Nonparametric Background Model-Based LiDAR SLAM in Highly Dynamic Urban Environments

Joohyun Park, *Member, IEEE*, Younggun Cho, *Member, IEEE*, and Young-Sik Shin, *Member, IEEE*

***Abstract*— In urban environments, simultaneous localization and mapping (SLAM) are essential for autonomous driving. Most light detection and ranging (LiDAR) SLAM methodologies have been developed for relatively static environments, despite real-world environments having many dynamic objects such as vehicles, bicycles, and pedestrians. This paper proposes an efficient and robust LiDAR SLAM. Our SLAM framework leverages the estimated background model to achieve robust motion estimation in dynamic urban environments. Based on probabilistic object estimation, the dynamic removal module estimates a nonparametric background model to recognize dynamic objects. This module estimates the probability of the difference of the range values from the accumulated LiDAR frames. Then, dynamic objects are removed by adapting the sensor velocity from the estimated ego-motion. In the local mapping module, our method optimizes the LiDAR motion considering the dynamic characteristics of LiDAR point clouds. Finally, the proposed method results in a global map with static point clouds and accurate LiDAR motion with global pose optimization. We tested the proposed method on the well-known public dataset (KITTI) and the custom dataset with complex environments, including various moving objects. Comparisons with state-of-the-art (SOTA) methods demonstrate that the our approach is more robust and efficient. For example, the proposed method performed an average 0.63% and 0.18°/100 *m* errors on the KITTI dataset with 0.96*ms* processing time that convinces real-time processing.**

***Index Terms*— SLAM, mapping, dynamic objects.**

I. Introduction

RECENTLY, many researchers have widely used 3D light detection and ranging (LiDAR) for autonomous navigation. Because 3D LiDAR provide precise and accurate measurements, most autonomous platforms utilize 3D LiDAR for path planning, collision avoidance, localization and consistent map building. Among the 3-stages of autonomous driving algorithms (perception, planning, and control), simultaneous localization and mapping (SLAM) is fundamental for the perception of surrounding environments. In particular, LiDAR-based SLAM methods have focused on accurate localization [1], [2], [3], [4]. Although previous studies achieved good localization performance, they assumes relatively static environments.

In urban environments, many objects and structures surround the ego vehicle. In particular, the robust SLAM recognizing static and dynamic objects are essential for autonomous vehicles. Additionally, *dynamic* objects in urban environments have various sizes, shapes, and speeds; therefore, it is difficult to estimate the accurate poses of ego-motion. To achieve robust and efficient autonomous driving in an arbitrary space, it is essential to establish *static* information for localization and mapping well in advance. In terms of localization and mapping for autonomous driving, dynamic objects are a representative factor that degrades performance. Accordingly, various algorithms [5], [6], [7] have been developed to remove dynamic objects from 3D LiDAR-equipped vehicles. Such static information is stored as a map in various forms, such as features, occupancy grids, voxels, and point clouds, depending on the given data type or sensor characteristics.

In this paper, we propose a dynamic-aware SLAM algorithm using LiDAR for highly dynamic urban environments. Fig. 1 presents the result map (black dots) and estimated dynamic objects (red dots) for the KITTI 00 sequence [8]. A nonparametric background model estimator is proposed for robust motion estimation and reliable mapping; this approach is inspired by the background-aware visual odometry in [9]. Unlike cameras, LiDAR provides only geometric information with a relatively low resolution, and it is challenging to directly utilize such computer vision-based methods. The proposed method estimates the motion of the non-static sensor and simultaneously estimates the probabilistic background model by using the difference in the range measurements. Furthermore, we exploit the estimated dynamic object to mitigate the degradation exhibited by motion estimation. The background model estimator is also used to reduce the trailing effect caused by moving objects in the 3D map building process. The main characteristics of the proposed method are as follows:

- We propose a novel dynamic object-aware and robust LiDAR-based SLAM approach for autonomous vehicles in highly dynamic environments. Our method estimates the accurate ego-motion of the vehicle and generates

Manuscript received 15 November 2021; revised 19 April 2022 and 5 July 2022; accepted 24 August 2022. Date of publication 21 September 2022; date of current version 5 December 2022. This work was supported in part by the Major Institutional Project of Korea Institute of Machinery and Materials funded by the Ministry of Science and ICT (MSIT) (Development of Core Machinery Technologies for Autonomous Operation and Manufacturing) under Grant NK236H; in part by the Korea Agency for Infrastructure Technology Advancement (KAIA) Grant funded by the Ministry of Land, Infrastructure and Transport under Grant RS-2022-00143717; in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) Grant funded by the Korea Government (MSIT) (Deep Total Recall: Continual Learning for Human-Like Recall of Artificial Neural Networks) under Grant 2022-0-00448; and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant 2022R1A4A3029480. The Associate Editor for this article was G. Mao. *(Corresponding authors: Younggun Cho; Young-Sik Shin.)*

Joohyun Park is with the Vision Group, NAVER LABS, Gyeonggi-do 13638, South Korea (e-mail: joohyun.p@naverlabs.com).

Younggun Cho is with the Department of Electrical Engineering, Inha University, Incheon 20220, South Korea (e-mail: yg.cho@inha.ac.kr).

Young-Sik Shin is with the Korea Institute of Machinery and Materials, Daejeon 34103, South Korea (e-mail: yshin86@kimm.re.kr).

This article has supplementary downloadable material available at https://doi.org/10.1109/TITS.2022.3204917, provided by the authors.

Digital Object Identifier 10.1109/TITS.2022.3204917

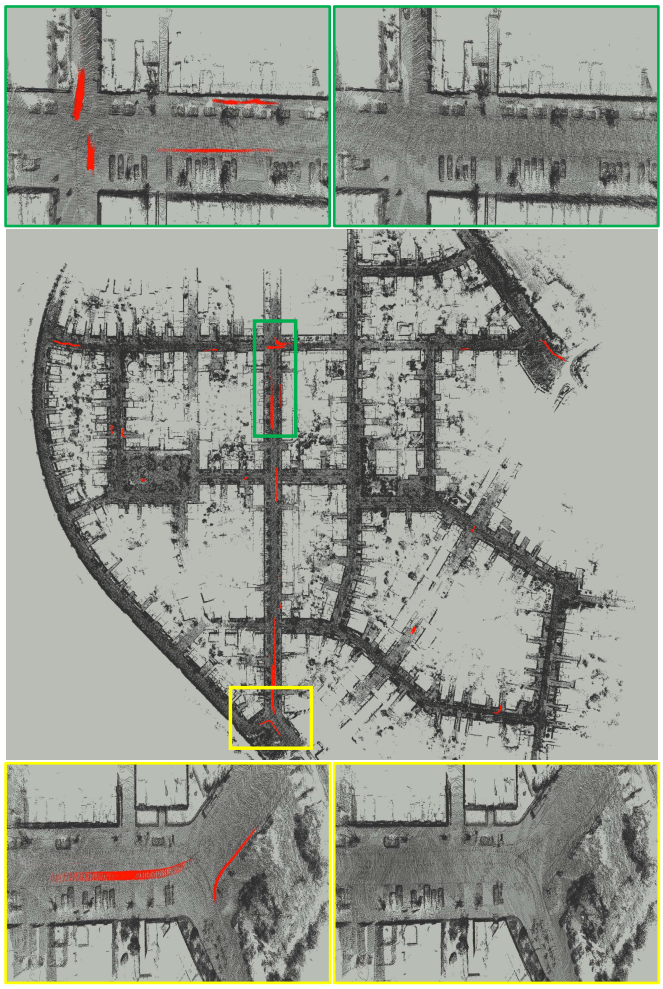


Fig. 1. The result map of dynamic-aware SLAM. The left column at the top and the bottom shows the mapping result without dynamic removal and the right column represents our dynamic-aware SLAM pipeline. The proposed method reconstructs 3D static scenes without dynamic objects.

a static map from nonparametric background model estimation.

- The proposed method fully utilizes a spherical image representation for efficient data association of LiDAR measurements. The background model and motion estimation use the same shared representation for the unified framework of the efficient LiDAR-based SLAM.
- The proposed method is thoroughly evaluated through our platform and on public datasets that include various environments. Finally, we demonstrate through the experimental results that dynamic object estimation based on a nonparametric background model improves SLAM performance and static map building performance in urban environments.

The rest of this paper is organized as follows: In Section II, we discuss related works and the uniqueness of our work dealing with moving objects and improving SLAM. Section III presents an overview of the proposed robust dynamic-aware LiDAR SLAM method. Section IV explains the details of the dynamic removal method with nonparametric background model estimation. The robust LiDAR-based SLAM method enhanced by dynamic removal is described in Section V. The evaluation results are shown in Section VI, and we also discuss the conclusions regarding the proposed methods.

II. Related Works

Many studies on motion estimation and dynamic object removal based on 3D LiDAR have been conducted in recent years. This section briefly reviews the relevant studies on 1) LiDAR-based localization and mapping, and 2) dynamic object removal methods in highly dynamic urban environments.

A. LiDAR-Based Localization and Mapping

A LiDAR-based localization and mapping algorithm is the most popular SLAM approach in urban environments. According to the representations of LiDAR measurements and associations, we can summarize these studies into several categories. The basic approach directly utilizes LiDAR point clouds with the iterated closest point (ICP) algorithm [10]. Recent approaches proposed various representations such as voxels [11], LiDAR features [2], and Surface Elements(Surfels) [4]. Also, correspondence searching is the main difference between the algorithms. Practically, the methodologies of ICP variants utilize subsampled points and various data structures such as kd-trees, voxel grids, and projective images to match corresponding points for computational efficiency [2], [3], [12], [13], [14].

Zhang and Singh [2] proposed LOAM, which subsamples input point clouds into planar edge features by exploiting the curvature of the points. Then, kd-trees for each feature set are constructed for the nearest-neighbor search. LOAM represents the global map based on LiDAR features. Also, methods based on the normal distribution transform (NDT) are another category of voxel-related approaches [11], [15], [16]. These methods utilize voxel-wise correspondence. Yokozuka et al. [17] proposed LiTAMIN2 based on the NDT. They approximated the input points as a normal distribution on the voxel grid to accelerate the speed. Then, normal distribution alignment was performed with the symmetric KL-divergence.

Methods that utilize projective geometry for nearest neighbor searches also were proposed [4], [14]. Behley and Stachniss [4] proposed an efficient 3D LiDAR-based SLAM called SuMa, which uses a shader language for parallel computation. They utilized spherical image projection for the input point clouds and performed association by considering those points to be the nearest points. Furthermore, this paper reported that Surfel-based mapping could significantly reduce motion drift. Recently, many researchers have proposed learning-based LiDAR motion estimation methods ([18], [19]) with projective LiDAR representations. Although most methods utilize the ICP loss for learning, supervised learning showed the disadvantage of being vulnerable to overfitting, and unsupervised learning reported limited performance.

We summarize much research on LiDAR-based motion estimation. For most of the above methods, the robustness of their algorithms in highly dynamic environments is noted. However,

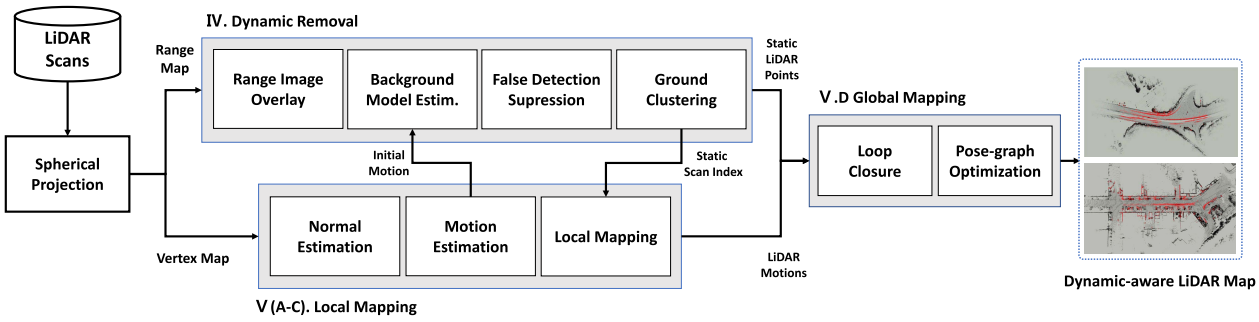


Fig. 2. Overview of the proposed method. The main algorithms are composed of three modules: dynamic removal, dynamic-aware local mapping and global mapping. The detail of the algorithms are described in Section IV and Section V.

previous studies acquired their performance *implicitly* through geometric feature selection or robust loss functions such as *RANSAC* [20], the *Huber loss* and the *Tukey loss*. These approaches have limitations with respect to various moving objects, and we aim to resolve these limitations in this paper.

B. Dynamic Object Removal (Static Mapping)

Many studies have been conducted on detecting or removing dynamic objects in urban environments. This section introduces dynamic object removal methods and categorizes them into visibility-based, learning-based, and map-based approaches.

Visibility-based methods, also known as change detection-based methods, rely on comparisons between consecutive scans. Xiao et al. [21] used the weighted Dempster-Shafer theory (WDST) to extract objects. This method uses the point-to-triangle (PTT) distance for classifying static or dynamic object points. The scan is compared against multiple scans before and after the current scan. Vallet et al. [22] used Dempster-Shafer theory (DST) for extracting objects and mapped them back to images to extract images of moving objects. Yoon et al. [23] proposed a model-free approach that compares the query scan against a reference scan based on geometric errors. Also, this method applied region growing-based clustering to extract object-wise dynamic point clouds.

Learning-based semantic segmentation or detection has led to additional significant improvements in performance. Ruchti and Burgard [24] combined a neural network and OctoMap to predict the probability of point clouds corresponding to dynamic objects. Pagad et al. [25] used an aggregate view object detection (AVOD) [26] network for object detection. This method combines object detection and OctoMap. A voxel traversal and occupancy probability update strategy and an occupancy map as a binary filter were proposed for this method to extract clean point clouds. Recently, Chen et al. [27] proposed the neural networks to detect dynamic objects with bounding boxes, and removing the points inside the bounding boxes is a trivial task.

Finally, map-based methods assume given pre-built map and LiDAR motions when removing dynamic objects. Hornung et al. [28] first proposed Octomap. This method utilized a Bayesian rule to construct an occupancy map via the ray casting algorithm of point clouds. Schauer and Nüchter [29] proposed removing dynamic points by traversing a voxel occupancy grid. Kim and Kim [5] proposed a pixel-to-window comparison method to consider incidence angle ambiguity. Lim et al. [6] proposed a visibility-free approach and leveraged their proposed representation of points organized in vertical columns, called pseudo-occupancy, based on an occupancy grid map. In the case of [5] and [6], a ground-truths-like pose should be a premise, and it has the characteristics of a post-processing algorithm, not a real-time algorithm that generates a map while estimating motion.

Among the categories above, map-based approaches are suitable for point cloud mapping in urban environments. In particular, in 3D static map generation using LiDAR sensors, most map-based methods rely on accurate and precise LiDAR motion to determine static point clouds. In this paper, we combine motion estimation and dynamic removal to simultaneously solve these problems.

III. System Overview

Our proposed robust LiDAR SLAM is composed of three main modules: dynamic removal, dynamic-aware local mapping and global mapping as shown in Fig. 2.

First, our method utilizes a projective transform to the LiDAR scan points as the base representation of the method. The inputs of the main algorithms are range and angle map-based representations. The converted LiDAR frames are applied to both main modules. The dynamic removal module identifies dynamic objects and static backgrounds from observed scenes through nonparametric background model estimation. Then the identified dynamic objects are utilized for the robust motion estimation and mapping process.

Note that our method estimates both the dynamic objects and relative ego-motions in a loosely-coupled manner. It is the main difference between existing LiDAR-based static mapping studies. The dynamic removal module utilizes the initial motion of the motion estimation of the local mapping module. Also, the motion estimation compensates for the initial LiDAR motions by using the static points filtered by the dynamic module. Through this complementary process, our method achieves the two objectives of accurate motion estimation and static map generation in various dynamic scenarios. Finally, the global mapping module generates a global map from static point clouds and precise LiDAR motion.

A. Preliminaries

1) Notation: We represent each scan of the LiDAR as frame $\mathcal{F}_t$ at timestamp t. Its poses are represented as a transformation matrix $\mathbf{T}_t \in \mathbb{R}^{4\times4}$. Let the corresponding rotational and translational parts be $\mathbf{R}_t \in \mathbb{R}^{3\times3}$ and $\mathbf{t} \in \mathbb{R}^3$. A point $\mathbf{p}$ in coordinate frame i is denoted by $\mathbf{T}_{ji}$ as $\mathbf{p}_j$ in coordinate frame j. Additionally, lie algebra elements ξ_t are used as minimal representations to optimize the state vector of motion estimation. The element $\xi = [\mathbf{w}, \mathbf{v}] = \left[w_x, w_y, w_z, v_x, v_y, v_z\right]$ consists of two vectors, angular ($\mathbf{w}$) and linear velocity ($\mathbf{v}$) vectors. It is mapped to $SE(3)$ by exponential mapping, and its inverse is denoted by log mapping. These mapping techniques are used to update the transformation matrix through the increment obtained by the linear solver: $se(3) \boxplus SE(3) \rightarrow SE(3)$.

B. Preprocessing

1) Spherical Projection: For projective data association, we first project the information (e.g, range, angle) of the point cloud $\mathcal{P}$ to generate an image representation. Each point $\mathbf{p}_i = (x, y, z)$ is converted from Cartesian coordinates to image coordinates (u, v) via a mapping $\mathbf{\Pi} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$

$$\Pi(\mathbf{p}_i) = \begin{pmatrix} \frac{1}{2}[1 - \arctan(y, x)\,\pi^{-1}]\; w \\ [1 - (acos(z\, r^{-1}) + f_{wn}^{do})/(f_{wn}^{do} + f_{up})]\; h \end{pmatrix}, \quad (1)$$

where (h, w) are the desired image representations of height and width. f_{wn}^{do} and f_{up} are the vertical field of view (FOV) of the LiDAR sensor. The range value $\mathbf{r}_i = \| \mathbf{p}_i \|_2$ represents the Euclidean distance from the corresponding point $\mathbf{p}_i$ of the sensor. The angle value $\mathbf{a}_i = \arctan \frac{\mathbf{p}_i z}{\|\mathbf{r}_i\|_2}$ represents the vertical angle of the sensor corresponding to $\mathbf{p}_i$. Each valid point $\mathbf{p}_i$ is now represented by a unique pixel in the range and angle image. If several 3D points are projected onto the same pixel, then we choose the nearest point as a pixel value. The images have not only the range and angle of the point, but also their 3D information (x, y, z). Therefore, projective data association is a key factor that can organize data without using other searching methods (e.g., kd-tree [30] or octree [31]) and can enable it to be managed highly efficiently.

2) LiDAR Distortion Compensation: In order to improve the accuracy, the LiDAR whose distortion caused by ego-motion is compensated is used as an input to the system. Given the consecutive two LiDAR poses $\mathbf{T}_k$ and $\mathbf{T}_{k+1}$, the linear interpolated pose $\mathbf{T}_\tau$ is defined as:

$$\mathbf{T}_\tau = \mathbf{T}_k exp\left(\lambda \xi_{k,k+1}\right) \quad (2)$$

where their timestamps satisfy $\tau_k < \tau < \tau_{k+1}$. $\xi_{k,k+1} = log\left(\mathbf{T}_k^{-1}\mathbf{T}_{k+1}\right)$ is used to interpolate pose $\mathbf{T}_\tau$ with interpolation ratio $\lambda = (\tau - \tau_k)\,/\,(\tau_{k+1} - \tau_k)$. Finally, based on interpolate poses, LiDAR scan points are compensated and associated with the nearest LiDAR frame.

IV. Dynamic Objects Removal

Our proposed background model estimation method addresses the limitations of the visibility method through the accumulation of differential images, and addresses the false detection caused by point ambiguity through false detection suppression and ground-aware clustering. The proposed background model estimation method consists of 4 steps, as shown in Algorithm 1. Differential images are accumulated through range overlay, and the result is $\mathcal{I}^d$ in Fig. 3. By estimating the nonparametric background model using the accumulated differential images, the *background model* results in Fig. 3 are obtained. Since the estimated background model includes false detections caused by point ambiguity and LiDAR motion ambiguity, the *labeled background model* $\left(\mathcal{B}^{est}\right)$ results in Fig. 3 are obtained by false detection suppression and ground-aware clustering.

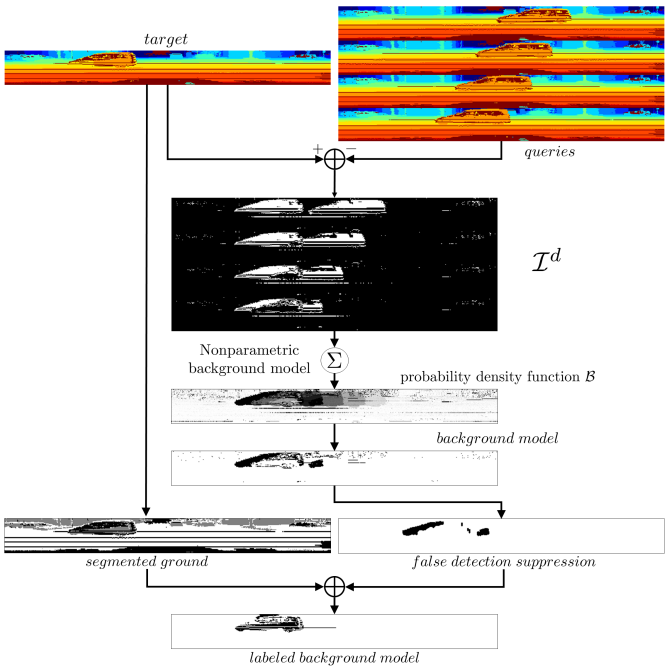


Fig. 3. Overview of the background model estimator.

A. Range Image Overlay

A range image is obtained by finding the position of the point cloud on the image through (1) and having the range value of the point cloud to the corresponding pixel value. For background model estimation, this module first accumulates differential range images called overlay images. The key idea is that if the points are transformed by the initial motions, then the static points should be projected into the same pixels on the overlay image. To create the viewpoints of successive images that are the same, the transform function $\mathcal{T}(\cdot)$ is defined as

$$\mathcal{T}(\xi_i^j) = \mathbf{T}_{ji} = \mathbf{T}_{wj}^{-1}\mathbf{T}_{wi} = \exp\left(\begin{bmatrix} 0 & -w_z & w_y & v_x \\ w_z & 0 & -w_x & v_y \\ -w_y & w_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{bmatrix}_i^j\right), \quad (3)$$

where ξ_i^j represents transformation in the 3D space. The relative pose transform between frames $\mathcal{F}_i$ and $\mathcal{F}_j$ can be

denoted by $\mathbf{T}_{ji} = \mathbf{T}_{wj}^{-1}\mathbf{T}_{wi}$ and this relationship can be used to transform the coordinates of the points as follows:

$$\mathbf{p}'_{ji} = \mathbf{T}_{ij}\mathbf{p}_i = \begin{bmatrix} \mathbf{R}_{ji} & \mathbf{t}_{ji} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{p}_i, \tag{4}$$

where $\mathbf{R}_{ji} \in SO(3)$ is a 3×3 rotation matrix and $\mathbf{t}_{ji} \in \mathbb{R}^3$ is a 3×1 translation vector. ξ_i^j is composed of an angular relative difference $\mathbf{R}_{ji}$ and translation relative difference $\mathbf{t}_{ji}$ of a 6-degree-of-freedom (DOF) rigid motion at a unit time. $\mathbf{p}_i$ is a point on frame $\mathcal{F}_i$ and $\mathbf{p}'_{ji}$ is the transformed point from frame $\mathcal{F}_i$ to $\mathcal{F}_j$.

From the above process, the range between frames is compensated by transforming the point cloud ($\mathcal{P}_{ji}$) to the current frame $\mathcal{F}_j$. A range image for the transformed point cloud is generated through (1), and a differential range image($\mathcal{I}_{ji}^d$) at frame $\mathcal{F}_j$ is generated.

$$\mathcal{I}_{ji}^d = \mathcal{I}_j^{range} - \mathcal{I}_i^{range}. \tag{5}$$

The background model is defined as an accumulation of the difference images $\mathcal{I}_{ji}^d$ between the current image and its remapped range image for a period of time.

B. Background Model Estimator

This section describes estimating a background model based on range images. We propose a nonparametric background estimator based on the probability of static-dynamic points. To estimate the background model, the previous N frames are transformed into the current frame and a differential image ($\mathcal{I}^d$) is created. The differential range image can be used to calculate the probability density function of the background. The calculated density function can be estimated nonparametrically for each pixel. The probability density function $\mathcal{B}(\cdot)$ is defined as

$$\mathcal{B}\Big(\sum_{n=k}^{N+k-1} \mathcal{I}_{(N+k-1)n}^d\Big) = \frac{1}{N}\sum_{n=k}^{N+k-1} \frac{1}{\sqrt{2\pi\sigma_z^2}} exp\left(-\frac{1}{2}\left[\mathcal{I}_{(N+k-1)n}^d \frac{1}{\sigma_z}\right]^2\right), \tag{6}$$

where σ_z is the deviation over the differential range images for each pixel and N is the size of the accumulation of differential range images. Since the background model is estimated using the differential range images corresponding to N frames, the value of N affects the accurate estimation of the background model. If the interval between frames is too great, the differential image may not be generated because there are few areas in common with each other.

The deviation of σ_z is estimated using the median absolute deviation for all the pixels in the range images as

$$\sigma_z = C \cdot \frac{med_n\left(\left|\mathcal{I}_{(N+n-1)n}^d\right|\right)}{\sqrt{2}} \simeq \frac{median_n\left(\left|\mathcal{I}_{(N+n-1)n}^d\right|\right)}{\Phi^{-1}(3/4)\sqrt{2}}, \tag{7}$$

Algorithm 1 Background Model Estimation

Input : Point Cloud $\mathcal{P}$, Initial Motion $\mathcal{M}$
Output : Estimated background model $\mathcal{B}^{est}$

Let $\mathcal{S}$ be set of point cloud $\mathcal{P}$ and initial motion $\mathcal{M}$
if $\mathcal{S}.size() \geq \mathcal{N}$ **then**
  Let $\mathcal{P}_{\mathcal{F}_T}, \mathcal{T}_{\mathcal{F}_T}^{\mathcal{M}}$ are data of target frame
  Let $\mathcal{P}_{\mathcal{F}_Q}^{\mathcal{N}}, \mathcal{T}_{\mathcal{F}_Q}^{\mathcal{N}}$ are data of target frame
  ___Range Overlay_____
  for $i = 1, 2, \ldots, N-1$ **do**
    $\mathcal{I}_i^d \leftarrow Range\ Overlay\left(\mathcal{P}_{\mathcal{F}_T}, \mathcal{T}_{\mathcal{F}_T}^{\mathcal{M}}, \mathcal{P}_{\mathcal{F}_Q}^{\mathcal{N}}, \mathcal{T}_{\mathcal{F}_Q}^{\mathcal{N}}\right)$
  end for
  ___Nonparametric Background Model_____
  Let $\mathcal{B}$ be a background model probability density function
  $\mathcal{B} \leftarrow Non\text{-}parametric\ background\ model\left(\mathcal{I}^d\right)$
  ___False Detection Suppression_____
  Let $\mathcal{B}^{FDS}$ be a false detection suppression
  $\mathcal{B}^{FDS} \leftarrow False\ Detection\ Suppression\left(\mathcal{B}\right)$
  ___Ground-aware Clustering_____
  Let ${}^{img}\mathcal{T}^g$ be a segmented ground of target frame
  $\mathcal{B}^{est} \leftarrow Ground\text{-}aware\ Clustering\left(\mathcal{B}^{FDS}, \mathcal{P}_{\mathcal{F}_T}\right)$
  Delete the first element of $\mathcal{S}$
end if

where C and Φ are the constant scale factor and cumulative distribution function, respectively. In the background model, the motion of dynamic objects is removed through the reference pixels of the threshold (ϵ^B), and the background model image ($\mathcal{I}_{k+N}^B$) from which the dynamic objects are removed is defined as follows:

$$\mathcal{I}_{k+N}^B(u,v) = \begin{cases} 1, & \mathcal{B}(\sum_{n=k}^{N+k-1} \mathcal{I}_{(N+k-1)n}^d) \geqq \epsilon_{k+N}^B \\ 0, & \mathcal{B}(\sum_{n=k}^{N+k-1} \mathcal{I}_{(N+k-1)n}^d) < \epsilon_{k+N}^B \end{cases}, \tag{8}$$

$$\epsilon_{k+N}^B = \frac{0.6744}{\sqrt{\pi v_{(k+N)}^2}} exp(-0.4548), \tag{9}$$

where ϵ_{k+N}^B is the threshold that determines the background model and is affected by speed. $v_{(k+N)}^2$ means the speed from frame $\mathcal{F}_k$ to frame $\mathcal{F}_{k+N}$. In the probability distribution of consecutive differential range images, $v_{(k+N)}^2$ is determined to have a 95% confidence interval of:

$$v_{(k+N)}^2 = V_c \frac{v_{(k+N)} - v_k}{\Delta t_{(k+N)k}} \sigma_r, \tag{10}$$

where σ_r and $\Delta t_{(k+N)k}$ are the standard deviation of the distribution of the differential range image and the time difference between two frames.

C. False Detection Suppression

False detections are more likely to occur in outdoor environments with background fluctuations. False detection occurs for the following reasons:

- Low resolution of the range image and imprecise pose
- Small movements in the background (e.g., vegetation, ground plane)

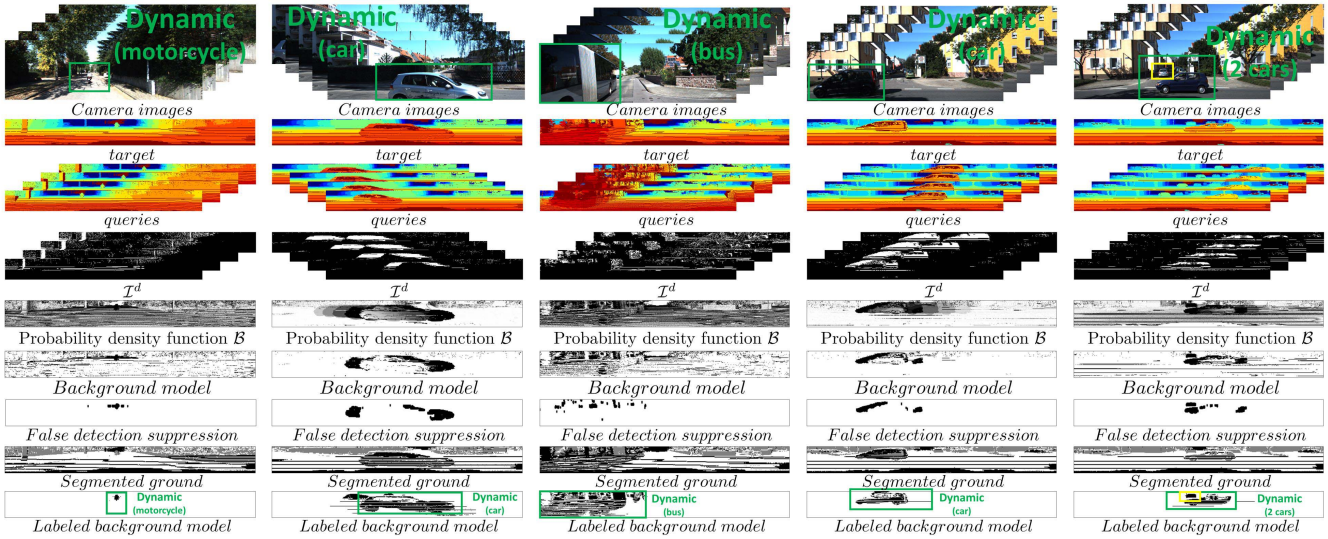


Fig. 4. Result of the background model estimator obtained by accumulating 5 frames ($N = 5$). The second row is the result of generating differential images between the target and the query images. For color information of range image, red means near and blue means far away. The probability density function $\mathcal{B}$ is calculated through (6), the area corresponding to the dynamic object is white, and the area corresponding to the static object is black. The estimated background image is obtained through (8). Due to the problem of LiDAR resolution and point ambiguity, many false measurements (outliers) are detected, which prevents accurate background model estimation. Therefore, if all outliers are removed using false detection suppression, the dynamic and static objects are segmented. Because the dynamic object is detected by accumulating consecutive scenes, the problem with the plane is not accurately detected, and the gap between the dynamic objects is filled through clustering. From the above results, it can be seen that dynamic objects, from small to large are detected well without being affected by the size.

The low resolution of an image causes many false detections at the edge. Imprecise poses cause many false positives due to the accumulation of errors in estimating the background model based on the accumulated differential range image. The points corresponding to vegetation and the ground plane have ambiguity because they change rather than always remaining in a fixed position. This ambiguity distinguishes points that should be included in the background model as false detections. Methods such as mathematical morphology and filtering can affect small or occluded objects, making it challenging to eliminate false detections accurately.

The detailed explanation of false detection suppression is given in Algorithm 2. We perform false detection suppression to clarify only dynamic objects. First, in lines 5 to 17 in Algorithm 2, the surrounding points of the dynamic object detected by the background model estimator are expanded through a nearest neighbor search mask kernel. Then, in lines 20 to 25 in Algorithm 2, dynamic objects are detected using a range-aware clustering algorithm such as region-growing. Through these two processes, our method effectively estimates false detections and detects only dynamic objects, and it obtains the same results as *false detection suppression* in Fig. 4.

D. Ground-Aware Clustering

In urban environments, filtering large objects (such as buses and trucks) is challenging. Because the geometric primitives of the objects are similar in consecutive frames, the background estimation model can easily confuse these as static LiDAR points. For this reason, clustering is necessary to detect planes of dynamic object areas that do not change significantly. However, in clustering, static object points, rather than dynamic object points, may be incorrectly clustered. Typically, a large portion of the point cloud collected in an urban environment corresponds to the ground plane. Therefore, the plane of the dynamic object is detected through segmentation of the ground plane and clustering based on the segmented ground plane. The extraction of the ground plane points based on the image requires two assumptions:

- Low curvature of the inclined ground plane, so that it 'is not too steep
- A starting point of ground plane segmentation that is in: the lowest row of the image

Therefore, points corresponding to the ground plane are extracted through a column-wise evaluation method using two images, which can be regarded as the ground plane estimation. Column-wise evaluation is a method of judging using only the columns of the image. The relevant area is determined by comparing the column-wise correspondence of pixels located below or above the target pixel. The angle between the two points evaluated in this way separates the ground and nonground according to the threshold. Algorithm 3 represents the ground segmentation and clustering procedures. In lines 10 to 12 of Algorithm 3, it is judged whether the points are ground points through the aforementioned column-wise evaluation. The *segmented ground* in Fig. 4 is the result of ground segmentation. Through lines 18 to 28 in Algorithm 3, we estimate the *labeled background model* $\left(\mathcal{B}^{est}\right)$ in Fig. 4, which is the result of clustering using the segmented ground and the results of false detection suppression.

V. Dynamic-Aware LiDAR SLAM

We designed an efficient and robust LiDAR SLAM pipeline using nearest neighbor search through a projective image

Algorithm 2 False Detection Suppression

Input : Background model probability density function $\mathcal{B}$
Output : False detection suppression $\mathcal{B}^{FDS}$

Let ${}^{img}\mathcal{Q}$ be a queue containing all the pixels to be masked
Let $\mathcal{N}\left({}^{img}\mathbf{p}\right)$ be set of neighbors of pixel ${}^{img}\mathbf{p}$
Let Λ be a mask kernel for neighbor search
Add all pixels of $\mathcal{B}$ from Λ to ${}^{img}\mathcal{Q}$
while ${}^{img}\mathcal{Q} \neq 0$ **do**
  Add all pixels inside ${}^{img}\mathcal{Q}$ to $\mathcal{B}^{FDS}$
  for all ${}^{img}\mathbf{p} \in {}^{img}\mathcal{Q}$ **do**
    for all ${}^{img}\mathbf{n} \in \mathcal{N}\left({}^{img}\mathbf{p}\right)$ **do**
      diff = $\| {}_{img}\mathbf{p}\mathcal{P}_{\mathcal{F}_T} - {}_{img}\mathbf{n}\mathcal{P}_{\mathcal{F}_T} \|$
      if diff $< \theta \cdot {}_{img}\mathbf{p}\mathcal{P}_{\mathcal{F}_T}$ **then**
        if $\mathbf{n} \notin \mathcal{B}^{FDS}$ **then**
          Add $\mathbf{n}$ to ${}^{img}\mathcal{Q}$
        end if
      end if
    end for
    Delete ${}^{img}\mathbf{p}$ from ${}^{img}\mathcal{Q}$
  end for
end while
Clear ${}^{img}\mathcal{Q}$ and $\mathcal{N}\left({}^{img}\mathbf{p}\right)$
Add all pixels of $\mathcal{B}^{FDS}$ from Λ to ${}^{img}\mathcal{Q}$
while ${}^{img}\mathcal{Q} \neq 0$ **do**
  for all ${}^{img}\mathbf{p} \in {}^{img}\mathcal{Q}$ **do**
    ${}^{img}\mathbf{p} \leftarrow min\left({}^{img}\mathbf{p}, \mathcal{N}\left({}^{img}\mathbf{p}\right)\right)$
    Add ${}^{img}\mathbf{p}$ to $\mathcal{B}^{FDS}$
    Delete ${}^{img}\mathbf{p}$ from ${}^{img}\mathcal{Q}$
  end for
end while

Algorithm 3 Ground Plane Segmentation and Clustering

Input : False Detection Suppression $\mathcal{B}^{FDS}$, Point Cloud $\mathcal{P}_{\mathcal{F}_T}$
Output : Labeled background model $\mathcal{B}^{est}$

Let $\mathcal{I}^{range}_{\mathcal{P}_{\mathcal{F}_T}}, \mathcal{I}^{angle}_{\mathcal{P}_{\mathcal{F}_T}}$ are range and angle image of $\mathcal{P}_{\mathcal{F}_T}$
Let ${}_r\mathbf{p}_l$, ${}_a\mathbf{p}_l$ are pixels of the starting point row
Let ϕ be threshold of ground segmentation
Let $\mathcal{I}^g$ be segmented ground image
___Ground Segmentation___
for $u = l-1, l-2, \ldots, 2$ **do**
  $\Delta z = \| {}_r\mathbf{p}_l \cdot sin\left({}_a\mathbf{p}_l\right) - {}_r\mathbf{p}_u \cdot sin\left({}_a\mathbf{p}_u\right) \|$
  $\Delta x = \| {}_r\mathbf{p}_l \cdot cos\left({}_a\mathbf{p}_l\right) - {}_r\mathbf{p}_u \cdot cos\left({}_a\mathbf{p}_u\right) \|$
  $\psi = \arctan\left(\Delta z / \Delta x\right)$
  if $\psi \leqq \phi$ **then**
    $\mathcal{I}^g\left(\mathbf{p}_l\right)$ is segmented by ground
  end if
end for
___Clustering___
Let ${}^c\mathbf{p}$ be pixels of not segmented ground of $\mathcal{I}^g$
Let $\mathcal{N}\left({}^c\mathbf{p}\right)$ be set of neighbors of pixel ${}^c\mathbf{p}$
Let θ be threshold of clustering
while ${}^c\mathbf{p} \neq 0$ **do**
  for all ${}^c\mathbf{n} \in \mathcal{N}\left({}^c\mathbf{p}\right)$ **do**
    if $\| {}_{{}^c\mathbf{p}}\mathcal{P}_{\mathcal{F}_T} - {}_{{}^c\mathbf{n}}\mathcal{P}_{\mathcal{F}_T} \| < \theta \cdot {}_{{}^c\mathbf{p}}\mathcal{P}_{\mathcal{F}_T}$ **then**
      if $\mathcal{B}^{FDS}\left({}^c\mathbf{n}\right)$ is not background **then**
        $\mathcal{B}^{est}\left({}^c\mathbf{p}\right)$ is labeled as dynamic objects
        $\mathcal{B}^{est}\left({}^c\mathbf{n}\right)$ is labeled as dynamic objects
      end if
    end if
  end for
  Delete ${}^c\mathbf{p}$
end while

plane. To ensure high accuracy, a frame-to-model scheme is applied as in [2], [3], [17], and [4].

The motion estimation module optimizes the relative motion T_t^{t-1} of the current frame from the local map with the current point cloud and local map as a model. The local map consists of a vertex map $\mathcal{V}_M$ and a normal map $\mathcal{N}_M$, and only static points obtained from the dynamic removal module are updated. In addition, we apply a pose graph-based SLAM backend, including the loop closing technique for global consistency.

A. Normal Estimation

Stable normal estimation is essential for pose update through point-to-plane registration. In the previous methods [4], [32], the surface normal is calculated by taking the cross-product of the gradients of the vertex map. We estimate the normal based on the spherical image plane using eigen decomposition. Only those points with sufficient neighboring points on 5×5 window that satisfy the planar score are retained as measurements. A plane is defined as $\mathbf{n} \cdot \mathbf{p} - d = 0$, where $\mathbf{n}$ and d correspond to the normal of the plane and distance parameters. Given a subset of points, the optimal normal is obtained by solving the cost function as described in [33].

B. Motion Estimation

To compute current motion corresponding to the local map, The spherical image has vertices $\mathbf{v}$ and normals $\mathbf{n}$ corresponding to its coordinates. Given each vertex and normal map from the current frame and local map, we can effectively perform projective data association on the spherical image plane. Given the vertex map $\mathcal{V}_C$ of the current frame and vertex map $\mathcal{V}_M$ and the normal map $\mathcal{N}_M$of the local map, we use the point-to-plane error to estimate the relative pose as shown below:

$$E_S(\mathcal{V}_C, \mathcal{V}_M, \mathcal{N}_M) = \sum_{\mathbf{v}_C \in \mathcal{V}_C} [\mathbf{n}_M^T (T_t^{t-1} \mathbf{v}_C - \mathbf{v}_M)]^2, \quad (11)$$

where each vertex $\mathbf{v}_C \in \mathcal{V}_C$ is associated with the vertex $\mathbf{v}_M \in \mathcal{V}_M$ and $\mathbf{n}_M \in \mathcal{N}_M$ from the local map via

$$\mathbf{v}_M = \mathcal{V}_M(\Pi(T_t^{t-1} \mathbf{v}_C)), \quad (12)$$

$$\mathbf{n}_M = \mathcal{N}_M(\Pi(T_t^{t-1} \mathbf{v}_C)). \quad (13)$$

Here, T_t^{t-1} is the relative pose between the current frames of the local map, and $\Pi(\cdot)$ is the projection function to the spherical image. If a point is projected out of the spherical image, that point is ignored in the error function. We solve the cost function of Equation (11) using Gauss-Newton optimization. In each iteration, increment $\delta \in se(3)$ is represented

as a Lie-algebra element. The Jacobian $\mathbf{J}_{\mathbf{v}_C}$ of each vertex under a single residual is given by

$$\mathbf{J}_M = \left[\mathbf{n}_M^T \ (\mathbf{n}_M \times \mathbf{v}_M)^T\right]. \tag{14}$$

By the Jacobian $\mathbf{J}_M$ in the above expression, the increment δ is determined as follows:

$$\delta = (\mathbf{J}^T\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^T\mathbf{W}\mathbf{r}, \tag{15}$$

where $\mathbf{W}$ is defined by the dynamic object estimated from Equation (8) and the weight based on the t-distribution for robust estimation. The dynamic-aware t-distribution weight is calculated iteratively and recursively as in [34] and [35]:

$$w_t = \mathcal{I}^B \frac{\nu + 1}{\nu + (\frac{r}{\sigma_r})^2}, \tag{16}$$

where $\mathcal{I}^B$ means static points given from (8). ν is the degree of freedom, which is set to five, and the variance of the residual σ_r is iteratively estimated during the iteration of the optimization process.

C. Dynamic-Aware Local Mapping

Given the estimated motions (T_t^{t+1}) and static point clouds with the normal ($\mathcal{V}^S = \mathcal{V}(B^{est})$, $\mathcal{N}^S = \mathcal{N}(B^{est})$), we update the point cloud into local map.

To facilitate projective data association, this module directly accumulates static point clouds from multiple frames. We keep the last 20 frames in memory and update the local map by transforming these frames to the most recent frame. For efficiency, we reuse the generated local map by projecting it to the next frame. To construct a local map, we first merge only the points that satisfy the criteria of $\left|\mathbf{n}_m^T(\mathbf{v}_s - \mathbf{v}_m)\right| < \delta_m$ and $\|\mathbf{n}_s \times \mathbf{n}_m\| < \theta_m$ in order to eliminate noisy measurements. If the points are compatible, the vertices and normals of the associated points are updated using an exponential moving average with weight γ as in [4] and [32]:

$$\mathbf{v}_m^t = (1-\gamma)\cdot\mathbf{v}_s + \gamma\cdot\mathbf{v}_m^{(t-1)}, \tag{17}$$

$$\mathbf{n}_m^t = (1-\gamma)\cdot\mathbf{n}_s + \gamma\cdot\mathbf{n}_m^{(t-1)}. \tag{18}$$

where $\mathbf{v}_m^t$ and $\mathbf{n}_m^t$ are updated vertex and normal from static points from the Section IV.

D. Global Mapping

1) Loop Closure: Although dynamic-aware frame-to-model motion estimation allows for significantly low drift, small errors eventually accumulate. Eventually, the global consistency of the trajectory and map is not guaranteed. Therefore, it is essential to recognize revisited places and optimize the trajectory, including the loop constraints between the revisited area and current frame, to alleviate these problems.

In our framework, the place recognition task depends on the LiDAR descriptor called *scan context (SC)* proposed in [36]. Given the candidate frame from the place recognition module, we construct a local map in a similar way to that in Section V-C.

2) Pose-Graph Optimization: As the back-end of the global mapping module, pose graph optimization is utilized to maintain global consistency. The formula for pose graph optimization is as follows when expressed with sequential nodes and loop constraints:

$$\mathbf{x} = \text{argmin} \sum_i \left\|\mathbf{z}_{i,i+1} - \hat{\mathbf{z}}_{i,i+1}\right\|^2_{\sum_{i,i+1}} + \sum_{i,j} \left\|\mathbf{z}_{i,j} - \hat{\mathbf{z}}_{i,j}\right\|^2_{\sum_{i,j}} \tag{19}$$

where $\mathbf{z}_{i,i+1}$ and $\hat{\mathbf{z}}_{i,i+1}$ denotes measurement and observation of sequential node, and $\mathbf{z}_{i,j}$ and $\hat{\mathbf{z}_{i,j}}$ denote the loop constraint. We utilize `iSAM2` [37] as a solver for online operation the SLAM problem.

Finally, we provide the loop constraint for the pose graph SLAM backend only for candidates who have passed geometric verification by utilizing point-to-plane ICP through projective association as in our previous work [35].

VI. Experimental Evaluations

In this section, we evaluate the proposed method via quantitative and qualitative results of LiDAR SLAM and dynamic object removal compared to recent state-of-the-art methods. Using publicly open datasets, such as KITTI [8], [38] and our dataset, we fairly test our method and determine the performance. For all experiments, the algorithm was fully developed utilizing a consumer-level laptop with an Intel Core i7-7700HQ and 8GB of RAM. The algorithms are implemented in C++, and Robot Operating System (ROS) of Ubuntu Linux is used.

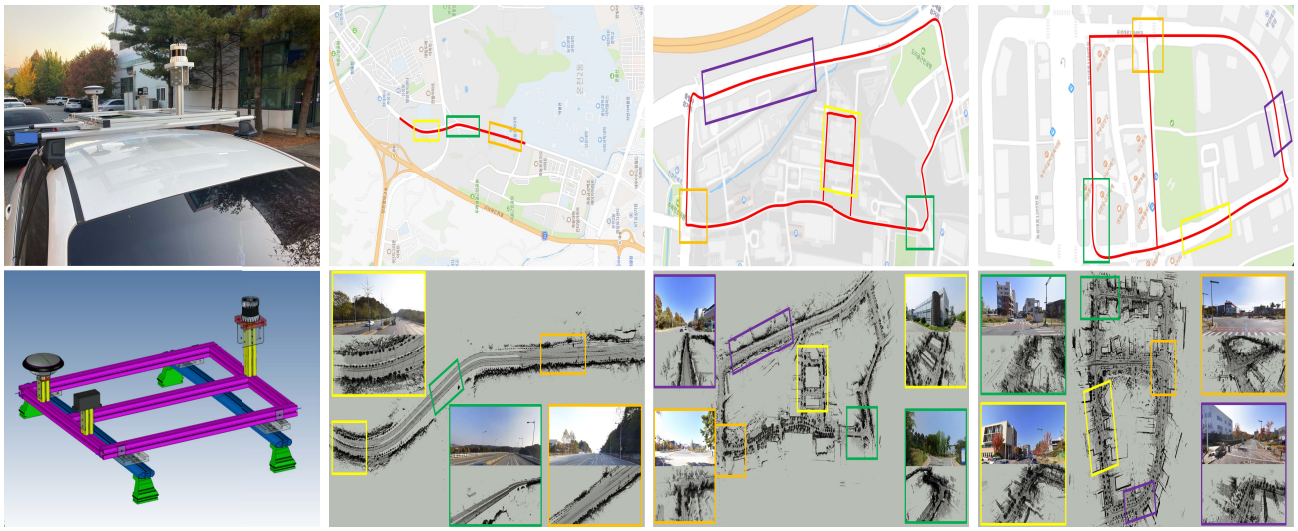
A. Datasets

To verify the performance of the algorithms, we tested the proposed method on datasets with having various dynamic conditions. For every dataset, we evaluated both the dynamic object removal and SLAM performance.

KITTI Benchmark Dataset. The original KITTI dataset [8] provides LiDAR that has 64 laser beams (Velodyne HDL-64E) with an FOV of 30° in vertical and 360° in horizontal. In addition, the dataset provides ground-truth poses synchronized with LiDAR using a high-precision inertial navigation system (INS). The KITTI dataset covers various environments, including small urban areas, suburban areas with small roads and vegetation, and highways with dynamic vehicles. The KITTI dataset provides 11 sequences with ground-truth poses, and we evaluate our robust LiDAR SLAM performance compared to the ground-truth.

Motivated by the original KITTI dataset, the SemanticKITTI dataset [38] provides pointwise labeled LiDAR points synchronized with the KITTI dataset and associated LiDAR SLAM-based poses together. The dataset uses a data format that is the same as that of the original KITTI dataset and contains 28 classes distinguishing nonmoving and moving objects. We evaluated the performance of the proposed dynamic removal module using only labels related to dynamic objects.

Our Datasets (High Dynamic Variations). We performed additional experiments to validate the proposed method in



(a) Sensor system (b) Juk-dong (c) Sinsung-dong (d) KIMM

Fig. 5. Our sensor system and datasets. The system is equipped with 64-channel LiDAR and RTK-GPS for the ground-truth poses. The dataset is composed of different sequences. (b) Juk-dong: Highway, High-dynamic. (c) Sinsung-dong: Complex residential area, various objects. (d) KIMM: Various road types.

more diverse environments. As shown in Fig(a), our sensor system is equipped with 64 channel LiDAR (Ouster OS-1 64), and high-precision INS (InertialLabs INS-P). The LiDAR has the same data form at as the original KITTI dataset, but with a vertical FOV of 45°, and it provides an organized point cloud. The INS has a built-in tactical-grade inertial measurement unit (IMU) and provides cm-level accuracy by using the RTK-correction signal. By assuming the INS measurements as the ground-truth, we computed the absolute position error of the proposed method. Unlike KITTI dataset which provide undistorted LiDAR measurements, we apply motion compensation to the raw LiDAR measurements.

We manually labeled the dynamic objects in the dataset utilizing the point cloud labeling tool [38]. As described in Fig. VI-D.2, the dataset includes 3 sequences (Juk-dong, Sinsung-dong and KIMM) with various dynamic objects and road conditions. The total travel distance of the 3 sequences is approximately 7.5 km. The Juk-dong sequence includes a highway scenario with highly dynamic vehicles. The Sinsung-dong sequence is based on complex residential areas. The dataset has various types of static-dynamic objects. The last sequence (KIMM) has mixed environments with inner (narrow) and outside (wide) roads. Each sequence has different characteristics common in urban environments. We believe that the proposed dataset is suitable for verifying the scalability of the algorithms.

B. Evaluation Metrics

1) Performance of Dynamic Removal: We evaluated the background model estimation through the error metrics proposed in [6]. The error metrics are defined as follows:

- PR : $\frac{\text{\# of preserved static points}}{\text{\# of total static points on the ground-truth}}$
- RR : $1-\frac{\text{\# of preserved dynamic points}}{\text{\# of total dynamic points on the ground-truth}}$

PR evaluates how well static objects are preserved, and RR evaluates how well dynamic objects are removed. PR and RR are calculated voxel-wise. The state-of-the-art methods are difficult to compare quantitatively and fairly because each voxelization is performed individually. We apply identical voxelization $\nu(\cdot)$ with voxel size 0.2 for ground-truths retrieved from the baseline models for fair comparison. Let $\mathbf{q} \in \nu(\mathcal{G})$ and $\mathbf{s} \in \nu(S_{est})$ where $\mathbf{s}$ is the nearest point to $\mathbf{q}$. $\mathcal{G}$ is the ground-truth map and S_{est} is the estimated static map. Then, the preservation function $\mathcal{K}(\cdot)$ is defined as follows:

$$\mathcal{K}(\mathbf{q}, \mathbf{s}) = \text{iVOI}(\mathbf{q}, \mathbf{s}) \wedge \psi(\mathbf{q}, \mathbf{s}) \tag{20}$$

where iVOI means inter-voxel inclusion and is a Boolean function that returns true if two points are in the same voxel, and false otherwise. Likewise, $\psi(\cdot)$ returns true if both points are static or dynamic and false otherwise. Through the evaluation method proposed in [6], both the state-of-the-art methods and the proposed method can be evaluated quantitatively and fairly.

2) Performance of LiDAR SLAM: To evaluate the performance of motion estimation on the KITTI dataset, we utilize the average translational and rotational errors for all possible subsequences of length (100 m, $\cdots$, 800 m). As described in [8], our evaluation computes the average translation $t_{rel}(\%)$ and rotation $r_{rel}(^\circ/100\ m)$ root mean square error (RMSE) drift, described as

$$r_{rel}(\mathcal{F}) = \frac{1}{|\mathcal{F}|} \sum_{(i,j)\in\mathcal{F}} \angle\left[\left(\hat{\eta}_j \ominus \hat{\eta}_i\right) \ominus \left(\eta_j \ominus \eta_i\right)\right], \tag{21}$$

$$t_{rel}(\mathcal{F}) = \frac{1}{|\mathcal{F}|} \sum_{(i,j)\in\mathcal{F}} \left\|\left(\hat{\eta}_j \ominus \hat{\eta}_i\right) \ominus \left(\eta_j \ominus \eta_i\right)\right\|_2, \tag{22}$$

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

C. Evaluation on KITTI Dataset

1) Dynamic Removal: We compared state-of-the-art methods that could leverage open-source implementation for the

experiments, namely, OctoMap [28], PeopleRemover [29], Removert [5], and ERASOR [6] through the KITTI dataset to quantitatively evaluate the results of the background model estimation.

As mentioned in [6], we selected the top-5 time frames with many dynamic objects in the SemanticKITTI dataset. The performance of the background estimation model was evaluated by performing the top-5 experiments, and the odometry performance was evaluated using the entire frame. Therefore, frames Seq 00 (4390∼4530), Seq 01 (150∼250), Seq 02 (860∼950), Seq 05 (2350∼2670) and Seq 07 (630∼820) were selected as the background model estimation benchmarks. The selected frames include various environments, such as intersections, countryside, and highways.

Seq 00 is an urban environment with pedestrians and cars as dynamic objects at an intersection. Seq 01 is a highway environment, and most of the dynamic objects are cars. Because there are bollards in the middle of the highway, it is difficult to detect moving cars in the opposite lane. Furthermore, the bollards are not uniformly located; this may cause them to be falsely detected as dynamic objects. Seq 02 is an urban environment with intersections similar to Seq 00. The dynamic objects include pedestrians, bicycle rider, and moving cars. Seq 05 is a typical urban environment. It is an environment in which intersections appear 4 times, there are moving cars, and a bus appears as a dynamic object. Unlike Seq 00, 02 and 05, 07 is an environment in which many dynamic objects appear while stopping at the beginning.

Table I presents quantative comparisons of the dynamic removal results. We compared our method to OctoMap with different voxel sizes (0.05 and 0.2), PeopleRemover, Removert with various remove and revert stages, and ERASOR with different sizes of scan ratio threshold (SRT) [6]. Unlike the proposed method, the compared methods rely on pre built map or known LiDAR poses. Therefore, we utilized baseline LiDAR poses provided from SuMa [4] which feature inherent uncertainty.

As described in Table I, the proposed method ranked first or second in F_1 score compared to existing state-of-the-art methods (Removert and ERASOR). Removert utilizes the discrepancy in multiresolution range images to remove dynamic objects. Particularly in case of Removert, the algorithm is overconfident for dynamic object estimation in remove stage and vice versa on the revert state. In these results, it can be seen that the results with the remove stage (RM2, 3) have a higher PR but lower RR compared to the result with the revert stage (RM3+RV1). ERASOR estimates dynamic objects by comparing the pre built map and the current scan represented as a regionwise pseudo-occupancy descriptor (R-POD) using SC. The SRT directly determines the performance of dynamic object estimation. Because the SRT is affected by point ambiguity and LiDAR motion ambiguity, ERASOR, with a higher SRT, shows a higher score in RR but a lower score in PR. Unlike the previous methods that depend on point discrepancy, the proposed method utilizes the accumulation of differential range images for nonparametric background model estimation. Additionally, our model shows better and more stable results in most of the scenarios using false detection suppression and ground-aware clustering.

TABLE I

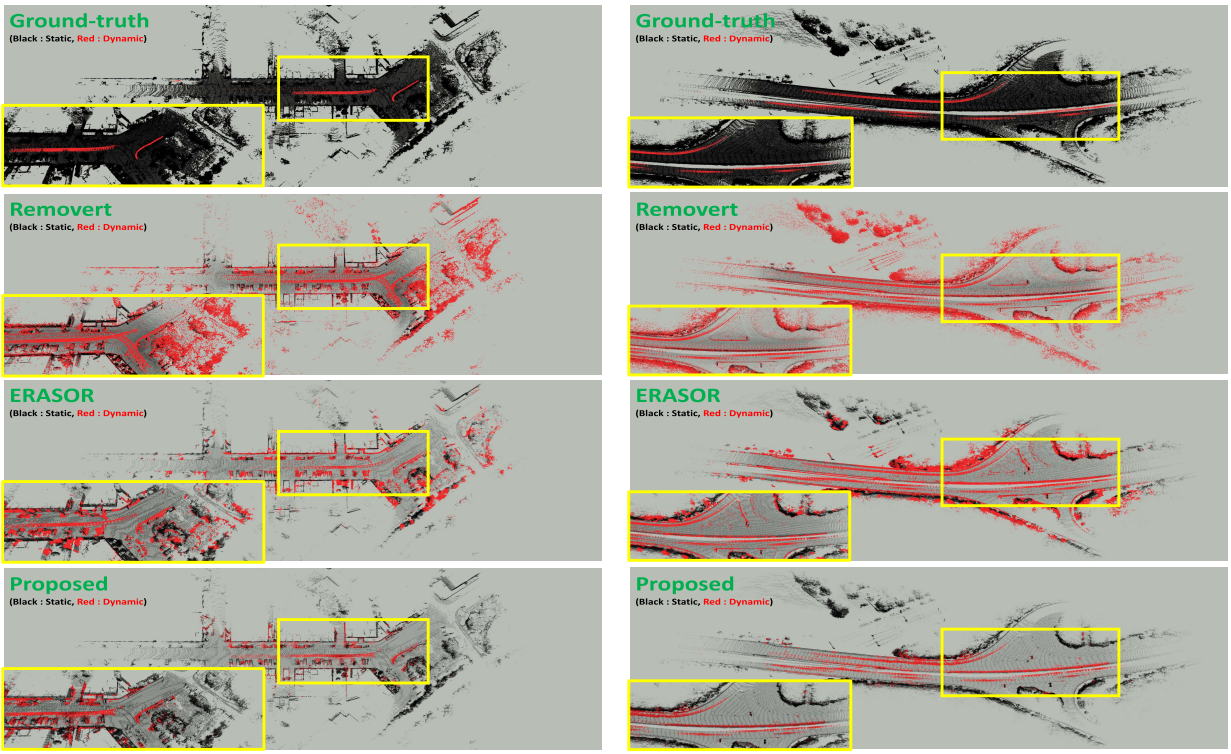
COMPARISON RESULTS OF THE BACKGROUND MODEL ESTIMATION USING THE KITTI DATASET AND SEMANTICKITTI DATASET

PR: Preservation Rate, RR: Rejection Rate

Seq.	Method	PR[%]	RR[%]	F_1 score
00	OctoMap - 0.05 [28]	76.731	99.124	0.865
	OctoMap - 0.2 [28]	34.568	**99.979**	0.514
	PeopleRemover [29]	37.523	89.116	0.528
	Removert - RM2 [5]	89.684	98.648	0.940
	Removert - RM3 (0.1°) [5]	86.154	98.784	0.920
	Removert - RM3 (0.05°) [5]	85.502	99.354	0.919
	Removert - RM3+RV1 [5]	86.829	90.617	0.887
	ERASOR - 0.2 [6]	93.980	97.081	0.955
	ERASOR - 0.4 [6]	80.769	98.353	0.887
	Proposed(Ours)	**94.050**	97.190	**0.956**
01	OctoMap - 0.05 [28]	53.163	99.663	0.693
	OctoMap - 0.2 [28]	20.777	**99.863**	0.344
	PeopleRemover [29]	36.349	93.116	0.523
	Removert - RM2 [5]	94.214	93.106	0.937
	Removert - RM3 (0.1°) [5]	92.831	92.998	0.929
	Removert - RM3 (0.05°) [5]	94.221	93.608	**0.939**
	Removert - RM3+RV1 [5]	**95.815**	57.077	0.715
	ERASOR - 0.2 [6]	91.487	95.383	0.934
	ERASOR - 0.4 [6]	79.807	95.080	0.868
	Proposed(Ours)	91.815	94.096	0.929
02	OctoMap - 0.05 [28]	54.112	98.769	0.699
	OctoMap - 0.2 [28]	23.746	**99.792**	0.384
	PeopleRemover [29]	29.037	94.527	0.444
	Removert - RM2 [5]	80.648	96.941	0.880
	Removert - RM3 (0.1°) [5]	74.159	97.493	0.842
	Removert - RM3 (0.05°) [5]	76.319	96.799	0.853
	Removert - RM3+RV1 [5]	83.293	88.371	0.858
	ERASOR - 0.2 [6]	87.731	97.008	0.921
	ERASOR - 0.4 [6]	78.319	98.464	0.872
	Proposed(Ours)	**91.208**	95.510	**0.933**
05	OctoMap - 0.05 [28]	76.341	96.785	0.854
	OctoMap - 0.2 [28]	33.904	**99.882**	0.506
	PeopleRemover [29]	38.495	90.631	0.540
	Removert - RM2 [5]	84.074	90.163	0.870
	Removert - RM3 (0.1°) [5]	83.115	90.421	0.866
	Removert - RM3 (0.05°) [5]	86.900	87.880	0.874
	Removert - RM3+RV1 [5]	88.170	79.981	0.839
	ERASOR - 0.3 [6]	88.730	98.262	0.933
	ERASOR - 0.6 [6]	72.440	97.831	0.832
	Proposed(Ours)	**93.820**	95.740	**0.947**
07	OctoMap - 0.05 [28]	77.838	96.938	0.863
	OctoMap - 0.2 [28]	38.183	**99.565**	0.552
	PeopleRemover [29]	34.772	91.983	0.505
	Removert - RM2 [5]	81.805	94.125	0.875
	Removert - RM3 (0.1°) [5]	77.967	99.107	0.873
	Removert - RM3 (0.05°)[5]	80.689	98.822	0.888
	Removert - RM3+RV1 [5]	82.038	95.504	0.883
	ERASOR - 0.2 [6]	90.624	99.271	**0.948**
	ERASOR - 0.4 [6]	82.875	98.221	0.899
	Proposed(Ours)	**91.146**	97.284	0.941

Fig. 6 and Fig. 7 represent the results of the background model estimation for KITTI dataset. We compare our proposed method to the ground-truth points and recent static mapping methods [5] and [6]. For detailed evaluation, an enlarged view of the point clouds is provided in the yellow box. In the figure, the black and red points represent static and dynamic objects.

In the case of Removert, because the revert stage is not provided in the open-source implementation, only the remove stage is used to evaluate the performance. As can be seen in the figures, Removert strictly deletes nonstatic points in



(a) Seq 00 (b) Seq 01

Fig. 6. Top to bottom: Ground-truth, Removert [5], ERASOR [6], proposed methods. The ground-truth image classifies dynamic and static objects using semanticKITTI labels, black point clouds indicate static objects and red point clouds indicate dynamic objects. For the results of the compared state-of-the-art methods, the red point clouds are determined as a dynamic object, and the black point clouds are determined as a static object. The image marked with a yellow box on the upper right is an enlarged scene of the corresponding area.

the removal stage and compensates for false negatives in the revert stage. For Seq 01, which is the highway scenario, Removert has the result that most areas, such as the ground, are classified well as static objects, but many other areas are incorrectly classified as dynamic objects. Areas such as vegetation are misclassified as dynamic objects because many differentials are created with even a small movement. Additionally, in the general urban sequences 00, 02, 05, and 07, Removert incorrectly classifies parked vehicles as dynamic objects because the discrepancy LiDAR points from the viewpoint change. These characteristics are represented in Table I. In the table, Removert shows a low PR and high RR with only the remove stage and a higher PR and lower RR with additional revert stage.

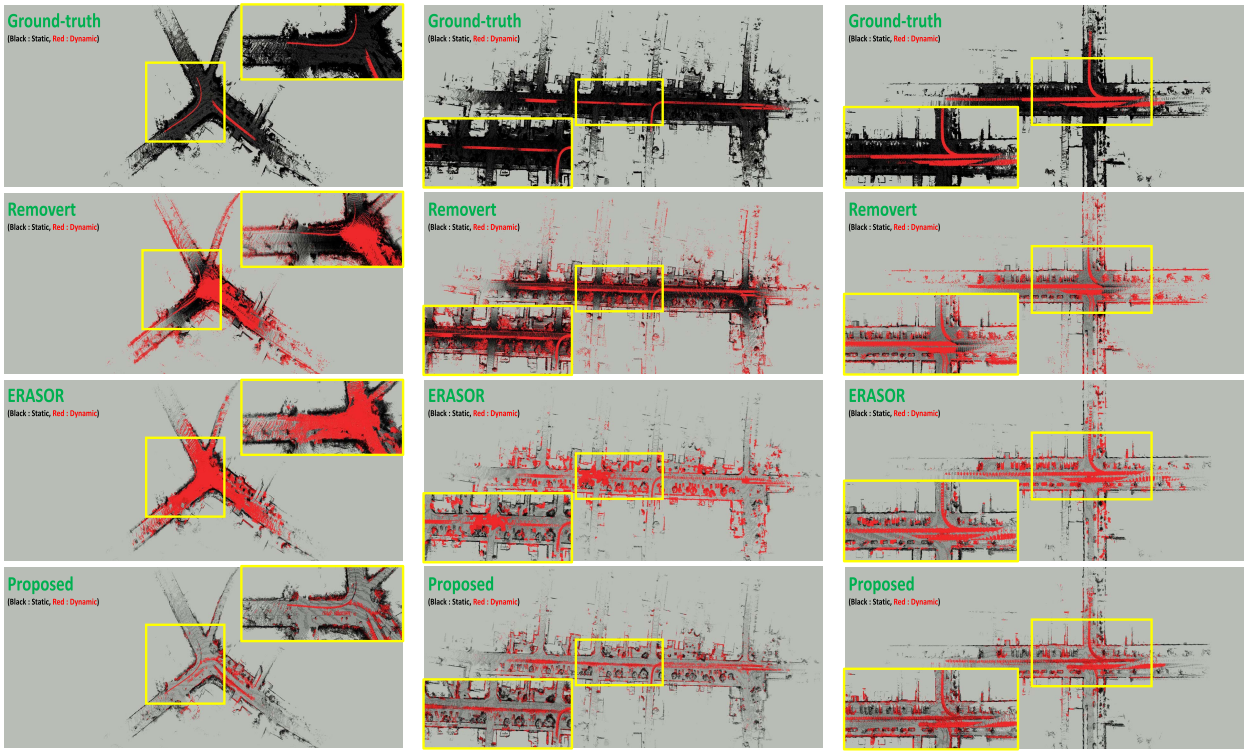
ERASOR shows similar characteristics. For Seq 02 and 05, ERASOR estimates the ground points as dynamic objects due to the ambiguity of LiDAR motion. Because ERASOR assumes an accurate and precise prior map, it can misclassify dynamic points as static objects when the R-POD of the voxel inclusion (VOI) of the dynamic object of the current scan is the same as the R-POD of the VOI of the prebuilt map. For Seq 01, ERASOR classifies static points (vegetation or trees) as dynamic points. While ERASOR relies on the SRT for estimation, objects with high uncertainty around roads, such as street trees, are easily misclassified.

Unlike previous studies, the proposed method has better performance in dynamic points detection (and the same is true for background model estimation). The proposed method estimates the nonparametric background model by accumulating the differences of consecutive range images to overcome varying point cloud distributions caused by viewpoint changes. Therefore, our method correctly classifies parked vehicles as static points. Additionally, the false detection suppression and ground-aware clustering successfully depress false estimations of points with high ambiguities. In addition, the results of Table II show that it can operate in real-time while simultaneously performing motion estimation and static map generation while showing good performance. Furthermore, it shows a runtime similar to [5] and [6], which are algorithms for correcting maps based on ground-truths-like poses. And it can be seen that the results in Table III work robustly for LiDAR horizontal angle resolution. In the case of LiDAR, data loss

TABLE II
RUNTIME OF OVERALL SYSTEM USING THE KITTI DATASET AND SEMANTICKITTI DATASET

DOR: Dynamic Object Removal

Method	Computation Time [ms]
OctoMap[28]	107.7
PeopleRemover[29]	100.0
Removert[5]	83.1
ERASOR[6]	73.2
Proposed (DOR)	59.9
Proposed (SLAM + DOR)	96.3



(a) Seq 02 (b) Seq 05 (c) Seq 07

Fig. 7. Comparison of the background model estimation results of Seq 02,05,07. Top to bottom: Ground-truth, Removert, ERASOR, proposed methods. The ground-truth classifies dynamic and static objects using SemanticKITTI labels; the black point clouds indicate static objects, and the red point clouds indicate dynamic objects. The area corresponding to the yellow box is enlarged.

TABLE III
ABLATION STUDY RESULTS ACCORDING TO LIDAR HORIZONTAL ANGLE RESOLUTION

Seq.	Resolution	PR[%]	RR[%]	F_1 score	Runtime[ms]
00	0.25	93.8	94.4	0.94	86.0
	0.3	91.5	93.8	0.93	78.6
01	0.25	89.2	92.4	0.91	81.8
	0.3	88.4	91.5	0.90	76.6
02	0.25	90.3	93.3	0.92	83.8
	0.3	89.5	92.6	0.91	78.8
05	0.25	90.2	92.7	0.91	89.2
	0.3	88.8	92.1	0.90	79.7
07	0.25	88.2	95.0	0.92	82.3
	0.3	88.1	90.2	0.89	80.1

may occur while image projection is performed by a sparse sensor compared to the camera, but in the proposed method, it can be seen that it operates robustly using a nonparametric background model.

2) LiDAR SLAM: We evaluated the proposed LiDAR SLAM results compared to existing 3D LiDAR SLAM approaches. We compared the proposed algorithms with the state-of-the-art methods among various LiDAR SLAM methods to evaluate the motion estimation performance. The results of each method in Table IV directly refer to the results reported in the corresponding papers. In the case of LOAM [2], IMLS-SLAM [3], and MC2SLAM [13], only translation drift is provided. For DeepLO [19], Seqs 00 to 08 were removed from the table because those sequences were used for training sets. For SuMa [4], SuMa++ [40], LiTAMIN2 [17], MULLS [39], we directly include the estimation results from the papers. The proposed method and SuMa, SuMa++, and DeepLO are approaches based on projective data association, while the others are tree or voxel-based data association approaches.

Table IV shows the comparison results of the average translational and rotational errors. As seen in the table, the trajectory estimation results are most dependent on the dynamic removal module. When we perform trajectory estimation without the dynamic removal module, it shows an error of 0.88% in translation and 0.25°/100 *m* in rotation despite full SLAM including loop closure. It shows lower performance than odometry with dynamic removal. Furthermore, the proposed method with dynamic removal is improved to 0.63% and 0.18°/100 *m* through full SLAM with loop closure. Although the translational error performance is slightly lower than that of the tree-based approaches (IMLS-SLAM, MC2SLAM), it has the best performance among the projective association approaches. In addition, our method improves the 3D mapping performance by dynamic removal of spherical range images, as shown in Section VI-C.1. In particular, in SuMa++, semantic information obtained from a deep neural network with semantic SLAM is used for dynamic object filtering. Therefore, additional hardware such as *GPGPU* and operations for dynamic removal are required. On the other hand, our method effectively performs dynamic removal without

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

	Seq. 00	Seq. 01	Seq. 02	Seq. 03	Seq. 04	Seq. 05	Seq. 06	Seq. 07	Seq. 08	Seq. 09	Seq. 10	Avg.
Proposed (odom) (w/o removal)	0.59/0.25	3.26/0.55	0.66/0.19	1.08/0.57	0.50/0.16	0.42/0.23	0.43/0.21	0.33/0.18	0.96/0.32	0.53/0.21	0.97/0.26	0.89/0.27
Proposed (odom) (w/ removal)	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
Proposed (slam) (w/o removal)	0.61/0.17	3.26/0.55	0.64/0.18	1.08/0.57	0.50/0.16	0.30/0.09	0.34/0.13	0.40/0.15	0.96/0.32	0.62/0.12	0.97/0.26	0.88/0.25
Proposed (slam) (w/ removal)	0.55/0.14	1.02/0.18	0.62/0.15	0.92/0.48	0.43/0.14	0.27/0.07	0.35/0.09	0.26/0.12	0.92/0.29	0.61/0.10	0.94/0.25	0.63/0.18
LOAM [2]	0.78/-	1.43/-	0.92/-	0.86/-	0.71/-	0.57/-	0.65/-	0.63/-	1.12/-	0.77/-	0.79/-	0.84/-
IMLS-SLAM [3]	0.50/-	0.82/-	0.53/-	0.68/-	0.33/-	0.32/-	0.33/-	0.33/-	0.80/-	0.55/-	0.53/-	0.52/-
MC2SLAM [13]	0.51/-	0.79/-	0.54/-	0.65/-	0.44/-	0.27/-	0.31/-	0.34/-	0.84/-	0.46/-	0.52/-	0.52/-
MULLS [39]	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	0.52/0.13
SuMa [4]	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
SuMa++ [40]	0.64/0.22	1.60/0.46	1.00/0.37	0.67/0.46	0.37/0.26	0.40/0.20	0.46/0.21	0.34/0.19	1.10/0.35	0.47/0.23	0.66/0.28	0.70/0.29
LiTAMIN2 [17]	0.70/0.28	2.10/0.46	0.98/0.32	0.96/0.48	1.05/0.52	0.45/0.25	0.59/0.34	0.44/0.32	0.95/0.29	0.69/0.40	0.80/0.47	0.85/0.33
DeepLO [19]	-/-	-/-	-/-	-/-	-/-	-/-	-/-	-/-	3.06/0.89	1.90/0.80	1.80/1.00	-/-
Lo-Net [18]	0.78/0.42	1.42/0.40	1.01/0.45	0.73/0.59	0.56/0.54	0.62/0.35	0.55/0.33	0.56/0.45	1.08/0.43	0.77/0.38	0.92/0.41	0.83/0.42

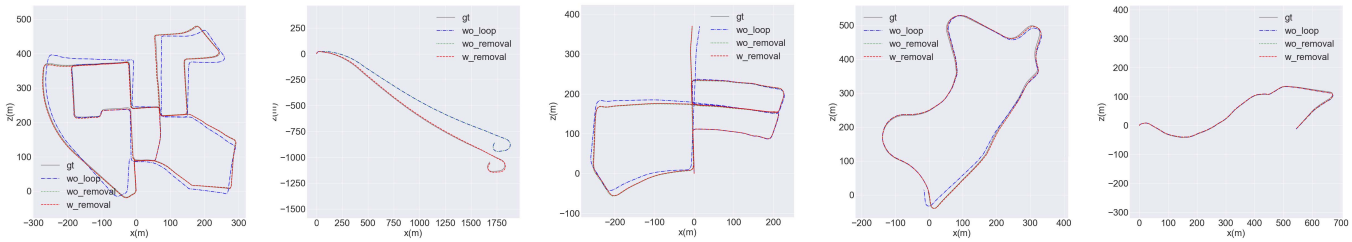


Fig. 8. Qualtitative results of the KITTI odometry benchmark (Seq `00`, `01`, `05`, `09`, `10`). Each plot includes the ground-truth trajectory (gray). The proposed method without both loop closure and dynamic removal (blue), the proposed method without dynamic removal (green) and the proposed full SLAM method with dynamic removal (red).

additional hardware and has better performance. In particular, our method shows better performance in Seq00 where several dynamic objects exist and in extreme dynamic environments (Seq `01`), such as highway environments. Fig. 8 shows the qualitative results from several sequences using the proposed method.

D. Evaluation on Our Dataset

As shown in Fig(a), our sensor system is equipped with 64-channel LiDAR, and high-precision INS for ground-truth trajectories. We manually classified the dynamic objects in the dataset utilizing the point cloud labeling tool [38]. The dataset includes 3 sequences with various dynamic objects and road conditions.

1) Dynamic Removal: This section evaluates the proposed system with our own datasets. As described for the KITTI dataset, we selected and labeled 200∼300 frames that include various dynamic objects to evaluate the dynamic removal performance.

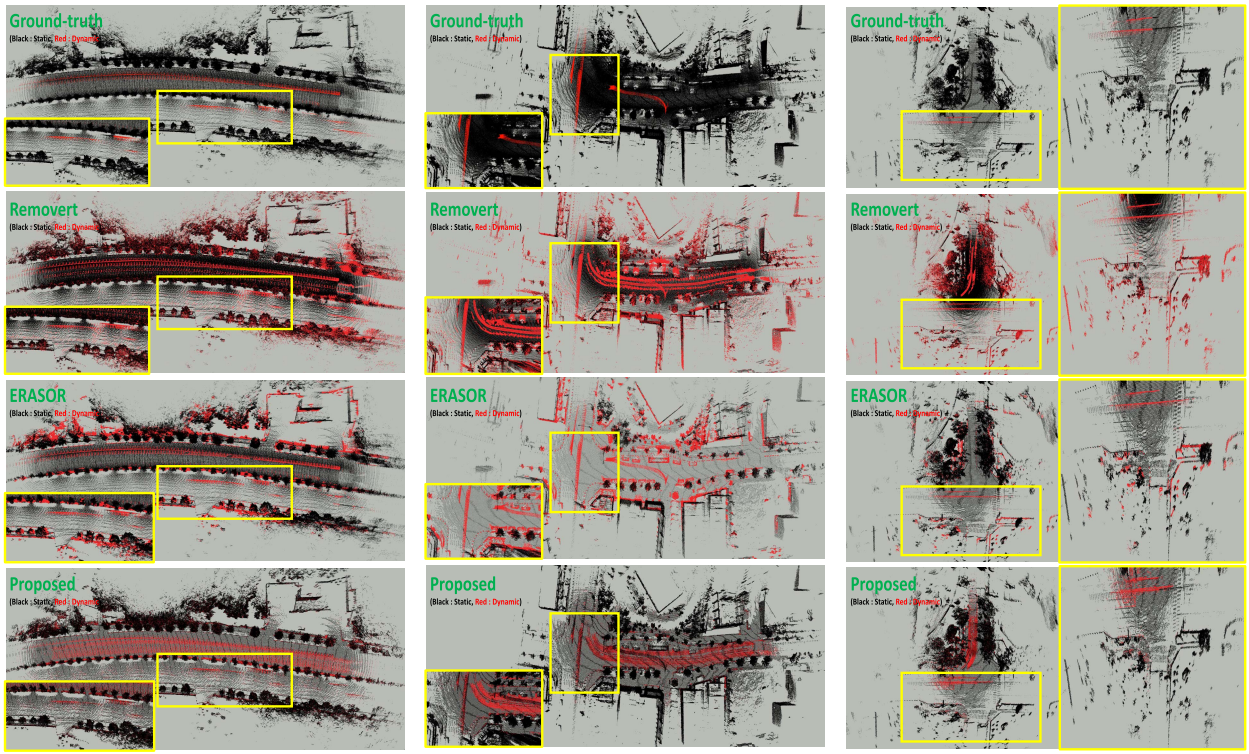
As shown as Table V, our method results in the best F1 score for all sequences. We primarily compare the proposed method to the baseline algorithms (ERASOR and Removert). Additionally, Fig. 9 represents the detailed evaluation on dynamic object detection using labeled point clouds. As described for the KITTI dataset, our method shows balanced and good performance compared to the ground-truth points. Because ERASOR and Removert misclassified static objects as dynamic objects, both methods showed a high RR but low PR. Fig. 9 shows this result well, and it can be seen that static objects or structures such as street trees and building walls are detected as dynamic objects.

TABLE V
COMPARISON RESULTS OF THE BACKGROUND MODEL ESTIMATOR USING OUR OWN DATASET

PR: Preservation Rate, RR: Rejection Rate

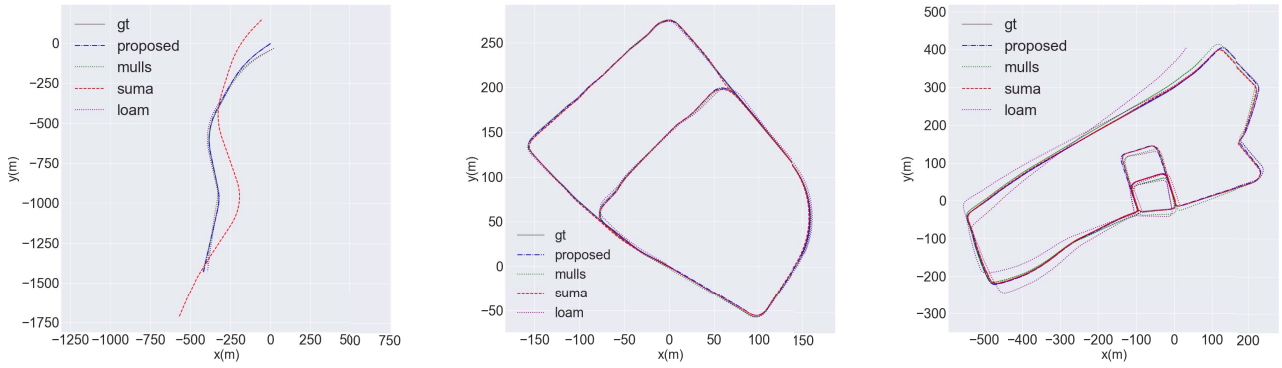
Seq.	Method	PR[%]	RR[%]	F_1 score
Juk-dong	Removert - `0.1` [5]	81.821	86.174	0.829
	ERASOR - `0.2` [6]	83.630	82.297	0.830
	Proposed(Ours)	**86.535**	**87.283**	**0.870**
Sinsung-dong	Removert - `0.1` [5]	80.219	91.581	0.839
	ERASOR - `0.2` [6]	80.730	**94.620**	0.871
	Proposed(Ours)	**84.928**	93.103	**0.882**
KIMM	Removert - `0.1` [5]	78.293	**97.714**	0.869
	ERASOR - `0.2` [6]	91.731	92.188	0.916
	Proposed(Ours)	**95.152**	90.091	**0.925**

2) LiDAR SLAM: As shown in Fig(a), the vehicle was equipped with LiDAR and INS. The INS guarantees cm-level accuracy, which is only used as the ground-truth. Based on the ground-truth measurement from the INS, we compute the absolute trajectory error. For a fair comparison, we compared single LiDAR-based methods, including loop closing, with the proposed method. The experiment was performed in various environments around the Korea Institute of Machinery and Materials (KIMM), including highway, small towns and mixed environments with highly dynamic objects.



(a) Juk-dong (b) Sinsung-dong (c) KIMM

Fig. 9. Comparison of background model estimation results of Juk-dong, Sinsung-dong and KIMM. Top to bottom: Ground-truth, Removert, ERASOR, proposed methods. The ground-truth classifies dynamic and static objects using semanticKITTI labeling tools, black point clouds indicate static objects and red point clouds indicate dynamic objects. The part corresponding to the yellow box is enlarged.



(a) Juk-dong (b) Sinsung-dong (c) KIMM

Fig. 10. Comparison result of estimated trajectory on. (a) Juk-dong, (b) Sinsung-dong, (c) KIMM dataset. Each plot includes the ground-truth trajectory (gray), the proposed method (blue), MULLS (green), SuMa (red) and LOAM (purple).

TABLE VI
COMPARISON OF ABSOLUTE TRAJECTORY ERRORS ON THE KIMM DATASET

	Proposed	MULLS [39]	SuMa [4]	LOAM [2]
Juk-Dong	3.624	32.8481	-	12.672
Sinsung-Dong	0.797	1.343	1.283	2.5413
KIMM	3.758	9.443	5.554	18.786

Fig. VI-D.2 shows the estimated trajectory and its comparison to the ground-truth in our dataset. Fig(a) shows the trajectory estimation on highway, and Fig(b) shows the estimation in the small downtown areas. Finally, the experimental results in a mixed environment of internal and external highways around our research institute are shown in Fig(c). The proposed method shows the best performance among all methods. Because SuMa does not consider dynamic objects, it performs poorly in highly dynamic environments. For MULLS, the KITTI dataset shows the best performance, but our dataset shows low accuracy. This means that the other methods are not guaranteed to have generality because they define features with strict criteria such as planes and edges. Also, LOAM does not include a loop closure, so it shows the lowest performance in the rest of the experiments except for Juk-Dong.

Table VI shows the quantitative results for the absolute trajectory error. The proposed method shows an error of

3.624 m on the Juk-dong dataset. It shows excellent results despite the highway environment with many dynamic vehicles. On the other hand, SuMa without considering dynamic objects completely fails motion estimation in this sequence. LOAM does not guarantee global consistency because it does not include loop closure, so it has the lowest accuracy in Sinsung-Dong and KIMM. However, both methods can be further improved through loop closure or dynamic removal. In all sequences, the proposed method shows the best performance. The proposed method shows an absolute trajectory error of 0.797 m in Sinsung-dong, a small downtown environment. Dynamic objects such as vehicles and pedestrians are effectively eliminated, and loop closure dramatically improves the global accuracy. The final experimental results show errors of 3.753m. These results indicate that the proposed method shows excellent results in terms of changes in speed and more diverse mixed environments.

VII. Conclusion

In this work, we introduce a robust LiDAR SLAM method for highly dynamic environments leveraging nonparametric dynamic removal. The proposed method effectively performs dynamic removal in terms of mapping and shows that it can improve SLAM performance by classifying static scenes and dynamic objects. The nonparametric dynamic removal method can effectively remove dynamic objects from a projective spherical range image. In addition, the identified dynamic objects are reflected in the cost function for motion estimation to further improve the performance of SLAM. Our method shows that global accuracy is also guaranteed in various real-world environments by applying the full SLAM framework.

Although the proposed method can effectively perform dynamic removal using spherical range images, it tends to depend somewhat on the performance of ground segmentation. The vertical angle resolution of 3D LiDAR can also affect the performance of algorithms based on spherical image representation. In this work, we only presented the performance based on a 64-layer LiDAR with a vertical angle resolution of approximately 0.4 degree. The proposed method can be extended to target LiDAR with lower resolution. Additionally, estimating the exact feature or the normal of an exact point can be a way to improve the SLAM performance of the proposed method.

In the future work, we plan to extend our approach to multi-session long term SLAM. In the case of multisession SLAM, it is necessary to consider not only instantaneous dynamic objects but also dynamic space from a long-term perspective. In addition, the application of recent deep learning-based semantic or panoptic segmentation methodologies can be considered. These are meant to be more robust and extend to high-level SLAM.

References

[1] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LiDAR-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, Mar. 2019, Art. no. 172988141984153.

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

[3] J.-E. Deschaud, "IMLS-SLAM: Scan-to-model matching based on 3D data," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2480–2485.

[4] 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.

[5] 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.

[6] 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.

[7] C. Yin *et al.*, "Removing dynamic 3D objects from point clouds of a moving RGB-D camera," in *Proc. IEEE Int. Conf. Inf. Autom.*, Aug. 2015, pp. 1600–1606.

[8] 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.

[9] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* Berlin, Germany: Springer, Jun. 2000, pp. 751–767.

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

[11] 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)*, vol. 3, Oct. 2003, pp. 2743–2748.

[12] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Auto. Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[13] F. Neuhaus, T. Koß, R. Kohnen, and D. Paulus, "MC2SLAM: Real-time inertial LiDAR odometry using two-scan motion compensation," in *Proc. German Conf. Pattern Recognit.* Springer, 2018, pp. 60–72.

[14] J. Serafin and G. Grisetti, "NICP: Dense normal based point cloud registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 742–749.

[15] E. Takeuchi and T. Tsubouchi, "A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 3068–3073.

[16] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, "Evaluation of 3D registration reliability and speed—A comparison of ICP and NDT," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3907–3912.

[17] 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. 1–7.

[18] Q. Li *et al.*, "LO-Net: Deep real-time LiDAR odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8473–8482.

[19] Y. Cho, G. Kim, and A. Kim, "Unsupervised geometry-aware deep LiDAR odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2145–2152.

[20] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] W. Xiao, B. Vallet, and N. Paparoditis, "Change detection in 3D point clouds acquired by a mobile mapping system," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 1, no. 2, pp. 331–336, 2013.

[22] B. Vallet, W. Xiao, and M. Brédif, "Extracting mobile objects in images using a velodyne LiDAR point cloud," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 2, no. 3, p. 247, 2015.

[23] 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.

[24] P. Ruchti and W. Burgard, "Mapping with dynamic-object probabilities calculated from single 3D range scans," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 6331–6336.

[25] S. Pagad, D. Agarwal, S. Narayanan, K. Rangan, H. Kim, and G. Yalla, "Robust method for removing dynamic objects from point clouds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 10765–10771.

[26] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–8.
[27] 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.
[28] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robot.*, vol. 34, no. 3, pp. 189–206, 2013.
[29] J. Schauer and A. Nuchter, "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.
[30] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.
[31] D. J. Meagher, "Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-D objects by computer," Elect. Syst. Eng. Dept., Rensseiaer Polytech., Troy, NY, USA, Tech. Rep. IPL-TR-80-111, 1980.
[32] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3D reconstruction in dynamic scenes using point-based fusion," in *Proc. Int. Conf. 3D Vis.*, Jun. 2013, pp. 1–8.
[33] H. Badino, D. Huber, Y. Park, and T. Kanade, "Fast and accurate computation of surface normals from range images," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3084–3091.
[34] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 3748–3754.
[35] Y.-S. Shin, Y. S. Park, and A. Kim, "DVL-SLAM: Sparse depth enhanced direct visual-LiDAR SLAM," *Auto. Robots*, vol. 44, no. 2, pp. 115–130, Jan. 2020.
[36] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4802–4809.
[37] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "ISAM2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, Feb. 2012.
[38] 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. 9297–9307.
[39] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "MULLS: Versatile LiDAR SLAM via multi-metric linear least square," 2021, *arXiv:2102.03771*.
[40] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4530–4537.

Joohyun Park (Member, IEEE) received the B.S. degree in electronics and computer engineering from Chonnam National University, Gwangju, Republic of Korea, in 2018, and the M.S. degree from the Robotics Program, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2020. He was a Senior Researcher with TWINNY, Daejeon, from 2020 to 2022. Since 2022, he has been a Researcher Engineer with NAVER LABS, Seongnam-si, South Korea. His current research interests include simultaneous localization and mapping, place recognition, navigation, high-definition map generation, and autonomous vehicle.

Younggun Cho (Member, IEEE) received the B.S. degree in electrical engineering from Inha University, Incheon, South Korea, in 2013, and the M.S. degree in electrical engineering and the Ph.D. degree in civil and environmental engineering with dual degree from the Robotics Program, Korea Advanced Institute of Science & Technology (KAIST), Daejeon, South Korea, in 2015 and 2020, respectively. He was an Assistant Professor with the Department of Robot Engineering, Yeungnam University. He is currently an Assistant Professor with the Department of Electrical Engineering, Inha University. His current research interests include robust sensing, long-term autonomy, and visual simultaneous localization and mapping.

Young-Sik Shin (Member, IEEE) received the B.S. degree in electrical engineering from Inha University, Incheon, South Korea, in 2013, and the M.S. and Ph.D. degrees in civil and environmental engineering with a dual degree from the Robotics Program, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2015 and 2020, respectively. He is currently a Senior Researcher with the Korea Institute of Machinery and Materials (KIMM), Daejeon. His research interests include robust simultaneous localization and mapping and navigation using heterogeneous sensors.