HeLiMOS: A Dataset for Moving Object Segmentation in 3D Point Clouds From Heterogeneous LiDAR Sensors

Hyungtae Lim[1†], Seoyeon Jang[1†], Benedikt Mersch[2], Jens Behley[2], Hyun Myung[1*], and Cyrill Stachniss[2‡]

***Abstract*— Moving object segmentation (MOS) using a 3D light detection and ranging (LiDAR) sensor is crucial for scene understanding and identification of moving objects. Despite the availability of various types of 3D LiDAR sensors in the market, MOS research still predominantly focuses on 3D point clouds from mechanically spinning omnidirectional LiDAR sensors. Thus, we are, for example, lacking a dataset with MOS labels for point clouds from solid-state LiDAR sensors. In this paper, we present a labeled dataset, called *HeLiMOS*, that enables to test MOS approaches on four heterogeneous LiDAR sensors, including two solid-state LiDAR sensors. Furthermore, we introduce a novel automatic labeling method to substantially reduce the labeling effort required from human annotators. To this end, our framework exploits an instance-aware static map building approach and tracking-based false label filtering. Finally, we provide experimental results regarding the performance of commonly used state-of-the-art MOS approaches on HeLiMOS that suggest a new direction for a sensor-agnostic MOS, which generally works regardless of the type of LiDAR sensors used to capture 3D point clouds.**

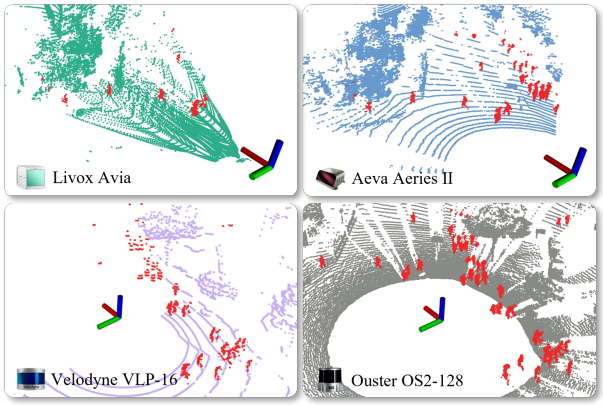


Fig. 1. Qualitative examples of our dataset, called *HeLiMOS*. Our dataset provides point-wise moving object segmentation (MOS) annotations for point clouds acquired by heterogeneous 3D LiDAR sensors from the HeLiPR dataset [18]. Red points indicate the annotated points from moving objects (best viewed in color).

I. Introduction

Robots need to understand their surroundings, including moving objects, to navigate and act safely. By doing so, robots can avoid collisions, optimize paths, and make informed decisions based on dynamic changes around them. As one of the solutions, moving object segmentation (MOS) with 3D light detection and ranging (LiDAR) sensors has been extensively studied, aiming to identify moving objects [1]–[10]. By distinguishing between moving objects such as buses, cars, and pedestrians, and static objects such as buildings, walls, and trees, MOS can enhance path planning and collision avoidance, but also prevent traces of moving objects, which we call *dynamic points*, from being left in a 3D point cloud map by filtering out these undesirable points at the perception level [11]–[16].

Meanwhile, diverse types of 3D LiDAR sensors have been developed, spanning from mechanically spinning omnidirectional LiDAR to solid-state LiDAR sensors. It should be noted that we classify them by the scanning mechanism and do not include the flash type because of the poor resolution by now. With the growing need for datasets to evaluate existing approaches across heterogeneous LiDAR sensor setups, some researchers [17], [18] proposed novel large-scale datasets captured by heterogeneous LiDAR sensor setups. In addition, Mersch *et al*. [6] and Wu *et al*. [7] showed pioneering works by demonstrating the feasibility of MOS with heterogeneous LiDAR configurations.

Despite these efforts, we see that existing public datasets have two limitations for evaluating the generalization capabilities of MOS across heterogeneous LiDAR sensor setups. First, the aforementioned heterogeneous LiDAR datasets mainly focus on evaluating place recognition [18] or pose estimation [17] without providing point-wise MOS labels. Second, while multiple datasets that provide point-wise MOS labels exist [19], [20], these datasets are only acquired by a single omnidirectional LiDAR sensor. Thus, publicly available datasets with point-wise MOS labels for heterogeneous LiDAR setups are still lacking.

To tackle the insufficiency of MOS labels for heterogeneous LiDAR sensors, as shown in Fig. 1, we build upon built upon the existing *HeLiPR* dataset [18] and provide MOS labels that enable the evaluation of MOS across diverse heterogeneous LiDAR sensor setups, which we call *HeLiMOS*. Furthermore, sharing the philosophy of the state-of-the-art automatic MOS labeling framework [1], we propose a novel instance-aware automatic labeling framework to substantially reduce the time needed for manual labeling. Finally, as a preliminary step, we set up benchmarks for evaluating MOS

*Corresponding author: Hyun Myung (E-mail: hmyung@kaist.ac.kr)
†The authors are equally contributed.
[1]Hyungtae Lim, Seoyeon Jang, and Hyun Myung are with the School of Electrical Engineering, KAIST (Korea Advanced Institute of Science and Technology), Daejeon, Republic of Korea.
[2]Benedikt Mersch, Jens Behley, and Cyrill Stachniss are with the Center for Robotics, University of Bonn, Germany (E-mail: cyrill.stachniss@igg.uni-bonn.de)
‡Cyrill Stachniss is additionally with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.
This work was supported in part by Korea Evaluation Institute of Industrial Technology (KEIT) funded by the Korea Government (MOTIE) under Grant No.20018216, Development of mobile intelligence SW for autonomous navigation of legged robots in dynamic and atypical environments for real application and in part by the European Union's Horizon Europe research and innovation programme under grant agreement No 101070405 (DigiForest). The Korean students are supported by the BK21 FOUR, Republic of Korea.

from an egocentric perspective and static map building from a map-centric perspective.

In summary, our main contributions are threefold:

- We provide point-wise annotations for a sequence of the HeLiPR dataset, which are captured by real-world multiple heterogeneous LiDAR sensors.
- We propose an efficient instance-aware automatic labeling framework by employing an instance-aware static map building approach, ERASOR2 [15], and tracking-based false label filtering [21]. We also make these MOS labeling tools publicly available.
- We evaluate state-of-the-art MOS approaches with heterogeneous LiDAR sensor setups as initial benchmarks.

We believe this dataset will stimulate further research, suggest new research directions, and enable reliable evaluation of novel algorithms.

II. Related Work

Over the past decade, numerous impactful datasets for autonomous vehicles have been released, providing novel benchmarks. One of the renowned datasets is the KITTI dataset [22], which provides both odometry and various perception benchmarks. Influenced by the KITTI dataset, existing datasets have evolved in two main directions in terms of (a) odometry and place recognition and (b) perception, as presented in Table I.

From the viewpoint of odometry and place recognition, the KITTI dataset has few loop closing situations and environmental changes, with only a single omnidirectional LiDAR sensor for a short data collection span. To provide more challenging environments for odometry and place recognition tasks [23], Carlevaris *et al*. [24] and Jeong *et al*. [25] proposed the NCLT and Complex Urban datasets, respectively, acquired by multiple 2D and 3D omnidirectional LiDAR sensors. Kim *et al*. [26] proposed the MulRan dataset focusing on multi-modal long-term mapping and place recognition by employing a 3D LiDAR sensor and an omnidirectional radar sensor.

As a further study, Carballo *et al*. [27] proposed the LIBRE dataset, which consists of point clouds from ten different omnidirectional LiDAR sensors. However, all the deployed sensors are omnidirectional LiDAR sensors, implying that all the sensors are homogeneous. Thus, this dataset is not available to test whether an algorithm generally works well in the heterogeneous LiDAR sensor suites. To tackle this problem, Qingqing *et al*. [17] proposed the TIERS dataset, which consists of three omnidirectional LiDAR sensors and three solid-state LiDAR sensors. Similar to the TIERS dataset, Jung *et al*. [18] proposed the HeLiPR dataset, which is acquired by two omnidirectional LiDAR sensors and two solid-state LiDAR sensors, including under-researched channels, i.e. reflectivity, near-infrared, and radial velocity. Unfortunately, these datasets only aim to evaluate odometry and place recognition, without any point-wise labels, as summarized in Table I.

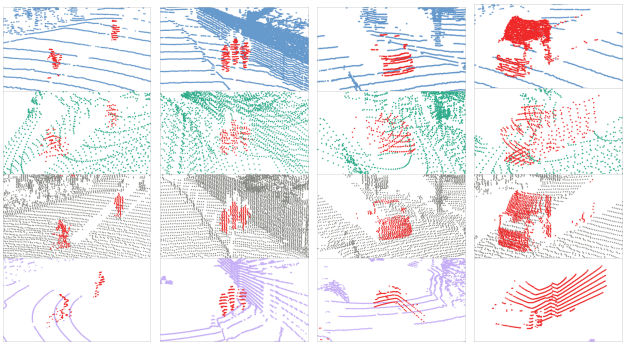
Regarding perception, Behley *et al*. [19] proposed the SemanticKITTI dataset, a pioneering work that first provides point-wise semantic, instance, and MOS labels for 3D sequential point clouds. Inspired by the SemanticKITTI, Pan *et al*. [20] proposed the SemanticPOSS dataset, which shares exactly the same labeling protocol with SemanticKITTI to support compatibility with existing SemanticKITTI dataloaders. While these datasets provide abundant point-wise labels, the SemanticKITTI and SemanticPOSS are only captured by a single omnidirectional LiDAR sensor.

In recent years, Caesar *et al*. [29] proposed the nuScenes dataset, which supports various perception tasks in 1,000 sequences. Ettinger *et al*. [30], Xiao *et al*. [31], and Chen *et al*. [32] proposed the WOMD, PandaSet, and WOMD-LiDAR datasets, respectively, which contain point clouds from multiple heterogeneous LiDAR sensors. However, these datasets are also inappropriate to evaluate the performance of MOS in the heterogeneous LiDAR sensor setups because they do not provide point-wise MOS labels. Therefore, to the best of our knowledge, we first propose a point-wise MOS dataset for heterogeneous LiDAR sensors, enabling the evaluation of MOS and static map building tasks across diverse LiDAR sensor setups.

Furthermore, we propose an efficient instance-aware automatic labeling framework to substantially lessen the annotation burden of a human labeler. It is challenging and time-consuming for human labelers to discern moving objects in the 3D point clouds owing to the sparse characteristics of 3D point clouds [33]. To account for this, Kim and Kim [13] proposed Removert, which is a range image-based scan-wise MOS labeling approach. Furthermore, Chen *et al*. [1] proposed an automatic labeling framework called Auto-MOS. In contrast to these prior approaches, we take instance information into account to reduce the number of false positives and thus minimize the need for manual corrections by a human labeler. Thus, we propose instance-aware MOS annotation using ERASOR2 [15], while accounting for the

TABLE I. Comparison between existing 3D point cloud datasets and our proposed dataset. The term *Hetero* indicates whether a dataset comprises both mechanically spinning omnidirectional and solid-state LiDAR sensors. We consider 2D and 3D omnidirectional LiDAR sensors to be homogeneous to each other. The symbol △ indicates that the dataset provides point-wise labels; however, it incorrectly labels parked vehicles as moving objects by naïvely considering all pedestrians and vehicles as in motion.

	Dataset	Year	Multiple LiDARs	Hetero	Point-wise MOS labels
Odometry & place recognition	KITTI [22]	2012	✗	✗	✗
	NCLT [24]	2016	✗	✗	✗
	Oxford Robotcar [28]	2017	✓	✗	✗
	Complex Urban [25]	2019	✓	✗	✗
	MulRan [26]	2020	✗	✗	✗
	LIBRE [27]	2020	✓	✗	✗
	TIERS [17]	2022	✓	✓	✗
	HeLiPR [18]	2023	✓	✓	✗
Perception	KITTI [22]	2012	✗	✗	✗
	SemanticKITTI [19]	2019	✗	✗	✓
	SemanticPOSS [20]	2020	✗	✗	△
	nuScenes [29]	2020	✗	✗	✗
	WOMD [30]	2021	✓	✓	✗
	PandaSet [31]	2021	✓	✓	✗
	WOMD-LiDAR [32]	2023	✓	✓	✗
	HeLiMOS (Ours)	2024	✓	✓	✓



(a) (b) (c) (d)

Fig. 2. Examples of moving objects in our dataset, which are shown as red points. (T-B): Zoomed point clouds captured by Aeva Aeries II, Livox Avia, Ouster OS2-128, and Velodyne VLP-16. Note that even though the same objects are shown, they have different patterns owing to the difference in scanning techniques and field of views of the sensors. MOS labels of (a) a bicyclist and pedestrian, (b) crowded pedestrians, (c) a car, and (d) a truck (best viewed in color).

pose uncertainty in the revisited scenes via a topology-based trajectory clustering approach.

III. Instance-Aware Automatic Labeling and Data Statistics

Our dataset is based on the `KAIST05` sequence of HeLiPR dataset [18], which contains various moving objects, such as buses, pedestrians, bicyclists, and cars, different from sequences (see Fig. 2). The dataset is acquired by four LiDAR sensors: Velodyne VLP-16 and Ouster OS2-128 as omnidirectional LiDAR sensors, and Livox Avia and Aeva Aeries II as solid-state LiDAR sensors. For brevity, we denote these sensors as Velodyne (`V`), Ouster (`O`), Livox (`L`), and Aeva (`A`) in this paper, respectively.

Our goal is to provide a point-wise label for each point in the point clouds of all the LiDAR sensors. Thus, we propose a merging-and-splitting-based efficient automatic MOS labeling framework, as illustrated in Fig. 3. Our approach mainly consists of four steps. First, we accumulate four point clouds from the four LiDAR sensors whose time steps are closest to each other by transforming them into the Ouster frame. By doing so, we synchronize the point clouds of these four LiDAR sensors at a software level, which is denoted by $\pi(\cdot)$ in Fig. 3. In addition, we represent the accumulated point cloud by $\mathcal{P}_t$ in Fig. 3(a). Second, initial MOS labels are automatically annotated by our proposed automatic labeling framework, as presented in Figs. 3(b) to (d). Third, we manually correct the labels under human supervision. Fourth, we backpropagate the refined MOS labels to the individual point clouds, as depicted in Figs. 3(f) and (g). The details are explained in the following subsections.

A. Topology-Based Trajectory Clustering and Submap-Based Pose Correction

In recent static map building approaches [14], [15], discrepancies in geometry or occupancy between individual scans and the map have often been used to estimate the dynamic points in the scans. However, these approaches heavily rely on the assumption that the given poses are accurate and thus the scans are sufficiently well-aligned with each other. Unfortunately, we have found that even though provided (near) ground truth poses are used, undesirable errors exist in the poses for revisited scenes, i.e. loop-closed scenes. These pose errors probably stem from systematic GNSS errors or potential errors arising from the process of aligning four point clouds because the point clouds were not originally synchronized at the hardware level. Consequently, these errors make automatic labeling incorrectly classify static points as dynamic points, leading to many false positives and false negatives.

To address this issue, we divide the trajectory with poses corresponding to $\mathcal{P}_t$ into multiple clusters and correct their poses to align their reference frames. The positions of the trajectory are neither dense nor have geometrical features, making existing clustering methods not work [34]. For this reason, as presented in Fig. 4, we propose topology-based trajectory clustering that prioritizes revisited sections, which are likely to have inherent pose errors owing to the significant time differences between scans taken during initial visits and those upon revisiting. This is because time discrepancies can lead to pose drift, which may not be fully minimized even after pose graph optimization.

As illustrated in Fig. 4, our trajectory clustering follows three steps. First, we identify areas, such as intersections or places where left/right turns occur, by examining the yaw differences within the trajectory and then group neighboring frames into a cluster based on their position values as inputs. Second, places that are revisited but not intersections and the unclustered frames with sufficiently large frame intervals are clustered. Finally, the remaining unclustered frames are merged into the adjacent cluster with the closest frame interval.

Next, poses corresponding to frames within the same cluster are corrected to minimize errors between the reference frames for each subcluster. Formally, let $\mathcal{C}$ be a cluster of the trajectory and the n-th consecutive frame set (or a subcluster) in $\mathcal{C}$ be $\mathcal{C}_n$, which satisfies $\mathcal{C} = \bigcup_{n=1}^{N_c} \mathcal{C}_n$, as visualized in Fig. 4(d); $N_c \geq 1$ denotes the number of the subclusters. By denoting the transformation matrix of the t-th body frame with respect to the reference frame w by $\mathbf{T}_t^w$, the n-th submap of the each subcluster $\mathcal{S}_n$ is defined as follows:

$$\mathcal{S}_n = \nu\Bigg(\bigcup_{t \in \mathcal{C}_n} \nu\Big(\{ \mathbf{T}_t^w \mathbf{p} \mid \mathbf{p} \in \mathcal{P}_t \} \Big) \Bigg), \tag{1}$$

where $\nu(\cdot)$ denotes a voxel sampling function with the voxel size ν, $\mathcal{P}_t$ is the synced scan whose origin is the t-th body frame, and $\mathbf{T}_t^w \mathbf{p}$ means that a point $\mathbf{p}$ is transformed into the reference frame w.

Based on the assumption that the poses in $\mathcal{C}_n$ are locally consistent, the inherent error is modeled as $\mathbf{T}_t^w = \mathbf{T}_{w_{\text{true}}}^{e_n} \mathbf{T}_t^{w_{\text{true}}}$, where $\mathbf{T}_{w_{\text{true}}}^{e_n}$ denotes the error between the actual global reference frame w_{true} and erroneous reference frame e_n of $\mathcal{C}_n$. Consequently, to locally unify the coordinate system into the reference frame of $\mathcal{S}_1$, i.e. e_1 frame, we apply submap-to-submap ICP between $\mathcal{S}_1$ and $\mathcal{S}_n$ to estimate relative transformation $\hat{\mathbf{T}}_{e_n}^{e_1}$ and all the poses of $\mathcal{C}_n$ are

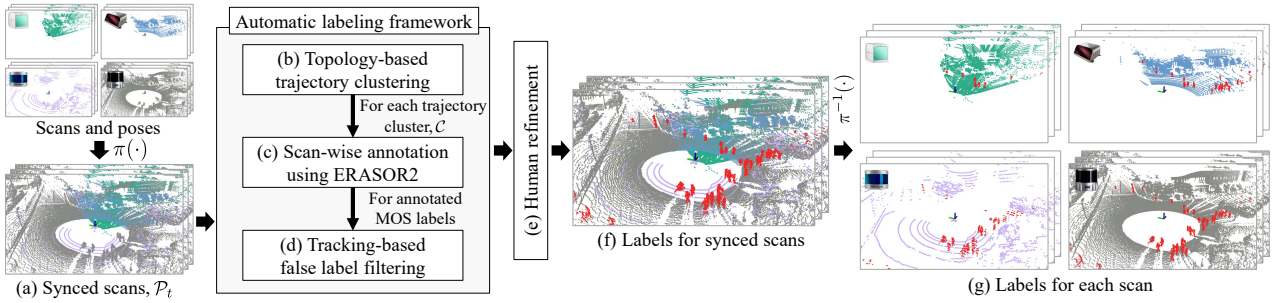


Fig. 3. Overview of our merging-and-splitting-based labeling framework. (a) Synchronization of the point clouds from the four LiDAR sensors at a software level. (b)-(d) Procedure of our proposed automatic labeling framework. (b) First, trajectories are segmented into multiple clusters. (c) For each trajectory cluster $\mathcal{C}$, we apply an instance-aware static map building, ERASOR2 [15], that produces initial scan-wise annotated labels. (d) Tracking-based false label filtering is applied to reduce false positive and false negative MOS labels. (e) Next, these labels are manually corrected under human supervision. (f)-(g) Finally, the refined labels of synced scans are backpropagated to individual point clouds, which is denoted by $\pi^{-1}(\cdot)$. Red points indicate the annotated dynamic points (best viewed in color).

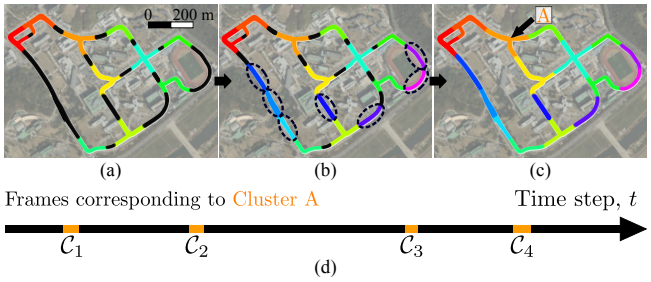


Fig. 4. (a)-(c) Procedure of our topology-based trajectory clustering. Black trajectory indicates unclustered frames and each color represents a different cluster (best viewed in color). (a) First, intersections are prioritized because these scenes are highly likely to have multiple revisits. (b) Next, revisited places yet are not intersections and the unclustered frames with sufficiently large frame intervals are clustered, as indicated by the black dashed circles. (c) Each unclustered frame is merged into the adjacent cluster with the closest frame interval. (d) Frames included in Cluster A, which is indicated in (c), visualized along the time step axis. As a result of the clustering, several sets of consecutive frames are clustered together.

updated as $\mathbf{T}_t^w \leftarrow \hat{\mathbf{T}}_{e_n}^{e_1} \mathbf{T}_t^w$, respectively. Thus, ICP is performed $N_c - 1$ times for each cluster.

B. Instance-Aware Initial Data Annotation

Next, by taking the corrected poses and corresponding synced scans of $\mathcal{C}$ as inputs, our instance-aware annotation pipeline is applied to generate initial scan-wise MOS labels by utilizing instance segmentation information [15], which corresponds to Fig. 3(c). The main difference between our approach and the previous automatic labeling approach is that Chen *et al*. [1] employed ERASOR [14] to initially annotate MOS labels and then clustering is applied, which is referred to as a *detect-then-cluster* scheme. As ERASOR does not account for instance information, it potentially fails to reject whole dynamic points from a moving object, considering some partial dynamic points as static.

In contrast, our *cluster-then-detect* approach first performs instance segmentation, followed by dynamic point removal at the instance level using the obtained instance information. By doing so, we can generate more accurate and reliable MOS labels.

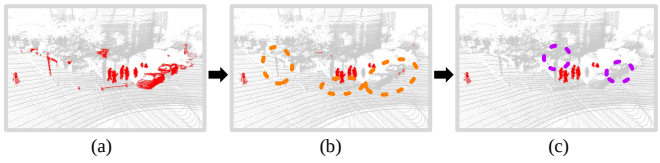


Fig. 5. (a)-(c) The annotation results in our proposed labeling framework. Red points denote the annotated dynamic points, while gray points represent points estimated to be static (best viewed in color). (a) The initial result obtained by using ERASOR2, which is an instance-aware static map building approach [15]. (b) Refined annotation through our tracking-based filtering. Orange dashed circles indicate that false positive points are successfully rejected. (c) Final annotation after human supervision. Purple dashed circles highlight the refined areas by a human labeler.

C. Multi-Object Tracking-Based False Label Filtering and Human Refinement

The so far detailed static map building approach-based automatic labeling is likely to remove dynamic points somewhat aggressively because static map building approaches are originally designed to preserve definite static points for performing localization or navigation. For this reason, as presented in Fig. 5(a), many static points are wrongly classified as dynamic points at the scan level. To address this issue, we leverage multi-object tracking-based filtering [21]. In contrast to Chen *et al*. [1], who also employed tracking-based filtering but primarily focused on reducing false positive points, we propose a bounding box augmentation to reduce the number of false negative points. That is, we augment additional bounding boxes in the frames where tracking is temporarily lost by interpolating the centroids of bounding boxes tracked in the previous frame and next frame. Subsequently, points within these augmented bounding boxes are also classified as dynamic points and thus are successfully rejected. As a result, more refined MOS labels can be obtained without human effort, as shown in Fig. 5(b).

Nevertheless, these procedures do not perfectly reject all false positives and negatives. Therefore, as a final stage, we perform a human-in-the-loop refinement process to enhance the quality of the MOS labels, as depicted in Fig. 5(c).

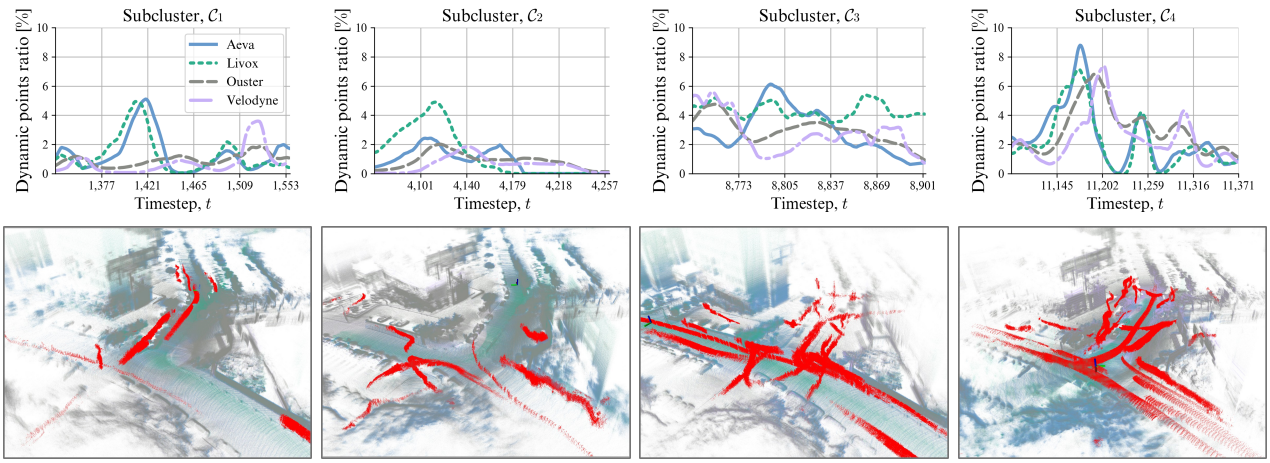


Fig. 6. (T-B, L-R): Dynamic points ratios, each of which is defined as $\frac{\text{\# of labeled dynamic points}}{\text{\# of total points of the } t\text{-th scan}}$, over time steps and the visualized cleaned point cloud maps of four subclusters, corresponding to $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$ in Fig. 4(d). Because the original dataset [18] targets place recognition, our dataset features a variety of dynamic point patterns owing to the varying trajectories of moving objects even though the scans are acquired in the same places. Red points indicate the annotated points, which are traces of moving objects (best viewed in color).

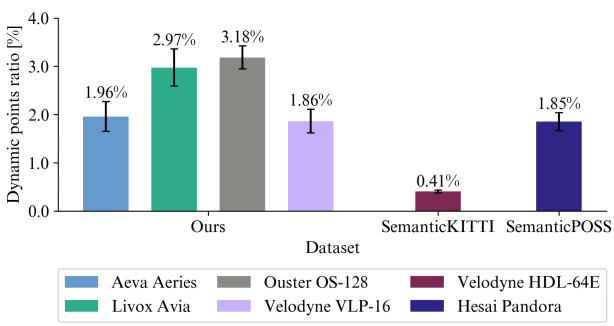


Fig. 7. Comparison of dynamic points ratio with other datasets. The numbers on the bars represent the average ratios, while the black error bars indicate the standard deviations. For calculating the dynamic points ratio of SemanticKITTI [19], we counted the points labeled by moving objects. As described in Table I, the SemanticPOSS [20] wrongly classifies parked vehicles as moving objects. Thus, for a fair comparison, we filtered out them using our tracking-based filtering and only used the actual moving objects for the dynamic points ratio calculation.

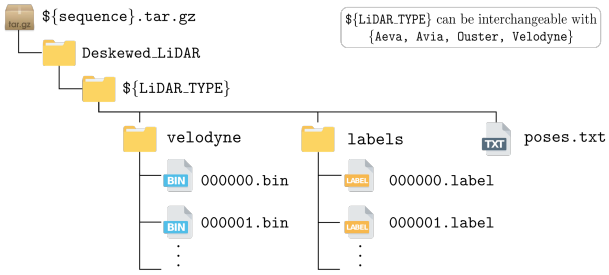


Fig. 8. File structure of our dataset, which follows the SemanticKITTI format [19]. It may seem awkward to use the folder name *velodyne* instead of *scans*, but we adhere to the convention of using *velodyne* as it is used in other datasets, such as SemanticPOSS [20]. Pose information is from the original dataset [18]

D. Data Statistics and File Structure

Our dataset provides a total of 12,188 labeled point clouds. Each MOS label follows the SemanticKITTI-MOS format, so it consists of three classes: *unlabeled*, *static*, and *dynamic*. Furthermore, as shown in Figs. 6 and 7, two distinctive features of our dataset are (a) the inclusion of several revisited scenes because the original dataset, the HeLiPR dataset [18], is used for place recognition, and (b) a significantly higher ratio of dynamic points compared with the existing MOS datasets.

As shown in Fig. 6, we can observe that dynamic points ratios and dynamic point patterns vary significantly over time, even though scans are acquired in the same place. For instance, by using the previously mentioned clusters, $\mathcal{C}_1$ and $\mathcal{C}_2$ show relatively low dynamic points ratios compared with $\mathcal{C}_3$ and $\mathcal{C}_4$, which implies that the scenes include fewer dynamic points. Conversely, $\mathcal{C}_3$ and $\mathcal{C}_4$ contain a higher number of moving objects and more complex trajectory patterns, resulting in higher dynamic points ratios compared with $\mathcal{C}_1$ and $\mathcal{C}_2$. In addition, owing to the different field of views of each sensor, the scanning patterns of static scenes also become different, as presented in Fig. 6. Therefore, our dataset can provide an opportunity to evaluate the generalization capabilities of MOS across diverse patterns in the same scene.

Furthermore, note that the most distinctive feature of our dataset is that it not only has higher dynamic points ratios than existing MOS datasets, but also has MOS labels of four heterogeneous LiDAR sensors. As presented in Fig. 7, our dataset shows consistently higher average dynamic points ratios across all LiDAR sensors compared with the SemanticKITTI [19] and SemanticPOSS [20] datasets.

Therefore, by using our dataset, researchers can evaluate the generalization capabilities of MOS approaches against untrained environments and different types of LiDAR sensors. As presented in Fig. 8, the file structure of our framework follows the SemanticKITTI format [19] to support compatibility with existing SemanticKITTI dataloaders. All the laser scans are deskewed and then saved by utilizing HeLiPR Pointcloud Toolbox[1]. Next, we split the dataset into training, validation, and test sets with ratios of 68%, 16%,

[1]https://github.com/minwoo0611/HeLiPR-Pointcloud-Toolbox

and 16%, respectively. Note that we do not randomly sample the frames; instead, we designate certain sequential frames from the revisited scenes, e.g. $\mathcal{C}_3$ or $\mathcal{C}_4$ in Fig. 6, for the validation and the test sets.

IV. Evaluation of Moving Object Segmentation and Static Map Building

The main focus of this work is to provide point-wise MOS labels for evaluating the generalization capabilities of MOS in heterogeneous LiDAR sensor setups. In addition, our dataset can be utilized to evaluate the performance of static map building approaches. Thus, we present three experiments by utilizing our dataset: (i) MOS performance of the models trained on the SemanticKITTI dataset [19] against both environmental changes and LiDAR sensor type variations, (ii) MOS performance across heterogeneous LiDAR sensors, and (iii) automatic labeling performance to support the rationale behind our choice to use ERASOR2 and the proposed tracking-based filtering. These novel experiments, which could not be evaluated using existing datasets, back up our key claim of the necessity of the heterogeneous LiDAR MOS dataset and our automatic labeling framework.

A. Experimental Setup

In the first experiment, we use the pre-trained MOS models on the SemanticKITTI dataset [19], which is captured by a 64-channel omnidirectional LiDAR sensor, and then quantitatively evaluate the inference results of the models by using all the labels. In the second experiment, we train MOS approaches on one type of LiDAR sensors and then test on heterogeneous LiDAR sensors, i.e. training with point clouds from solid-state LiDAR and testing with those from omnidirectional LiDAR sensors, or vice versa, to examine performance variations across different LiDAR types.

As a quantitative metric, we use the intersection-over-union (IoU) metric for MOS [5], IoU_{MOS}, which is defined as follows:

$$\text{IoU}_{\text{MOS}} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{2}$$

where TP, FP, and FN denote the true positive, false positive, and false negative points from the perspective of MOS, respectively.

For the third experiment, we evaluate the modules of our labeling framework and existing approaches by using preservation rate (PR), rejection rate (RR), and F_1 score [14], [15], defined as:

- $\text{PR} = \frac{\text{\# of preserved static voxels}}{\text{\# of total static voxels on the naively accumulated map}}$,
- $\text{RR} = 1 - \frac{\text{\# of remaining dynamic voxels}}{\text{\# of total dynamic voxels on the naively accumulated map}}$,
- $\text{F}_1 = 2\text{PR} \cdot \text{RR}/(\text{PR} + \text{RR})$.

We assess the performance of the static map building approaches with synced scans, i.e. $\mathcal{P}_t$, as inputs.

For simplicity, we refer to each sensor type used in our dataset as `L`, `A`, `O`, and `V`, respectively, as described in Section III.

TABLE II. Mean IoU of MOS approaches trained on the SemanticKITTI dataset to evaluate generalization capabilities in terms of both environmental changes and LiDAR sensor variations (`L`: Livox Avia, `A`: Aeva Aeries II, `O`: Ouster OS2-128, and `V`: Velodyne VLP-16).

Method	Solid-state		Omnidirectional		Total
	`L`	`A`	`O`	`V`	
4DMOS, online [5]	41.96	62.83	65.06	4.84	43.67
4DMOS, delayed [5]	**48.44**	68.60	71.53	5.46	48.51
MapMOS, Scan [6]	37.60	68.28	81.24	6.86	48.50
MapMOS, Volume [6]	45.17	**69.32**	**81.53**	**9.74**	**51.44**

B. Moving Object Segmentation Performance Against Environmental Changes and LiDAR Sensor Variations

First, we evaluate the generalization capabilities of MOS approaches in untrained environments and the different types of LiDAR sensors. To this end, we mainly employ 4DMOS [5] and MapMOS [6], which are state-of-the-art volumetric MOS approaches that do not employ range image projection and thus can be directly applied in other LiDAR setups.

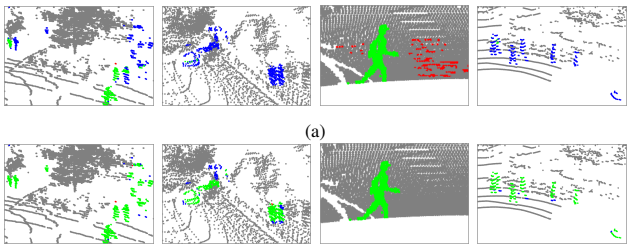
We can analyze the results of this experiment in three aspects. First, we demonstrate the robustness of these volumetric MOS approaches against environmental changes. This is evidenced by the relatively little performance degradation with `O`, which is the sensor most similar to the 64-channel sensor used to acquire SemanticKITTI. Second, in contrast, we observed substantial performance degradation in solid-state LiDAR cases. Third, when using sparser point clouds as inputs, the performance of MOS approaches was more significantly degraded (see columns `L` and `V` in Table II). This is because these MOS approaches heavily depend on the pose estimation modules to use temporal information from LiDAR sequences. Consequently, once the estimated poses are imprecise owing to the sparse point clouds, the performance becomes worse.

C. Moving Object Segmentation Performance Across Heterogeneous LiDAR Sensors

The next experiment specifically focuses on evaluating the performance changes caused by domain shifts across different LiDAR sensor types within the same environments. The MOS models showed substantial performance improvements across all sensors after training with our dataset, as presented in Table III and Fig. 9.

Interestingly, unlike 4DMOS, whose performance for each test sensor type increased as more diverse train data were provided, the performance of MapMOS showed inconsistency. MapMOS has better generalization capabilities [6] by taking a local map to reduce the geometrical differences between each scan from different sensors by accumulating scans over time and a scan as inputs. For this reason, even though MapMOS was trained by using `L`+`A`, it showed promising performance in `O`. This is because the local maps generated by `L` and `A`, and those from `O` are more similar when compared with the raw scans themselves.

Unfortunately, scans from `V` are too sparse to precisely estimate the relative poses, making local maps sparse and



(a)

(b)

Fig. 9. (a)-(b) Qualitative comparison of MapMOS [6] across all sensors before and after training with our dataset. Green, red, and blue points indicate true positives, false positives, and false negatives, respectively. The fewer red and blue points there are, the better (best viewed in color).

TABLE III. Mean IoU of MOS approaches when trained solely on data from specific LiDAR sensors. The bold and the gray highlight indicate the best performance among all trials and that for each method within each test set, respectively (L: Livox Avia, A: Aeva Aeries II, O: Ouster OS2-128, and V: Velodyne VLP-16).

Method	Train data	Solid-state			Omnidirectional			Total
		L	A	Avg	O	V	Avg	
4DMOS [5]	L+A	63.44	80.09	71.77	73.29	51.28	62.29	67.01
	O+V	44.78	66.00	55.39	77.02	54.44	65.73	60.56
	All	68.13	**81.85**	74.99	78.71	**57.57**	**68.14**	**71.57**
MapMOS [6]	L+A	**72.86**	81.38	**77.12**	81.46	38.13	59.80	68.46
	O+V	62.61	77.21	69.91	79.38	52.85	66.12	68.01
	All	72.55	80.18	76.37	**83.53**	43.74	63.64	70.00

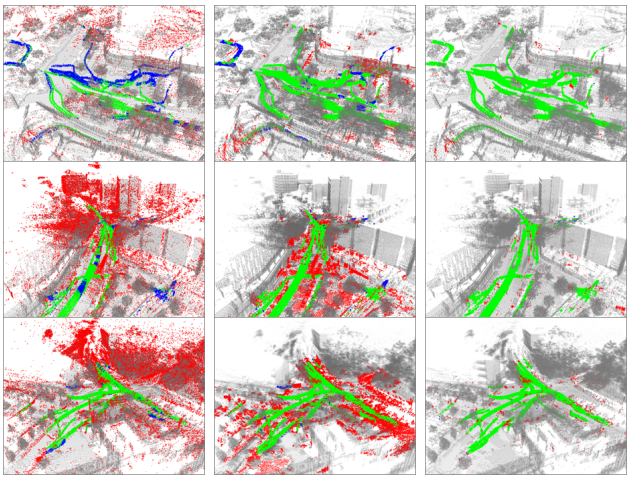
more distorted compared with other sensors. As a result, MapMOS showed lower IoUs with V in Table III and relatively poor performance. Nevertheless, MapMOS was on par with 4DMOS regarding total mean IoU and showed the highest performance in O when all train data were employed.

Therefore, these two experiments imply that there is still room for improvement in making existing MOS methods operate in a sensor-agnostic manner.

D. Automatic Labeling Performance

Finally, we demonstrate the superiority of our automatic labeling framework. Because only a part of the Auto-MOS [1] is open-sourced, we separately evaluate the performance of (a) static map building approaches for initial MOS labeling and (b) tracking-based filtering.

First, we demonstrate that ERASOR2 shows a substantially higher F_1 score compared with Removert, a range image-based approach, and ERASOR, an initial MOS labeling module in the Auto-MOS, as shown in Table IV. In particular, ERASOR showed lower PR and RR than ERASOR2. This is because ERASOR directly subtracts the estimated dynamic points from the map cloud without considering instance information, incorrectly estimating static points as dynamic while leaving some dynamic points on the map, as described in Fig. 10(b). In contrast, by leveraging instance information, ERASOR2 precisely rejected traces of moving objects in the map cloud while preserving most static points, as presented in Fig. 10(c). This could be interpreted as ERASOR2 consistently labeling the dynamic points within each scan.



(a) Removert [13] (b) ERASOR [14] (c) ERASOR2 [15]

Fig. 10. (a)-(c) Qualitative comparison of static map building results produced by state-of-the-art methods on our dataset using synced scans. Green, red, and blue points indicate true positives, false positives, and false negatives, respectively. The fewer red and blue points there are, the better (best viewed in color).

TABLE IV. Comparison of static map building approaches for the most crowded frame sequences in our dataset (PR: Preservation Rate, RR: Rejection Rate).

Frame range	Method	PR [%]	RR [%]	F_1 score
2,250-2,500	Removert [13]	84.615	54.836	0.665
	ERASOR [14]	95.448	87.556	0.913
	ERASOR2 [15]	**99.374**	**98.497**	**0.989**
8,600-8,800	Removert [13]	81.102	77.173	0.791
	ERASOR [14]	92.667	91.179	0.919
	ERASOR2 [15]	**99.455**	**98.193**	**0.988**
11,050-11,300	Removert [13]	78.419	85.497	0.818
	ERASOR [14]	93.146	97.642	0.953
	ERASOR2 [15]	**99.372**	**99.959**	**0.997**

TABLE V. Mean IoU before and after the application of tracking-based filtering approaches.

Method	IoU_{MOS}
ERASOR2 [15]	21.4
ERASOR2 + Tracking-based filtering in Auto-MOS [1]	25.2
ERASOR2 + Our tracking-based filtering	**34.8**

Second, as shown in Table V, the tracking-based filtering in Auto-MOS showed a substantial performance increase, which indicates that it significantly reduces the number of false positive points. However, as described in Section III.*C*, it cannot reduce the number of false negatives. In contrast, by introducing augmented bounding boxes, our approach could suppress the impact of false negative points. By doing so, our proposed filtering showed higher IoU_{MOS}.

Therefore, we conclude that the combination of ERASOR2 and our proposed tracking-based filtering is a suitable automatic labeling framework to help human labelers reduce the time needed for manual labeling.

V. Conclusion

In this paper, we have presented a novel moving object segmentation dataset for heterogeneous LiDAR sensors and

an instance-aware automatic labeling framework. Furthermore, we have proposed a novel instance-aware automatic labeling framework to reduce the time cost and effort of a human labeler when annotating labels in large-scale scenes. Finally, we demonstrate the necessity of a heterogeneous LiDAR moving object segmentation dataset by suggesting new research directions towards sensor-agnostic segmentation and enable better evaluations in this field of research.

In future work, we will further study domain generalization of MOS approaches in terms of different environments and different settings between existing datasets and our HeLiMOS.

ACKNOWLEDGEMENTS

Above all things, we thank Prof. Ayoung Kim's group, particularly Minwoo Jung, for making their HeLiPR dataset [18] and development tools available.

REFERENCES

[1] X. Chen, B. Mersch, L. Nunes, R. Marcuzzi, I. Vizzo, J. Behley, and C. Stachniss, "Automatic labeling to generate training data for online LiDAR-based moving object segmentation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6107–6114, 2022.

[2] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss, "Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data," *IEEE Robot. Automat. Lett.*, vol. 6, pp. 6529–6536, 2021.

[3] J. Sun, Y. Dai, X. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen, "Efficient spatial-temporal information fusion for LiDAR-based 3D moving object segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2022, pp. 11 456–11 463.

[4] J. Kim, J. Woo, and S. Im, "RVMOS: Range-view moving object segmentation leveraged by semantic and motion features," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 8044–8051, 2022.

[5] 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. Automat. Lett.*, vol. 7, no. 3, pp. 7503–7510, 2022.

[6] B. Mersch, T. Guadagnino, X. Chen, I. Vizzo, J. Behley, and C. Stachniss, "Building volumetric beliefs for dynamic environments exploiting map-based moving object segmentation," *IEEE Robot. Automat. Lett.*, pp. 5180–5187, 2023.

[7] H. Wu, Y. Li, W. Xu, F. Kong, and F. Zhang, "Moving event detection from LiDAR point streams," *Nature Comm.*, vol. 15, no. 1, pp. 345–358, 2024.

[8] J. Cheng, K. Zeng, Z. Huang, X. Tang, J. Wu, C. Zhang, X. Chen, and R. Fan, "MF-MOS: A Motion-focused model for moving object segmentation," *arXiv preprint arXiv:2401.17023*, 2024.

[9] S. Gu, S. Yao, J. Yang, C. Xu, and H. Kong, "LiDAR-SGMOS: Semantics-guided moving object segmentation with 3D LiDAR," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2023, pp. 70–75.

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

[11] C. Stachniss and W. Burgard, "Mobile robot mapping and localization in non-static environments," in *Proc. National Conf. Artif. Intell.*, 2005, pp. 1324–1329.

[12] C. Stachniss, *Robotic Mapping and Exploration*. Springer, 2009, vol. 55, Accessed: Feb. 28th, 2024. [Online]. doi: https://doi.org/10.1007/978-3-642-01097-2.

[13] G. Kim and A. Kim, "Remove, then revert: Static point cloud map construction using multiresolution range images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2020, pp. 10 758–10 765.

[14] 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. Automat. Lett.*, vol. 6, no. 2, pp. 2272–2279, 2021.

[15] H. Lim, L. Nunes, B. Mersch, X. Chen, J. Behley, H. Myung, and C. Stachniss, "ERASOR2: Instance-aware robust 3D mapping of the static world in dynamic scenes," in *Robot. Sci. Syst.*, 2023, doi: https://doi.org/10.15607/rss.2023.xix.067.

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

[17] L. Qingqing, Y. Xianjia, J. P. Queralta, and T. Westerlund, "Multi-modal LiDAR dataset for benchmarking general-purpose localization and mapping algorithms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2022, pp. 3837–3844.

[18] M. Jung, W. Yang, D. Lee, H. Gil, G. Kim, and A. Kim, "HeLiPR: Heterogeneous LiDAR dataset for inter-LiDAR place recognition under spatial and temporal variations," *arXiv preprint arXiv:2309.14590*, 2023.

[19] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9297–9307.

[20] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao, "SemanticPOSS: A point cloud dataset with large quantity of dynamic instances," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 687–693.

[21] S. Jang, M. OH, B. YU, I. Nahrendra, S. Lee, H. Lim, and H. Myung, "TOSS: Real-time tracking and moving object segmentation for static scene mapping," in *Proc. Int. Conf. Robot Intell. Tech. Appl.*, 2023, Accepted. To appear.

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

[23] H. Yin, X. Xu, S. Lu, X. Chen, R. Xiong, S. Shen, C. Stachniss, and Y. Wang, "A survey on global LiDAR localization: Challenges, advances and open problems," *Int. J. Comput. Vis.*, 2024, Accepted. To appear.

[24] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and LiDAR dataset," *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1023–1035, 2016.

[25] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 642–657, 2019.

[26] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "MulRan: Multimodal range dataset for urban place recognition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6246–6253.

[27] A. Carballo, J. Lambert, A. Monrroy, D. Wong, P. Narksri, Y. Kitsukawa, E. Takeuchi, S. Kato, and K. Takeda, "LIBRE: The multiple 3D LiDAR dataset," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 1094–1101.

[28] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford Robotcar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.

[29] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11 621–11 631.

[30] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, *et al.*, "Large scale interactive motion forecasting for autonomous driving: The Waymo open motion dataset," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9710–9719.

[31] P. Xiao, Z. Shao, S. Hao, Z. Zhang, X. Chai, J. Jiao, Z. Li, J. Wu, K. Sun, K. Jiang, *et al.*, "PandaSet: Advanced sensor suite dataset for autonomous driving," in *Proc. IEEE Int. Intell. Transport. Syst. Conf.*, 2021, pp. 3095–3101.

[32] K. Chen, R. Ge, H. Qiu, R. Ai-Rfou, C. R. Qi, X. Zhou, Z. Yang, S. Ettinger, P. Sun, Z. Leng, *et al.*, "WOMD-LiDAR: Raw sensor dataset benchmark for motion forecasting," *arXiv preprint arXiv:2304.03834*, 2023.

[33] A. H. Gebrehiwot, P. Vacek, D. Hurych, K. Zimmermann, P. Pérez, and T. Svoboda, "Teachers in concordance for pseudo-labeling of 3D sequential data," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 536–543, 2022.

[34] L. McInnes, J. Healy, and S. Astels, "HDBSCAN: Hierarchical density based clustering," *J. Open Source Softw.*, vol. 2, no. 11, pp. 205–206, 2017.