score s_{k,t+1}^{ol} through (21);
            if s_{k,t+1}^{ol} < h_s then
                Continue;
            else
                if B_{k,t+1}^{ol} is PD then
                    Combine B_{k,t+1}^{ol} into B_{id⊕1,t+1}^{PD};
                else
                    Correct B_{id,t}^{PD} as AS;
                end
            end
        end
    end
end
Return {B_{id,t}^{LD}, B_{id,t}^{AS}};
```

where $\hat{b_n} - \mathbf{T}_{t+1,t}\hat{a_m}$ can be interpreted as smoothing the target point distributions. Then, the distribution of $\hat{d_m}$ is given by

$$\begin{aligned}\hat{d_m} &\sim \left(\mu^{d_m}, C^{d_m}\right)\\ \mu^{d_m} &= \sum_n \left(\hat{b_n} - \mathbf{T}_{t+1,t}\hat{a_m}\right) = 0\\ C^{d_m} &= \sum_n \left(C_n^{t+1} + \mathbf{T}_{t+1,t} C_m^t \mathbf{T}_{t+1,t}^T\right).\end{aligned} \tag{23}$$

Therefore, the transformation $\mathbf{T}_{t+1,t}$ is estimated by maximizing the $log(\cdot)$ likelihood of $\hat{d_m}$ as follows:

$$\begin{aligned}\mathbf{T}_{t+1,t} &= \arg\min_{\mathbf{T}_{t+1,t}} \sum_m \left(N_m \tilde{d_m}^T \tilde{C_m}^{-1} \tilde{d_m}\right)\\ \tilde{d_m} &= \frac{\sum_n b_n}{N_m} - \mathbf{T}_{t+1,t} a_m\\ \tilde{C_m} &= \frac{\sum_n C_n^{t+1}}{N_m} + \mathbf{T}_{t+1,t} C_m^t \mathbf{T}_{t+1,t}^T\end{aligned} \tag{24}$$

where N_m is the number of points in neighbor SCV_t. The estimated transformation $\mathbf{T}_{t+1,t}$ can be efficiently computed by substituting the mean of the distributions of the points b_n around a_m and be weighted by N_m. The details are presented in Algorithm 3 (lines 3–16).

B. Local Updating

Given the estimated motions $\mathbf{T}_{t+1,t}$ and static point clouds resulting from the dynamic removal module, we update the LiDAR poses and merge the point clouds into global maps.

Utilizing the estimated poses $\mathbf{T}_{t+1,t}$ as initial values, the correspondences $\{\mathrm{SCV}_m^t, \mathrm{SCV}_n^{t+1}\}$ between $\mathcal{S}_t$ and $\mathcal{S}_{t+1}$ enable us to distinguish dynamic curved voxels from static ones after *object tracking* (see Section V). Then, the dynamic-aware transformation error d'_m is defined as

$$d'_m = \sum_n \left(b'_n - \mathbf{T}_{t+1,t} a'_m\right). \tag{25}$$

We estimate the optimized LiDAR pose $\mathbf{T}_{t+1,t}^*$ by maximizing the log likelihood of d'_m as follows:

$$\mathbf{T}_{t+1,t}^* = \arg\min_{\mathbf{T}} \sum_m d_m'^T \left(\sum_n \left(C_n^{t+1} + \mathbf{T} C_m^t \mathbf{T}^T\right)\right)^{-1} d'_m \tag{26}$$

where $\mathbf{T}$ represents the initial motion $\mathbf{T}_{t+1,t}$. Finally, the resulting point clouds are merged into the local maps using the optimized LiDAR poses $\mathbf{T}_{t+1,t}^*$. The details can be found in Algorithm 3 (lines 17–28).

VII. Experimental Results

Employing KITTI [26] and the custom dataset, we compared the proposed method with state-of-the-art (SOTA) LiDAR-based methods in dynamic removal and pose estimation. For all experiments, we set the parameters to proposed SLAM (Base w/*): the maximum radial distance $R_{\max} = 50$ m, the boundary of azimuth angles $[A_{\min}, A_{\max}] = [-30°, 60°]$, the size of curved voxel $[\Delta\rho, \Delta\sigma, \Delta\varphi] = [0.20\text{ m}, 2.0°, 2.0°]$, the object overlap ratio $h_r = 0.5$, and the working modules "RPC" and "TC," as "Base w/ RPC + TC."

A. Dataset

1) KITTI Dataset: The KITTI dataset [26] provides LiDAR scan with 64 laser beams (Velodyne HDL-64E) with an FOV of 26.9° in vertical and 360° in horizontal. The SemanticKITTI dataset [33] provides pointwise ground-truth labels synchronized with the KITTI dataset and associated LiDAR SLAM-based poses. We manually selected the frames from the KITTI dataset with the maximum number of dynamic objects to quantitatively evaluate the proposed algorithms. Therefore, Seq 00 (4320–4530), Seq 01 (120–270), Seq 02 (820–980), Seq 05 (2350–2670), and Seq 07 (650–840) were chosen as our static map construction benchmark where the numbers in parenthesis indicate the start and end frames.

2) Our Datasets: As shown in Fig. 4, our sensor system is equipped with Livox Mid360, generating LiDAR scan with an FOV of 59°×360°. Fig. 9 shows that our custom datasets contain scenarios like schools, highways, and urban streets. We manually selected five typical sequences, namely School 1 (SH 1), Highway 1 (HG 1), Highway 2 (HG 2), Urban1 (UB 1), and Urban 2 (UB 2), with a length of 200 frames as the benchmark and labeled the dynamic objects utilizing the point cloud labeling tool [33].

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

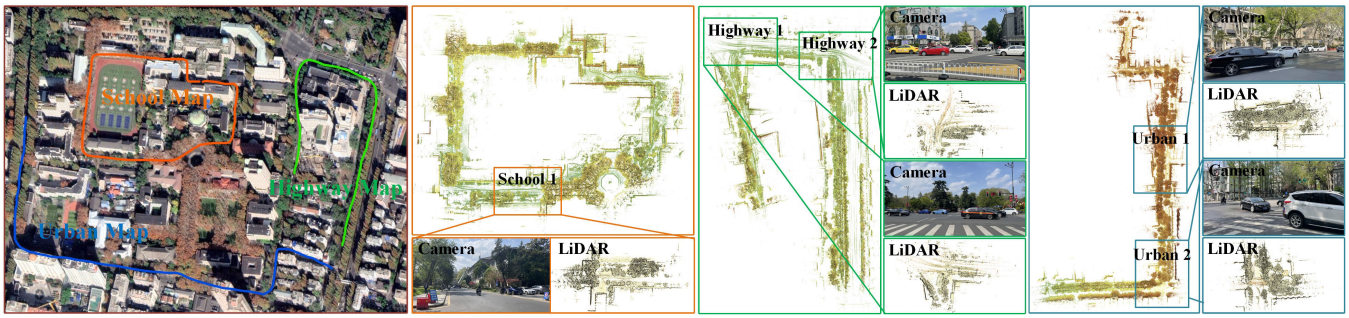


Fig. 9. Custom dataset using Livox Mid360. (a) Our dataset comprises different sequences. (b) School: building complexes with many pedestrians and bicyclists. (c) Highway: traveling cars at high speed. (d) Urban: various road types.

Algorithm 3 VGICP-Based LiDAR Mapping

Data: Source $\mathcal{S}_t$ and target $\mathcal{S}_{t+1}$
Result: LiDAR pose $\mathbf{T}^*_{t+1,t}$
Let e_i be the i-th residual;
Let J_i be the i-th Jacobi matrix;
————-**Motion Estimation**————
Extract central-point Cloud$(\cdot)$ of $\mathcal{S}_t$ and $\mathcal{S}_{t+1}$;
$\mathbf{T}_{t+1,t} \leftarrow$ ICP(Cloud($\mathcal{S}_t$), Cloud($\mathcal{S}_{t+1}$));
while $\boldsymbol{T}_{t+1,t}$ *is not converged* **do**
  $e \leftarrow [\ ]$, $J \leftarrow [\ ]$;
  Correspondences $\{\mathrm{SCV}^m_t, \mathrm{SCV}^n_{t+1}\} \leftarrow$ Kd-Tree(Cloud($\mathcal{S}_t$), Cloud($\mathcal{S}_{t+1}$));
  Calculate covariances $C^t = \{C^t_0, \ldots, C^t_M\}$ and $C^{t+1} = \{C^{t+1}_0, \ldots, C^{t+1}_N\}$;
  for $i \in 0, \ldots, M$ **do**
    $e_i, J_i \leftarrow \mathrm{Cost}(\mathbf{T}_{t+1,t}, C^t_i, C^{t+1}_n)$;
    $e \leftarrow e \cup e_i$, $J \leftarrow J \cup J_i$;
  end
end
$\delta\mathbf{T} \leftarrow J^T J^{-1} J^T e$ // Gauss-Newton update;
$\mathbf{T}_{t+1,t} \leftarrow \mathbf{T}_{t+1,t} \otimes \delta\mathbf{T}$;
————-**Local Updating**————
$\mathbf{T}^*_{t+1,t} \leftarrow Unit$;
for $i \in 0, \ldots, M$ **do**
  if SCV^i_t *is covered by dynamic object* **then**
    Continues;
  end
  else
    $e \leftarrow e \cup e_i$, $J \leftarrow J \cup J_i$;
    $\delta\mathbf{T} \leftarrow J^T J^{-1} J^T e$;
    $\mathbf{T}^*_{t+1,t} \leftarrow \mathbf{T}^*_{t+1,t} \otimes \delta\mathbf{T}$;
  end
end
Return $\mathbf{T}^*_{t+1,t}$;

B. Metrics

1) Dynamic Removal: The voxelwise and map-oriented quantitative metrics called *preservation rate* (PR), *rejection rate* (RR), and *F1 Score* [22] are defined as follows.

1) *PR:* # of static points retained/ # of total static points in the raw map.
2) *RR:* 1 − (# of dynamic points removed/ # of total dynamic points in the raw map).
3) *F1 Score:* $(\mathrm{PR} * \mathrm{RR} * 2)/(\mathrm{PR} + \mathrm{RR})$.

Here, PR and RR are calculated voxelwise, denoting the proportion of static points retained and dynamic points removed, respectively. F1 Score represents the overall performance considering both PR and RR. We apply identical voxelization with a voxel size of 0.2 m for static maps retrieved from the baseline models for fair comparison and # represents the number of relating voxels.

We also utilize intersection-over-union (IoU) [34] as the pointwise dynamic detection metric

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{27}$$

where TP, FP, and FN represent true positive, false positive, and false negative predictions of dynamic points, respectively.

2) Pose Estimation: We utilize the average translation trel(%) and rotation rrel(°/100 m) root mean square error (RMSE) drift for all possible subsequences of length [19], described as

$$t_{\mathrm{rel}}(\mathcal{F}) = \frac{1}{|\mathcal{F}|} \sum_{(i,j)\in\mathcal{F}} \left|\left|\left(\hat{\zeta}_j \ominus \hat{\zeta}_i\right) \ominus \left(\zeta_j \ominus \zeta_i\right)\right|\right|_2$$
$$r_{\mathrm{rel}}(\mathcal{F}) = \frac{1}{|\mathcal{F}|} \sum_{(i,j)\in\mathcal{F}} \angle\left[\left(\hat{\zeta}_j \ominus \hat{\zeta}_i\right) \ominus \left(\zeta_j \ominus \zeta_i\right)\right] \tag{28}$$

where $\mathcal{F}$ is a set of frames (i, j), $\hat{\zeta} \in \mathrm{SE}(3)$, and $\zeta \in \mathrm{SE}(3)$ are the estimated and ground-truth poses, respectively, $\ominus$ denotes the inverse compositional operator, and $\angle[\cdot]$ is the rotation angle.

C. Evaluation on the KITTI Dataset

1) Dynamic Removal: The proposed method was compared with SOTA methods, namely OctoMap [20] with voxel sizes 0.2, PeopleRemover [21], Removert [17] with various remove and revert stages, ERASOR [22] with the size of scan ratio threshold 0.2, Park et al. [19], Arora et al. [35], and Zhang et al. [36].

Table I presents the quantitative results on the KITTI dataset [26]. The proposed method ranks first in PR and F1 scores. Though the SOTA methods are capable of removing most dynamic data points, they often incorrectly remove a

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON WITH SOTA METHODS ON THE KITTI DATASET

Seq.	Method	PR[%]	RR[%]	F1 Score
00	OctoMap - 0.2 [20]	34.568	**99.979**	0.514
	PeopleRemover [21]	37.523	89.116	0.528
	Removert - RM3 [17]	85.502	99.354	0.919
	Removert - RM3+RV1 [17]	86.829	90.617	0.887
	ERASOR - 0.2 [22]	93.980	97.081	0.955
	Park *et al.* [19]	94.050	95.242	0.956
	Arora *et al.* [35]	77.840	93.200	0.848
	Zhang *et al.* [36]	93.061	98.670	0.958
	Proposed (Ours)	**98.621**	95.544	**0.970**
01	OctoMap - 0.2 [20]	20.777	**99.863**	0.344
	PeopleRemover [21]	36.349	93.116	0.523
	Removert - RM3 [17]	94.221	93.608	0.939
	Removert - RM3+RV1 [17]	95.815	57.077	0.715
	ERASOR - 0.2 [22]	91.487	95.383	0.934
	Park *et al.* [19]	91.815	94.096	0.929
	Arora *et al.* [35]	77.410	88.270	0.825
	Zhang *et al.* [36]	89.352	93.652	0.914
	Proposed (Ours)	**94.388**	94.257	**0.943**
02	OctoMap - 0.2 [20]	23.746	**99.792**	0.384
	PeopleRemover [21]	29.037	94.527	0.444
	Removert - RM3 [17]	76.319	96.799	0.853
	Removert - RM3+RV1 [17]	83.293	88.371	0.858
	ERASOR - 0.2 [22]	87.731	97.008	0.921
	Park *et al.* [19]	91.208	95.510	0.933
	Arora *et al.* [35]	77.910	96.850	0.864
	Zhang *et al.* [36]	90.285	94.563	0.924
	Proposed (Ours)	**98.521**	96.127	**0.973**
05	OctoMap - 0.2 [20]	33.904	**99.882**	0.506
	PeopleRemover [21]	38.495	90.631	0.540
	Removert - RM3 [17]	86.900	87.880	0.874
	Removert - RM3+RV1 [17]	88.170	79.981	0.839
	ERASOR - 0.2 [22]	88.730	98.262	0.933
	Park *et al.* [19]	93.820	95.740	0.947
	Arora *et al.* [35]	78.230	94.670	0.857
	Zhang *et al.* [36]	93.540	92.480	0.930
	Proposed (Ours)	**98.972**	96.674	**0.978**
07	OctoMap - 0.2 [20]	38.183	**99.565**	0.552
	PeopleRemover [21]	34.772	91.983	0.505
	Removert - RM3 [17]	80.689	98.822	0.888
	Removert - RM3+RV1 [17]	82.038	95.504	0.883
	ERASOR - 0.2 [22]	90.624	99.271	0.948
	Park *et al.* [19]	91.146	97.284	0.941
	Arora *et al.* [35]	79.740	91.960	0.854
	Zhang *et al.* [36]	91.661	96.256	0.939
	Proposed (Ours)	**94.747**	98.666	**0.967**

large number of static points. The high resolution of cubes in Octomap [20] causes a high-sensitive detection to the change of occupancy, which leads to the top-ranking performance in RR, but it also results in the terrible PR and F1 Score. To address this problem, Arora et al. [35] designed a K-nearest neighbors algorithm (KNN)-based voting method to recover false dynamic points in Octomap, resulting in a higher PR but a lower RR. Removert [17] despite being proficient in erasing dynamic points during the removal stage, fails to supplement the false detection through its revert stage. It can be seen that the results with the remove stage (RM3) have a much higher RR but a minimal difference in PR compared with the result with the revert stage (RM3+RV1). Park et al. [19] proposed false detection suppression to solve the false dynamic points resulting from range image-based methods, thus compared to Removert, this method resulted in a higher PR. ERASOR [22] erases most dynamic objects through the scan ratio test (SRT), but the regionwise pseudo-occupancy descriptor (R-POD) considers occupancy changes only in terms of the spatial distribution of points. In contrast, our proposed method removes dynamic points at the object level using curved voxels as occupancy elements, resulting in the highest PR and F1 score while obtaining a high RR. The reason why the proposed method does not outperform others (i.e., Octomap, ERASOR) in RR is that our method is easily affected by the precision loss in the object segmentation module. For instance, the bottom of moving objects are incorrectly extracted as ground points and the objects far from LiDAR are difficult to be clustered together. Octomap and ERASOR do not implement object segmentation, but the addition of object segmentation in our method has improved the robustness and accuracy of dynamic removal. This helps us achieve a better balance between PR and RR, resulting in the best F1 Score.

Fig. 10 represents the qualitative comparison results on the KITTI dataset [26]. We compare our proposed method with open-source and representative studies. In Fig. 10, estimated static points and estimated dynamic points are colored in gray and red. Removert [17] strictly deletes dynamic points in the removal stage but incorrectly removes a lot of AS points on vegetation and parked vehicles. ERASOR [22] estimates dynamic vertical bins with changing heights. Though assuming an accurate and precise prior map, objects with uncertain heights such as trees and buildings are easily misclassified. In contrast, the proposed method performs well in dynamic removal and static retaining. Our approach only implements object tracking on PD objects according to the prior knowledge provided by the object segmentation, which prevents static objects from being wrongly identified as dynamic ones. Fig. 10 show that there are few static points being wrongly erased and most LD objects (e.g., parked cars) are tracked well in the same color.

2) Pose Estimation: We evaluated the trajectory results from the local mapping modules by comparing with existing LiDAR odometry approaches, namely MULLS [3], LOAM [5], SuMa [7], T-LOAM [8], LiTAMIN2 [37], Park et al. [19], Wang et al. [38], and Chang et al. [39]. The results of each method in Table II directly refer to the corresponding papers.

Table II shows the comparison results of the average translational and rotational errors, showing that the trajectory estimation is dependent on the dynamic removal process. The proposed method with the dynamic removal presents an error of 0.68% in translation and 0.25°/100 m in rotation, which reduces the translational and rotational errors by 0.39% and 0.07°/100 m. Although the error performance is slightly lower than that of MULLS [3] using a multimetric linear least-square optimization, our proposed method overcomes the LiDAR motion ambiguity while also adapting various data formats. SuMa [7] utilizes the semantic information obtained from a deep neural network to achieve pose estimation and loop closure. However, it works without considering dynamic objects effectively and thus results in a worse error performance. LOAM [5] and T-LOAM [8] extract features to estimate the poses, but the presence of dynamic objects makes the stable

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

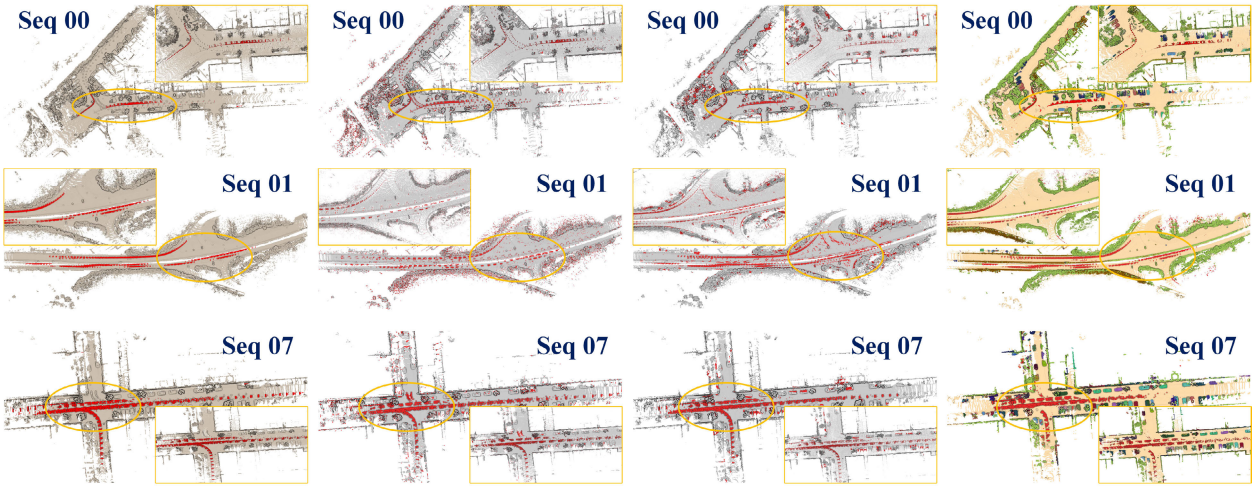


Fig. 10. Dynamic removal results of Seq 00, 01, and 07 on the KITTI dataset [26]. (a) Ground truth, (b) Removert [17], (c) ERASOR [22], and (d) our proposed method. The ground-truth classifies dynamic and static objects in brown and red. For the resulting images, the red point clouds are determined as dynamic objects, and the gray point clouds are determined as static objects.

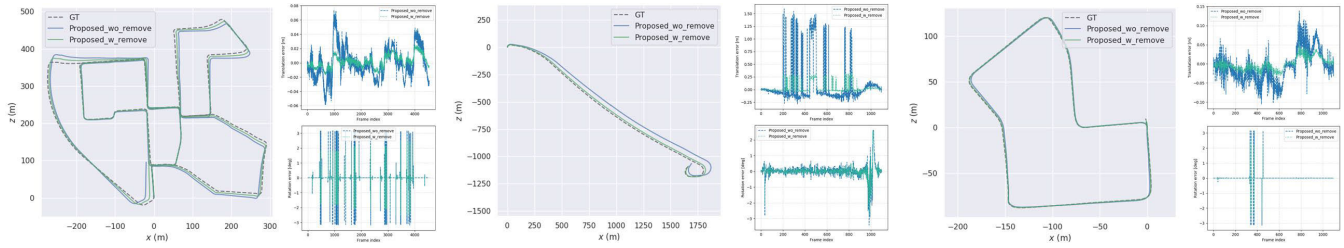


Fig. 11. (Left to right) Trajectory results of the KITTI benchmark [26] with Seq 00, 01, and 07. Each trajectory presents the ground truth (gray), the proposed method without dynamic removal (blue), and the proposed method with dynamic removal (green).

TABLE II
COMPARISON OF AVERAGE TRANSLATIONAL AND ROTATIONAL ERRORS ON THE KITTI DATASET

Method	Seq. 00	Seq. 01	Seq. 02	Seq. 03	Seq. 04	Seq. 05	Seq. 06	Seq. 07	Seq. 08	Seq. 09	Seq. 10	Avg.
Proposed (w/o remove)	0.72/0.22	4.66/0.75	0.66/0.18	1.25/0.67	0.48/0.15	0.50/0.33	0.40/0.21	0.35/0.19	1.01/0.30	0.61/0.29	1.12/0.27	1.07/0.32
Proposed (w/ remove)	0.61/0.22	1.02/0.22	0.66/0.17	1.02/0.55	0.44/0.12	0.47/0.29	0.39/0.21	0.32/0.15	0.96/0.27	0.60/0.29	0.96/0.30	0.68/0.25
MULLS [3]	0.54/0.13	0.62/0.09	0.69/0.13	0.61/0.22	0.35/0.08	0.29/0.07	0.29/0.08	0.27/0.11	0.83/0.17	0.51/0.12	0.61/0.19	**0.49/0.13**
LOAM [5]	0.78/-	1.43/-	0.92/-	0.86/-	0.71/-	0.57/-	0.65/-	0.63/-	1.12/-	0.77/-	0.79/-	0.84/-
SuMa [7]	0.68/0.23	1.70/0.54	1.20/0.48	0.74/0.50	0.44/0.27	0.43/0.20	0.54/0.30	0.74/0.54	1.20/0.38	0.62/0.22	0.72/0.32	0.83/0.36
T-LOAM [8]	0.98/0.60	2.09/0.52	1.01/0.39	1.10/0.82	0.68/0.68	0.55/0.32	0.56/0.31	0.50/0.47	0.94/0.33	0.80/0.40	1.12/0.61	0.93/0.49
LiTAMIN2 [37]	0.70/0.28	2.10/0.46	1.00/0.37	0.67/0.46	0.3/0.26	0.40/0.20	0.46/0.21	0.34/0.19	1.10/0.35	0.47/0.23	0.66/0.28	0.70/0.29
Park *et al.* [19]	0.60/0.26	1.02/0.18	0.66/0.19	0.92/0.48	0.43/0.14	0.40/0.21	0.43/0.21	0.33/0.18	0.95/0.31	0.52/0.20	0.94/0.25	0.65/0.24
Wang *et al.* [38]	0.83/0.33	0.55/0.21	0.71/0.25	0.49/0.38	0.22/0.11	0.34/0.21	0.36/0.24	0.46/0.38	1.14/0.41	0.78/0.33	0.80/0.46	0.80/0.40
Chang *et al.* [39]	0.59/0.19	0.80/0.21	0.55/0.16	0.60/0.23	0.35/0.12	0.39/0.21	0.37/0.16	0.28/0.17	0.84/0.23	0.47/0.10	0.50/0.17	0.51/0.18

and effective features insufficient. Park et al. [19] developed a dynamic-aware and range image-based SLAM to remove dynamic objects and update local poses, but it still encounters the resolution setting of the range image, and its precision is heavily affected by the false detection. Wang et al. [38] proposed an end-to-end framework for LiDAR posed recovery in real-time, but it cannot perform well in environmental generalization. Chang et al. [39] designed a novel strategy using multiple weights to reduce the feature extraction errors, but it still struggles with the geometric distribution of points cloud where there are moving objects. Fig. 11 depicts the qualitative results of the KITTI benchmark [26]. As shown in Fig. 11, the proposed method with dynamic removal results in a more accurate trajectory and a stable six-DoF estimation.

D. Evaluation on Our Dataset

1) Dynamic Removal: Table III shows that our proposed method results in the best PR and F1 score. Additionally, Fig. 12 represents the detailed evaluation. Removert [17] removes most dynamic objects through the remove stage, but a lot of AS points on trees and buildings are incorrectly erased. ERASOR [22] can retain more AS points, but it cannot completely remove dynamic objects. Because it erases over-ground

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

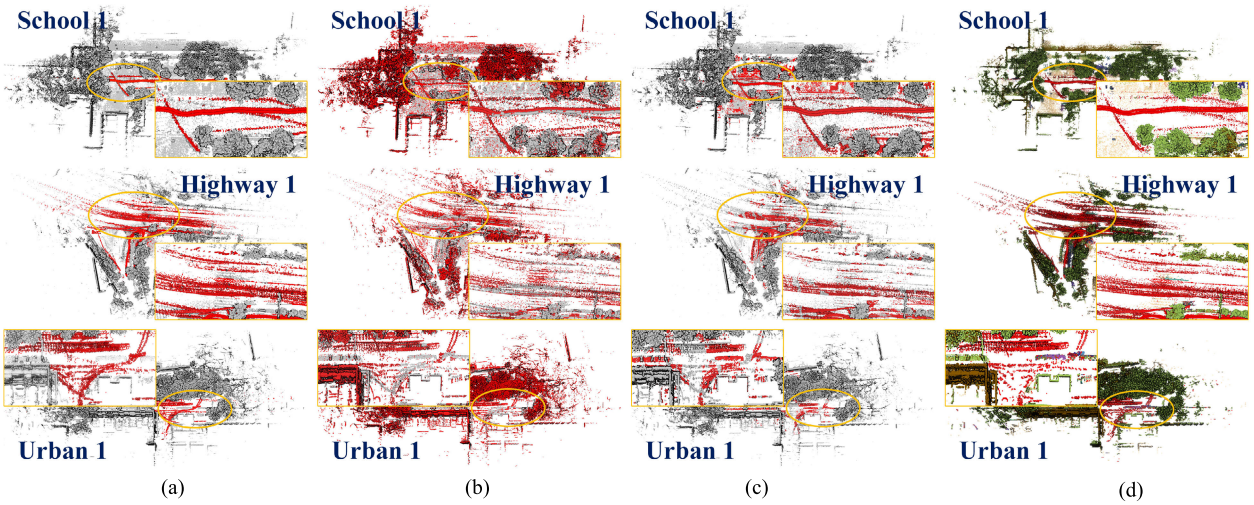


Fig. 12. Comparison results of dynamic removal on School 1, Highway 1, and Urban 1. (a) Ground truth, (b) Removert [17], (c) ERASOR [22], and (d) our proposed methods. The gray point clouds indicate static objects and the red point clouds indicate dynamic objects.

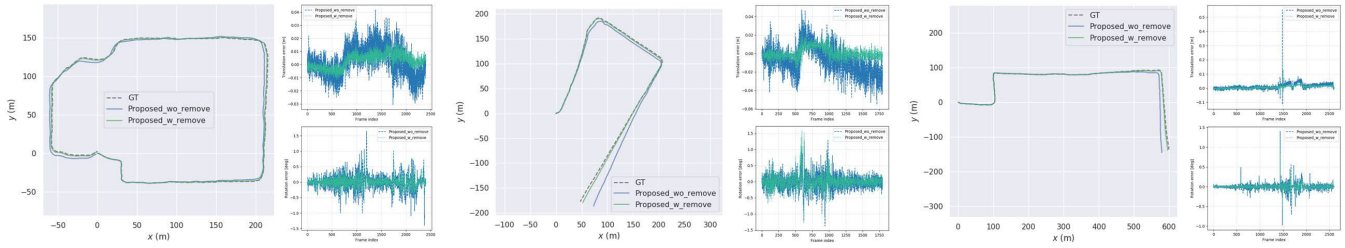


Fig. 13. (Left to right) Trajectory results of the custom dataset including School, Highway, and Urban. The ground truth (gray), the proposed method without dynamic removal (blue), and the proposed method with dynamic removal (green) are painted.

TABLE III
COMPARISON WITH SOTA METHODS ON THE CUSTOM DATASET

Seq.	Method	PR[%]	RR[%]	F1 Score
SH 1	Removert - RM3 [17]	61.489	86.117	0.717
	ERASOR - 0.2 [22]	83.660	80.265	0.819
	Proposed (Ours)	**94.471**	**91.836**	**0.931**
HG 1	Removert - RM3 [17]	62.010	84.259	0.714
	ERASOR - 0.2 [22]	80.928	88.103	0.844
	Proposed (Ours)	**92.123**	**94.771**	**0.934**
HG 2	Removert - RM3 [17]	58.209	**92.689**	0.715
	ERASOR - 0.2 [22]	84.630	90.481	0.875
	Proposed (Ours)	**92.152**	91.619	**0.919**
UB 1	Removert - RM3 [17]	63.669	91.125	0.750
	ERASOR - 0.2 [22]	86.551	91.423	0.889
	Proposed (Ours)	**89.972**	**93.674**	**0.918**
UB 2	Removert - RM3 [17]	64.174	92.667	0.758
	ERASOR - 0.2 [22]	91.624	93.865	0.927
	Proposed (Ours)	**92.747**	**94.666**	**0.937**

points in the bins that contain PD objects, our dataset with sparse ground points will cause incorrect ground estimation. Compared with these methods, our proposed method can perform better in dynamic object removal and static mapping in naturally dynamic environments.

2) Pose Estimation: As shown in Fig. 4, our system is equipped with Livox Mid360 but without INS or GPS measurements. To obtain the ground truth of the custom dataset, we believe that the trajectory resulting from the Fast-Lio2 [14] is sufficiently reliable as the ground truth. Considering solid-state LiDAR (Livox Mid360), we evaluated the proposed method comparing with LOAM-Livox [6] and Lio-Livox.

TABLE IV
COMPARISON OF ABSOLUTE TRAJECTORY ERRORS ON THE CUSTOM DATASET

Method	School	Highway	Urban
Proposed (w/o remove)	7.24	18.79	13.36
Proposed (w/ remove)	4.02	9.13	7.11
LOAM-Livox [6]	6.50	-	14.93
Lio-Livox	**1.25**	**2.61**	**2.11**

Table IV shows the comparison results of absolute trajectory errors on the custom dataset. The error performances present that the dynamic removal can effectively optimize the trajectory estimation. Due to the complex environments with various dynamic objects, it is a challenge for LOAM-Livox [6] to recover the accurate trajectory by extracting hand-crafted features (i.e., edge and plane points). The IMU-incorporated Lio-Livox produces the best performance because the IMU preintegration module provides precise prior poses without considering dynamic objects. Fig. 13 shows that dynamic removal is necessary for LiDAR odometry to estimate stable and accurate poses in LiDAR mapping.

E. Evaluation on Dynamic Detection

We evaluate the performance of our method and compare it with several baseline methods, including ERASOR [22], LMNet [23], and 4DMOS [40]. Fig. 14 shows the comparison

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

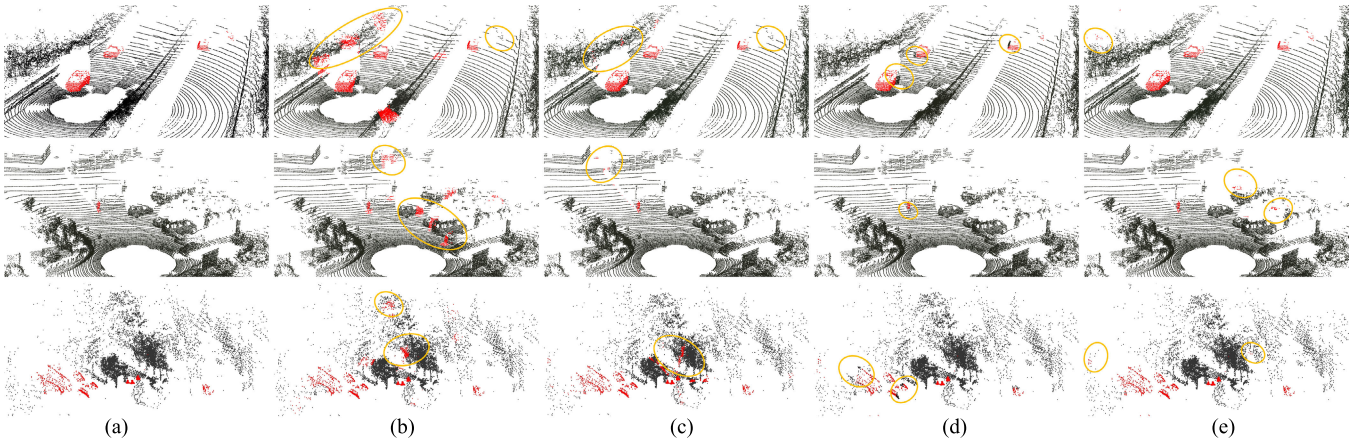


Fig. 14. Comparison results of dynamic object detection on KITTI [26] (the first two lines) and our custom dataset (the third line). (a) Ground truth, (b) ERASOR [22], (c) LMNet [23], (d) 4DMOS [40], and (e) our proposed methods. The gray and red points indicate the predicted static and dynamic objects. The orange circles mark the incorrectly detected point clouds.

TABLE V
PERFORMANCE COMPARISON ON DYNAMIC OBJECT DETECTION ON KITTI AND OUR DATASET

Seq.	Method	PR[%]	RR[%]	F1 Score	IoU[%]
00	ERASOR [22]	93.980	**97.081**	0.955	52.6
	LMNet [23]	94.562	94.107	0.943	63.1
	MotionSeg3D [24]	96.834	94.119	0.955	67.7
	4DMOS [40]	98.618	92.364	0.954	69.2
	Proposed (Ours)	**98.621**	95.544	**0.970**	**69.7**
HG 1	ERASOR [22]	62.010	84.259	0.714	42.2
	LMNet [23]	89.256	92.661	0.909	57.6
	MotionSeg3D [24]	88.230	91.594	0.899	55.1
	4DMOS [40]	91.987	92.889	0.924	61.5
	Proposed (Ours)	**92.123**	**94.771**	**0.934**	**62.7**

results of dynamic object detection on KITTI [26] and our custom dataset. ERASOR could remove most dynamic objects but it still generates plenty of false dynamic points on static objects (e.g., trees and parked cars). LMNet is the first learning-based work on LiDAR MOS exploiting range images, which is fast but results in many wrong predictions. However, LMNet exhibits blurry edges in the detection of dynamic objects in the range image, which leads to incorrect detections in the reprojected point cloud map. 4DMOS performs well on fast-MOS, but not as well on slow-MOS. As 4DMOS cannot capture the instance information of the moving points, only partial points of the moving instance (e.g., driving cars) can be correctly predicted. Table V shows the quantitative comparison results. Our proposed method ranks first in PR, F1 Score, and IoU on the KITTI dataset. While our method is lower than ERASOR in terms of RR, it performs better in all the other aspects. On our custom dataset, our method outperforms the other methods, achieving top-ranking PR, RR, F1 Score, and IoU, which indicates that our method demonstrates stronger transferability and capability of generalization in real-world environments. Moreover, compared to the learning-based method (i.e., LMNet, MotionSeg, and 4DMOS), our method operates on a single CPU without GPU acceleration, and the single-frame dynamic detection takes approximately 180 ms.

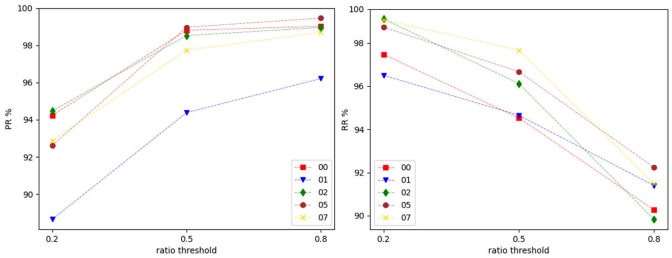


Fig. 15. Dynamic removal performance changes based on the ratio threshold h_r. The higher the PR and RR, the better.

F. Ablation Study

1) Size of Curved Voxels: Table VI illustrates the impact of curved-voxel resolution. We set different parameters of $[\Delta\rho, \Delta\sigma, \Delta\varphi]$ to test the performance of object segmentation, dynamic removal, pose estimation, and run time. These hyperparameters are related to the resolution of curved voxels, where a higher resolution leads to higher precise clustering and increased sensitivity in detecting changes in occupancy.

2) Threshold of Object Overlap Ratio: As shown in Fig. 15, increasing h_r leads to an improvement in the PR because it allows more static points to be retained. In contrast, higher h_r also allows dynamic points to be preserved because their related objects can be tracked well in consecutive scans, which yields a lower RR. In most sequences, PR shows sharper increments from 0.2 to 0.5, whereas RR shows sharper decrements from 0.5 to 0.8. Therefore, we can conclude that the h_r value of 0.5 yields the most reasonable dynamic object removal performance throughout the experiment.

3) Modules for Dynamic Removal: We mainly implement ablation experiments to compare the performance of the two modules, namely RPC which denotes the RPC, and TC that represents the tight coupling manner. Table VII illustrates that RPC and TC can improve the PR, RR, and F1 score values. Because the RPC improves object segmentation accuracy by incorporating LiDAR intensity and the TC refines imperfect objects via shape and label correction.

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
ABLATION WITH SIZE OF CURVED VOXELS [$\Delta\rho$, $\Delta\sigma$, $\Delta\varphi$] ON THE KITTI AND OUR DATASET

Seq.	[$\Delta\rho$, $\Delta\sigma$, $\Delta\varphi$] ↑	IoU[%] AS ↑	IoU[%] PD ↓	Dynamic Removal PR [%] ↑	Dynamic Removal RR [%] ↓	Dynamic Removal F1 Score	Pose Estimation t_{rel} [%] ↑	Pose Estimation r_{rel} [°/100m] ↑	Time [ms/fps] ↓
02	[0.15m, 1.0°, 2.0°]	62.49	**97.73**	95.110	**96.994**	0.951	**0.60**	**0.15**	277.06
	[0.20m, 2.0°, 2.0°]	68.63	96.62	98.521	96.127	**0.973**	0.66	0.17	236.41
	[0.25m, 3.0°, 2.0°]	**70.20**	92.11	**98.666**	92.447	0.955	0.68	0.18	**201.88**
07	[0.15m, 1.0°, 2.0°]	60.17	**97.22**	90.131	**98.841**	0.943	**0.31**	**0.12**	269.01
	[0.20m, 2.0°, 2.0°]	67.07	96.11	94.747	98.666	**0.967**	0.32	0.15	219.24
	[0.25m, 3.0°, 2.0°]	**70.52**	91.03	**95.010**	93.787	0.944	0.35	0.15	**184.61**
HG 1	[0.15m, 1.0°, 2.0°]	55.57	**91.49**	88.616	**95.439**	0.919	**5.11**	**2.20**	200.73
	[0.20m, 2.0°, 2.0°]	59.31	89.10	90.125	94.771	**0.924**	5.26	2.26	187.19
	[0.25m, 3.0°, 2.0°]	**62.11**	83.77	**92.633**	91.710	0.922	5.38	2.31	**169.50**
UB 1	[0.15m, 1.0°, 2.0°]	60.31	**87.47**	87.364	**94.113**	0.906	**3.96**	**2.34**	226.76
	[0.20m, 2.0°, 2.0°]	62.99	85.36	89.972	93.674	**0.918**	4.02	2.41	192.36
	[0.25m, 3.0°, 2.0°]	**64.06**	82.12	**91.001**	90.176	0.906	4.10	2.47	**185.70**

TABLE VII
DYNAMIC REMOVAL WITH DIFFERENT MODULES ON THE KITTI AND OUR DATASET

Seq.	Method	PR[%]	RR[%]	F1 Score
02	Base	92.278	89.927	0.911
	Base w/ RPC	94.132	92.983	0.936
	Base w/ TC	96.098	90.481	0.932
	Base w/ RPC+TC	**98.521**	**96.127**	**0.973**
HG 1	Base	85.230	89.031	0.871
	Base w/ RPC	87.111	89.966	0.885
	Base w/ TC	88.689	90.325	0.895
	Base w/ RPC+TC	**90.125**	**94.771**	**0.924**

TABLE VIII
DYNAMIC REMOVAL WITH DIFFERENT DESCRIPTORS ON THE KITTI AND OUR DATASET

Seq.	Descriptor	PR[%]	RR[%]	F1 Score
02	R-POD [22]	87.731	**97.008**	0.921
	R-GPOD [41]	73.500	86.200	0.793
	SCV-OD (Ours)	**98.521**	96.127	**0.973**
HG 1	R-POD [22]	80.928	88.103	0.544
	R-GPOD [41]	66.518	84.310	0.744
	SCV-OD (Ours)	**90.125**	**94.771**	**0.924**

4) Descriptors for Dynamic Removal: We test different descriptors, namely the proposed SCV-OD, R-POD, and *regionwise ground pseudo-occupancy descriptor* (R-GPOD) [41], in dynamic removal. R-GPOD inspired by R-POD incorporates the z-directional information of the ground points into the z-directional difference. Table VIII and Fig. 16 shows that our proposed SCV-OD outperforms the other two descriptors. That is because our proposed SCV-OD not only focuses on voxel occupancy changes, but also employs object-level concepts in the dynamic detection process, allowing it to remove more dynamic objects while retaining more static ones.

G. Evaluation Real-Time Performance

Fig. 17 displays the real-time performance impacts of the five processes on the proposed SLAM framework. Specifically, motion estimation and object segmentation modules account for a significant portion of the system's optimization time. The proposed LiDAR odometry and mapping algorithm requires approximately 200 ms per frame (5 frames/s), efficiently adapting for real-time sensing in low-speed scenes.

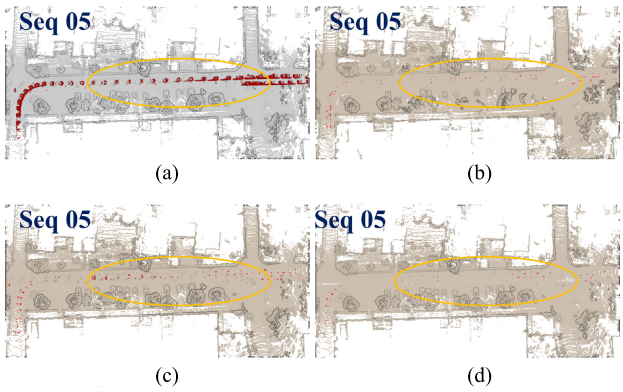


Fig. 16. Maps constructed by different descriptors on Seq 05 of the KITTI [26]. The red and gray points are dynamic and static objects, respectively. (a) Original Map. (b) w/ R-POD. (c) w/ R-GPOD. (d) w/ SCV-OD.

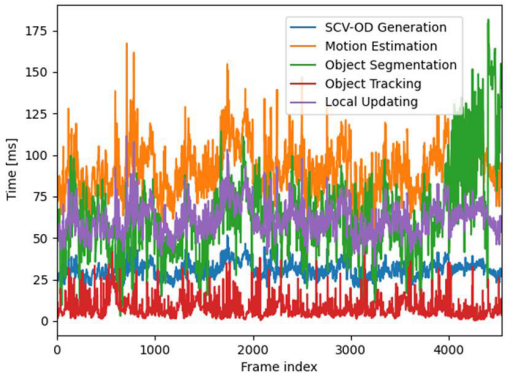


Fig. 17. Real-time performance impacts of different modules on Seq 00 of the KITTI dataset [26].

VIII. CONCLUSION

This article proposes a novel egocentric descriptor termed SCV-OD for building a dynamic-aware and LiDAR-only

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

SLAM. Using the SCV-OD as the backbone, the proposed SLAM framework integrates object segmentation, object tracking, and LiDAR mapping in a tight-coupled and consistent manner. Through qualitative and quantitative comparisons with SOTA methods using the KITTI dataset and a custom dataset, our proposed approach demonstrates superior performance in static mapping. In practical applications, the proposed method can be used for robust localization and map updating for mobile sensing with various LiDAR types (e.g., mechanical Velodyne and solid-state Livox). For future works, we plan to extend our research by developing a multisession SLAM system that enables collaboration between multiple platforms in complex environments. We intend to make our algorithms open-source at the appropriate time, facilitating wider adoption and contribution to the research community.

REFERENCES

[1] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets: Open-source library and experimental protocol," *Auto. Robots*, vol. 34, no. 3, pp. 133–148, Apr. 2013.

[2] 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)*, Jun. 2003, pp. 2743–2748.

[3] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "MULLS: Versatile LiDAR SLAM via multi-metric linear least square," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11633–11640.

[4] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for fast and accurate 3D point cloud registration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11054–11059.

[5] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot., Sci. Syst.*, vol. 2, Berkeley, CA, USA, 2014, pp. 1–9.

[6] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3126–3131.

[7] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Proc. Robot., Sci. Syst.*, 2018, p. 59.

[8] P. Zhou, X. Guo, X. Pei, and C. Chen, "T-LOAM: Truncated least squares LiDAR-only odometry and mapping in real time," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2021.

[9] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: Segment based place recognition in 3D point clouds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5266–5272.

[10] 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.

[11] H. Luo et al., "Semantic labeling of mobile LiDAR point clouds via active learning and higher order MRF," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 3631–3644, Jul. 2018.

[12] 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.

[13] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled LiDAR-visual-inertial odometry via smoothing and mapping," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 5692–5698.

[14] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.

[15] X. Han, Y. Gao, Z. Lu, Z. Zhang, and D. Niu, "Research on moving object detection algorithm based on improved three frame difference method and optical flow," in *Proc. 5th Int. Conf. Instrum. Meas., Comput., Commun. Control (IMCCC)*, Sep. 2015, pp. 580–584.

[16] X. Liu, Z. Yang, J. Hou, and W. Huang, "Dynamic scene's laser localization by NeuroIV-based moving objects detection and LiDAR points evaluation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[17] 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.

[18] H. Liu, C. Lin, B. Gong, and D. Wu, "Extending the detection range for low-channel roadside LiDAR by static background construction," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2022.

[19] J. Park, Y. Cho, and Y.-S. Shin, "Nonparametric background model-based LiDAR SLAM in highly dynamic urban environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24190–24205, Dec. 2022.

[20] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auto. Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.

[21] 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.

[22] 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.

[23] X. Chen et al., "Automatic labeling to generate training data for online LiDAR-based moving object segmentation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6107–6114, Jul. 2022.

[24] J. Sun et al., "Efficient spatial–temporal information fusion for LiDAR-based 3D moving object segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2022, pp. 11456–11463.

[25] N. Wang, C. Shi, R. Guo, H. Lu, Z. Zheng, and X. Chen, "InsMOS: Instance-aware moving object segmentation in LiDAR data," 2023, *arXiv:2303.03909*.

[26] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[27] H. Li, B. Tian, H. Shen, and J. Lu, "An intensity-augmented LiDAR-inertial SLAM for solid-state LiDARs in degenerated environments," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[28] S. Park, S. Wang, H. Lim, and U. Kang, "Curved-voxel clustering for accurate segmentation of 3D LiDAR point clouds with real-time performance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6459–6464.

[29] A. G. Kashani, M. J. Olsen, C. E. Parrish, and N. Wilson, "A review of LiDAR radiometric processing: From ad hoc intensity correction to rigorous radiometric calibration," *Sensors*, vol. 15, no. 11, pp. 28099–28128, 2015.

[30] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3D object classification," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2011, pp. 2987–2992.

[31] K. G. Derpanis, "Overview of the RANSAC algorithm," *Image Rochester NY*, vol. 4, no. 1, pp. 2–3, 2010.

[32] N. Bhatia and Vandana, "Survey of nearest neighbor techniques," 2010, *arXiv:1007.0085*.

[33] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9297–9307.

[34] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, Jun. 2010.

[35] M. Arora, L. Wiesmann, X. Chen, and C. Stachniss, "Static map generation from 3D LiDAR point clouds exploiting ground segmentation," *Robot. Auto. Syst.*, vol. 159, Jan. 2023, Art. no. 104287.

[36] Q. Zhang, D. Duberg, R. Geng, M. Jia, L. Wang, and P. Jensfelt, "A dynamic points removal benchmark in point cloud maps," 2023, *arXiv:2307.07260*.

[37] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "LiTAMIN2: Ultra light LiDAR-based SLAM using geometric approximation applied with KL-divergence," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11619–11625.

[38] 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.

[39] D. Chang, S. Huang, R. Zhang, M. Hu, R. Ding, and X. Qin, "WiCRF2: Multi-weighted LiDAR odometry and mapping with motion observability features," *IEEE Sensors J.*, vol. 23, no. 17, pp. 20236–20246, Sep. 2023.

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.

[40] B. Mersch, X. Chen, I. Vizzo, L. Nunes, J. Behley, and C. Stachniss, "Receding moving object segmentation in 3D LiDAR data using sparse 4D convolutions," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7503–7510, Jul. 2022.

[41] Z. Wang, Z. Zhang, X. Kang, M. Wu, S. Chen, and Q. Li, "DOR-LINS: Dynamic objects removal LiDAR-inertial SLAM based on ground pseudo occupancy," *IEEE Sensors J.*, vol. 23, no. 20, pp. 24907–24915, Oct. 2023.

Yixin Fang received the B.S. degree from the Nanjing University of Information Science and Technology, Nanjing, China, in 2021. He is currently pursuing the M.S. degree with the School of Automation, Southeast University, Nanjing.

His research interests include SLAM.

Kun Qian (Member, IEEE) received the Ph.D. degree in control theory and control engineering from Southeast University, Nanjing, China, in 2010.

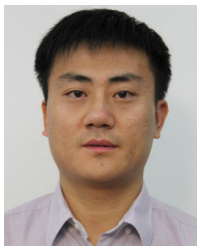
He is currently an Associate Professor with the School of Automation, Southeast University. His research interests include robot vision and SLAM.

Yun Zhang received the B.S. degree from Nanjing Normal University, Nanjing, China, in 2015, and the M.S. degree from Guangxi University, Nanning, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Automation, Southeast University, Nanjing, China.

His research interests include SLAM.

Tong Shi received the B.S. degree from the Harbin Institute of Technology, Weihai, China, in 2023. He is currently pursuing the M.S. degree with the School of Automation, Southeast University, Nanjing, China.

His research interests include SLAM.



Hai Yu received the B.S. degree from Southeast University, Nanjing, China, in 2003, and the M.S. degree from Nanjing University, Nanjing, in 2006.

He is currently with the Global Energy Interconnection Research Institute, State Grid Smart Grid Research Institute, Nanjing. His research interests include information and communication technology in power systems.

Authorized licensed use limited to: Shenyang Institute of Automation. Downloaded on October 14,2024 at 10:50:48 UTC from IEEE Xplore. Restrictions apply.