IEEE Open Journal of Signal Processing

Received 9 September 2024; revised 18 December 2024; accepted 30 December 2024. Date of publication 20 January 2025; date of current version 12 February 2025. The review of this article was arranged by Associate Editor N. Rodrigues.

Digital Object Identifier 10.1109/OJSP.2025.3532199

Efficient Moving Object Segmentation in LiDAR Point Clouds Using Minimal Number of Sweeps

ZOLTAN ROZSA [1,2] (Member, IEEE), AKOS MADARAS[1], AND TAMAS SZIRANYI[1,2] (Senior Member, IEEE)

[1]Machine Perception Research Laboratory of HUN-REN Institute for Computer Science and Control (HUN-REN SZTAKI), H-1111 Budapest, Hungary
[2]Department of Material Handling and Logistics Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics, H-1111 Budapest, Hungary

CORRESPONDING AUTHOR: ZOLTAN ROZSA (email: rozsa.zoltan@sztaki.hun-ren.hu).

This work was supported in part by János Bolyai Research Scholarship of the Hungarian Academy of Sciences, in part by the European Union Within the Framework of the National Laboratory for Autonomous Systems under Grant RRF-2.3.1-21-2022-00002, and in part by the Ministry of Culture and Innovation of Hungary from the National Research, under the development and innovation fund under Grant STARTING 149552, Grant K 139485, and Grant 2024-1.2.6-EUREKA-2024-00002. *(Zoltan Rozsa and Akos Madaras contributed equally to this work.)*

The implementation with examples and pre-trained networks is available: https://github.com/madak88/2DPASS-MOS.

ABSTRACT LiDAR point clouds are a rich source of information for autonomous vehicles and ADAS systems. However, they can be challenging to segment for moving objects as - among other things - finding correspondences between sparse point clouds of consecutive frames is difficult. Traditional methods rely on a (global or local) map of the environment, which can be demanding to acquire and maintain in real-world conditions and the presence of the moving objects themselves. This paper proposes a novel approach using as minimal sweeps as possible to decrease the computational burden and achieve mapless moving object segmentation (MOS) in LiDAR point clouds. Our approach is based on a multimodal learning model with single-modal inference. The model is trained on a dataset of LiDAR point clouds and related camera images. The model learns to associate features from the two modalities, allowing it to predict dynamic objects even in the absence of a map and the camera modality. We propose semantic information usage for multi-frame instance segmentation in order to enhance performance measures. We evaluate our approach to the SemanticKITTI and Apollo real-world autonomous driving datasets. Our results show that our approach can achieve state-of-the-art performance on moving object segmentation and utilize only a few (even one) LiDAR frames.

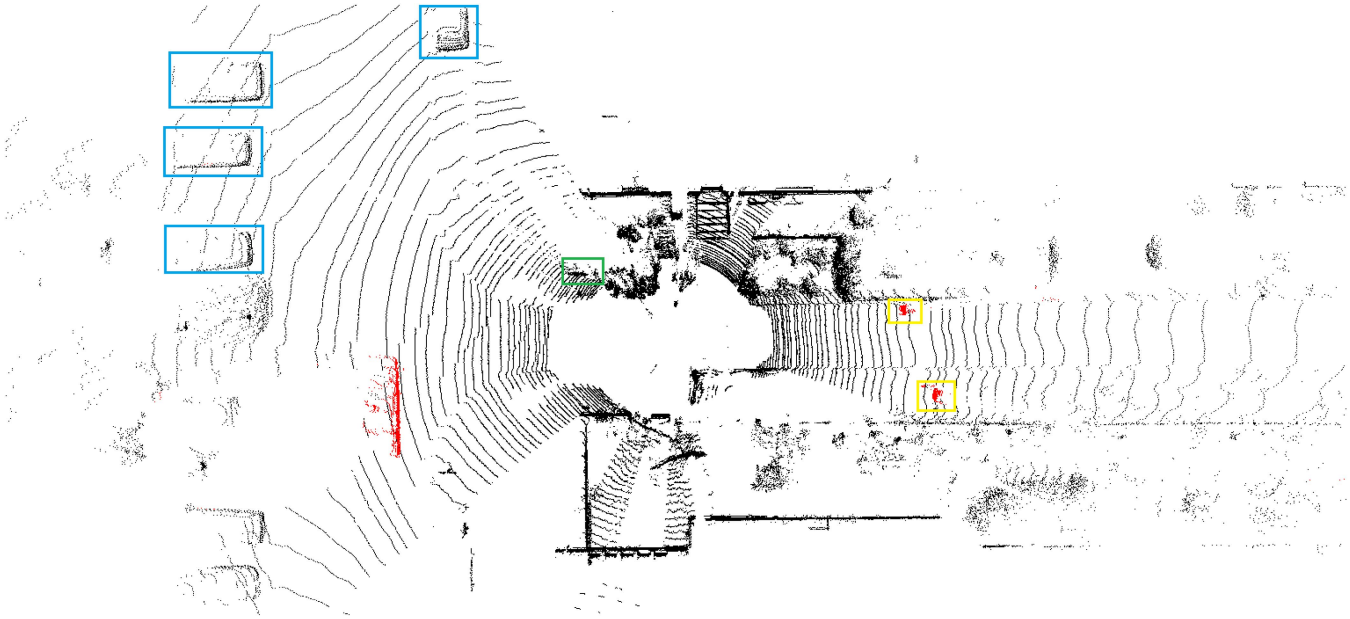
INDEX TERMS LiDAR, point clouds, moving object segmentation, knowledge transfer, autonomous driving.

I. INTRODUCTION

Moving object segmentation is critical for many applications, including autonomous driving, robotics, and surveillance [1], [2], [3]. In autonomous driving, it is usually interpreted as the point-wise detection of dynamic objects, mainly pedestrians, cyclists, cars, and other vehicles. This information is used to avoid collisions and maintain safe distances.

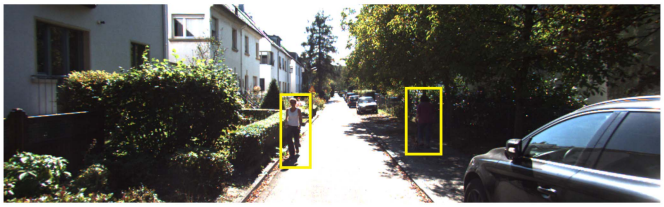
One of the drawbacks of previous approaches is the computational inefficiency resulting from the simultaneous processing of several point clouds (about 0.5–1 million points). Another disadvantage of them is that they rely on predetermined classes. Ignoring previously unseen classes of moving objects (e.g., animals) could cause a serious threat. Our proposal (later referred to as 2DPASS-MOS) is efficient in terms of required sweeps (or points, thus computation), and it can benefit from the semantic information without restricting categories by applying instance segmentation only as a refinement step.

MOS could be realized with different sensors like cameras [4], LiDARs [5], or with sensor fusion [6]. Each modality



(a) Estimated moving object points (colored with red) in LiDAR point cloud

(b) Camera's (earlier) frame, (only for) illustrating the street view from the left-hand side of the LiDAR top-view



(c) Camera's (corresponding) frame, (only for) illustrating the street view from the right-hand side of the LiDAR top-view

FIGURE 1. Example of our single scan-based moving object prediction. (Reference objects are framed with blue - parking cars, green - parked bicycle, and yellow - pedestrians.).

performance should be maximized to optimize cooperative performance and redundancy. Our work focuses on LiDAR modality in application, but its performance optimization is done by utilizing another sensor (camera) during the training. The advantage of solely relying on LiDAR in the prediction phase is that we can leverage its inherent strengths such as independence from lighting conditions.

On the one hand, MOS is an essential topic as it can significantly improve safety by drawing attention to hazardous objects which should be carefully considered in trajectory planning. Also, situational awareness can be enhanced by better understanding the environment around us.

On the other hand, it is highly relevant as it can aid other components of intelligent transportation systems. Simultaneous Localization and Mapping (SLAM) generally works so that filtering moving objects from the scene can seriously increase their performance [7]. This can be explained by the fact that the registration problem of consecutive frames (we will refer to this as local map building) can be more robustly solved without outliers [8].

Other LiDAR MOS methods require localization and local (using about 10 consecutive sweeps [9]) or global maps [10] to work with high performance. That is why using them as a preprocessing step to a SLAM algorithm would be highly inefficient.

Besides, utilizing very little information (a few frames) and providing decision information as quickly as possible [11], [12] is an important ambition for computation efficiency and real-time operation. We consider an algorithm to be capable of real-time operations at a frequency that is at least as high as the sensor's acquisition rate, typically ranging from 5 to 20 Hz in rotating LiDARs. Our algorithm meets the KITTI dataset's 10 Hz frame rate.

Our experiments indicate that only a few LiDAR frames (two or even one, without any map building or even poses) can be eligible to moving object segmentation. The term 'eligible' in this context refers to the suitability of a specific number of LiDAR frames for achieving satisfactory performance.

One sweep-based MOS, illustrated in Fig. 1, can be learned by interpreting the environment like human perception. The

model can discern that moving pedestrians typically exhibit different limb positions compared to stationary ones, as seen in Fig. 1(a) and (c). In addition, bikes occupied by a rider (Fig. 1(a) and (b)) are often associated with movement. In contrast, unoccupied bikes are more likely to be stationary. Furthermore, the model can distinguish between moving and static vehicles based on their location: vehicles on roads are generally in motion, while those parked in designated areas are static (Fig. 1(a) and (b)). Also, distortion of both the static and the dynamic part of the point cloud could correspond to some (ego and relative to ego) velocity information.

A. CONTRIBUTIONS

The paper contributes to the following:

- We demonstrated that only a few frames (even one) are eligible for efficient moving object segmentation in LiDAR point clouds.
- This is the first work successfully fusing camera and LiDAR data in training to enhance the LiDAR-only inference for LiDAR MOS.
- We improve the state-of-the-art by adapting a multi-modal learning scheme and extending it by multi-frame instance segmentation.
- An efficient MOS pipeline is proposed which is robust against decreasing the number of input frames. By utilizing fewer frames than others, delay and inference time can be reduced.
- To foster further research, we provide source code, pre-trained models and comprehensive sensitivity analysis highlighting both strengths and weaknesses.
- State-of-the-art performance is reached both in SemanticKITTI and Apollo on the most referenced LiDAR-MOS datasets.

B. OUTLINE OF THE PAPER

The paper is organized as follows: Section II surveys the literature about the related topics. Section III describes the proposed method and the concept in detail. Section IV shows our test results and evaluates them, while Section V elaborates further discussion. Finally, Section VI draws some conclusions and anticipates future work.

II. RELATED WORKS

LiDAR-based moving object segmentation is a relatively new challenge in the field of point cloud processing. The reason for that point-wise annotation of LiDAR data is exhausting work; one of the first databases that provided this kind of data (and still one of the most commonly used ones) is SemanticKITTI [13], [14] which is based on the even more popular autonomous driving dataset, the KITTI vision benchmark suite [15]. The second most frequently used dataset in the LiDAR MOS domain is based on the Apollo dataset [16], which is generally used to demonstrate the generalization capabilities of the different algorithms. We also use these datasets for benchmarking. The research related to LiDAR MOS can be sorted into two categories: offline and online methods.

A. OFFLINE MOS METHODS

Offline moving object segmentation methods, which require a larger set of precisely registered point clouds (global maps), can serve mainly two purposes. They either used for filtering static maps [17], [18], [19], [20], [21] or generating labels for online methods [10], [22]. There are different approaches among these frameworks, but most of them build their solution to occupancy grids with enhanced ground segmentation and line-of-sight-based calculations. As their category name states, they cannot be utilized for decision-making in autonomous driving.

B. ONLINE MOS METHODS

According to the leaderboard of the SemanticKITTI dataset, the most successful published LiDAR moving object segmentation methods are [9], [23], [24], [25], [26] and [27]. Ref. [9] was the first to apply residual images of range images with different semantic segmentation networks to solve the problem of moving object segmentation. This approach proved successful and was later adopted by others, e.g., by [24]. Range image representation is also used by [28] and [25]. Ref. [28] applied vision transformers and leveraged pre-trained models on RGB images and achieved comparable results to methods that use CNNs on similar data. Ref. [25] combined a semantic and a motion network to reach their best performance. They applied semantics as a prior, restricting recognizable moving object classes, which is highly disadvantageous. Ref. [23] - instead of the previously used range image representations - turned the LiDAR scans into voxelized sparse 4D point clouds and applied 4D convolutions in their model to estimate moving objects. Ref. [26] is similar to our work in the sense that they are utilizing instance information. However, our estimation of object instances works differently (explained in detail in Section III-D), resulting in higher accuracy in the MOS task. Currently, the best-performing model among the published MOS researches, on the SemanticKITTI dataset is proposed by [27], which utilizes residual maps to represent motion features. Constructing the representation proved highly inefficient in terms of computation resources. Our proposal outperforms it without having such a computational burden.

The above methods have the following in common: they all need at least 6 consecutive frames and/or poses to achieve high-performance MOS.

Compared to the above online methods, the method of this paper offers several advantages. Our proposal operates in real-time, fuses camera and LiDAR data during training to optimize LiDAR point cloud inference performance, and does not necessitate predetermined object categories. Additionally, our method requires only one or two frames for high-performance estimation, and our single-frame estimation

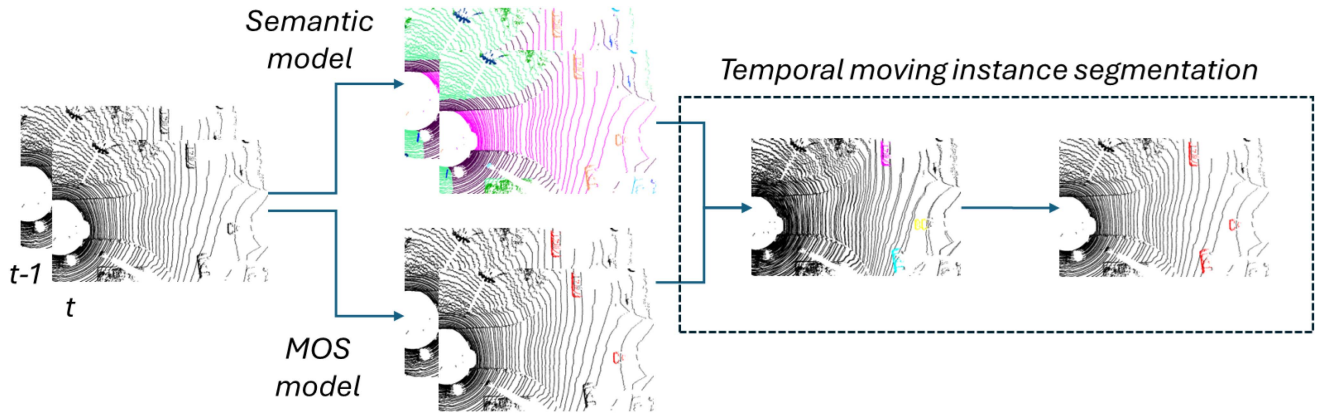


FIGURE 2. Proposed pipeline for LiDAR-based moving object segmentation from two frames.

does not rely on pose information. Most importantly, our approach achieves the highest scores on the KITTI and Apollo benchmarks.

III. THE PROPOSED METHOD

Here, the four steps of the proposal are described in detail.

1) Prerequisites
2) Camera and LiDAR fusion in the training process
3) Inferencing on LiDAR data
4) Enhance the estimates by semantics-based instance segmentation

These steps are described in the following subsections. The schematics of the pipeline is illustrated in Fig. 2.

A. PREREQUISITES

As a prerequisite before running the framework, the intrinsic calibration [29] of the camera should be executed together with the LiDAR-camera calibration [30]. The transformation matrix from the LiDAR coordinate system to the camera coordinate system will be indicated as $T_{L,C}$ and the intrinsic matrix as K in the following. In this way, the 3D point cloud can be projected to the image plane using the equation.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \cdot \mathbf{T_{L,C}} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

where [u v] are the coordinates in the image plane and [X Y Z] are the 3D point coordinates.

B. CROSS-MODAL KNOWLEDGE TRANSFER

Our framework adapts the cross-modal knowledge transfer with 2DPASS network architecture from [31]. The network was designed and applied earlier to semantic segmentation. A method utilizing this scheme is currently in first place in the latter task of the SemanticKITTI leaderboard. The essence of the knowledge distillation scheme across different modalities is the transfer of 2D knowledge through a multi-scale fusion-to-single manner; this takes care of the modal-specific knowledge. The multi-scale fusion-to-single knowledge distillation scheme first fuses features of both images and point clouds and then conducts unidirectional alignment between the point cloud fused and the fused features. The goal is to preserve modal-specific information and retain complete information from the fusion. Implementation details can be found in [31].

Base model: For our MOS problem, we defined three categories: static, dynamic, and 'do not care' classes. (The latter is necessary as in the SemanticKITTI dataset, there are 'unlabeled' and 'outlier' labeled points.) The total loss of the segmentation ζ_{all} is the sum of Lovasz (ζ_{iou}) [32] and Cross-entropy (ζ_{acc}):

$$\zeta_{all} = \zeta_{iou} + \zeta_{acc} \tag{2}$$

The former one enables the direct optimization of the mean Intersection over Union (or Jaccard index, see (5)), while the latter one is:

$$\zeta_{acc} = -\sum_{c=1}^{C} w_c \log(p_c) y_c \tag{3}$$

where p is predicted the probability of the given class, y is the target and w is the weight of the c^{th} class among C number of classes. The class weights were determined based on the inverse of the class frequency in the training dataset to ensure that the model pays sufficient attention to the underrepresented moving object class.

The training processes ran 64 epochs long, and an SGD (Stochastic Gradient Descent) optimizer was used. For the one-frame prediction case, batch size 8 was used, while in the two-frame prediction case, the batch size was 4. For the baseline model the knowledge distillation scheme parameters were the same as in [31] including segmentation loss and and Kullback–Leibler divergenece proportion (1:0.05).

The above-described learning method (adapting the knowledge transfer scheme from semantic segmentation problem to moving object segmentation) gives our baseline (indicated as Base in Table 4 of our Ablation study). This approach, without direct motion information, already outperforms some of the

current MOS methods. This can be explained by observations (listed in Section I) learned from images.

Two consecutive frames: Besides adapting it, we also extended the knowledge transfer scheme to a two consecutive scan-based model. We use only two consecutive frames as this is the lowest number of scans, including direct motion features.

From a computational point of view, one-frame-based prediction is the most efficient as it does not require registration of LiDAR frames and operates with the smallest number of points (influencing runtime).

However, applying only two consecutive frames is still considered efficient, as bundle adjustment is not needed, and the input points of the model are also low. In our implementation, two consecutive LiDAR point clouds are merged into one to access motion information in a common coordinate system. Concatenating the point clouds ($P = [P_{t-N+1,t-N+1} \quad P_{t,t-N+1}]$) happens after the coordinate transformation of the second point cloud to the coordinate system of the first one:

$$P_{t,t-N+1} = \mathbf{T_{t,t-N+1}} \cdot P_{t,t} \tag{4}$$

where the first index t of $P_{t,t-N+1}$ point cloud indicates the time moment of its acquisition. The second index $t - N + 1$ refers to the time moment of the measurement to which the coordinate system is transformed by the homogenous transformation matrix $\mathbf{T_{t,t-N+1}}$ calculated from the ego-movement of the vehicle. If the ego-motion is not measured directly, it can be calculated by registration algorithms like KISS-ICP [33] from the LiDAR data. We define N to get a general description, but a maximum of two consecutive frames are used for the training in our experiments, $N = 2$ (in the base model $N = 1$).

C. INFERENCE

The cross-modal knowledge transfer happens in the training phase. The inference requires only LiDAR point clouds, maximizing the efficiency of single-modal estimation. During the inference, no preprocessing was applied in the case of a single sweep, and only coordinate transformation (described in Section III-B) was in the case of a two-sweep model.

A voting scheme was proposed for a semantic segmentation problem in [34]. We also adopted this test-time augmentation (referenced as 'voting' or 'TTA' later) for our MOS problem. During the inference, the TTA rotates the input scene at different angles around the Z-axis and averages the prediction scores. It is included in the 'Base' model in Table 4. Prediction accuracy can be increased with this voting scheme. However, applying it is a trade-off between performance and runtime, which should be carefully considered. We suggest applying it to our unique single-frame inference; as with this configuration, one can still achieve better runtime performance than competitors. We also investigated the influence of the scheme in the case of two-frame inference; our experimental results related to this are introduced in Section V.

Algorithm 1: Object Level Decision.

Require: Merged point cloud P, target category y, moving threshold $moving\_threshold$
Ensure: P with updated point labels indicating moving objects

```
points ← SelectPoints(P, y)
clusters ← DetectObjectInstances(points, min_points = 200, max_distance = 0.5)
for all cluster ∈ clusters do
    moving_points ← FilterPoints(cluster, 'moving')
    moving_ratio ← Count(moving_points)/Count(cluster)
    if moving_ratio ≥ moving_threshold then
        CategorizeAllPointsAsMoving(cluster)
    end if
end for
```

D. APPLYING SEMANTIC INFORMATION

We propose to utilize semantic information as well. Previously to our work, LM-net [9] and MF-MOS [27] applied semantic information to the MOS problem. Their approach is different from ours, as they checked whether the predicted moving objects were movable or not, and consensus was necessary for a point to get a final moving label.

In our framework, semantic segmentation results are used to create instances, and only the addition of points to the moving category is possible based on them. This approach is consistent with our proposition that semantics categories must not be applied as prior knowledge, as it would exclude possible moving objects (e.g., animals). The semantic categories post factum considered are: pedestrian, car, cyclist, motorcyclist, bus, truck, rail and other vehicle.

InsMOS [26] also aimed to create instances to help the MOS problem. Their approach differs from our model as they do not use direct semantic information; instead, an instance detection head is part of their model.

2DPASS-MOS merges the advantages of these approaches and so outperforms them (as it is visible in Section IV-A).

In our proposal, the semantic labels are utilized in the following steps (pseudo code of Algorithm 1):

1) For a given point cloud (concatenated from multiple scans in a common coordinate system, $P = [P_{t-K+1,t-K+1} \quad P_{t-K+2,t-K+1} \quad \dots \quad P_{t,t-K+1}]$), the points of a given category are selected based on the previously predicted semantic labels.
   Note: K (the number of frames used in the instance segmentation process) can differ from N. The original 2DPASS [31] semantic segmentation network was applied to generate these labels in our tests.
2) On the remaining points, object instances are detected using dbscan [35] extension (details in Algorithm 1).
   Note: Our temporal (and conditional) dbscan extension is efficient (see Table 6) even in the case of increasing

TABLE 1. Performance Comparison of Methods Do Not Utilize Ego-Poses on SemanticKITTI Validation Dataset

Method	No. of input frames used by the model (N)	IoU
MInet [36]	1	36.9
Rangenet++ [37]	1	39.5
4DMOS [23]	5	39.9
LM-net [9]	1	51.9
SalsaNext [38]	1	53.4
Proposed ($TTA = 2$, $K = 1$)	1	**64.1**
Proposed^{+} ($TTA = 12$, $K = 1$)	1	**66.0**

TABLE 2. Performance Comparison of Methods Which Utilizes Ego-Poses on SemanticKITTI Dataset.

Method	No. of input frames used by the model (N)	IoU
LM-net [9]	2	56.0
LM-net + residuals [9]	2	59.9
4DMOS [23]	2	69.0
Proposed ($TTA = 1$, $K = 2$)	2	**71.8**
InsMOS [26]	5	60.8
LM-net + semantics [9]	9	67.1
RVMOS [25]	6	71.2
Motionseg3D [24]	8	71.4
4DMOS [23]	10	71.9
InsMOS*$^{+}$ [26]	10	73.2
Proposed^{+} ($TTA = 12$, $K = 2$)	2	74.9
MF-MOS*$^{+}$ [27]	8	76.1
Proposed^{+} ($TTA = 12$, $K = 10$)	2	**77.8**

* indicates that the given solution used the KITTI-road additional dataset for the training. + indicates that the given solution was not considered a real- time one, as about 5hz running speed has not been reached with the test configuration.

K. The reason for that is that the algorithm deals with a small number of points of the point cloud, selected in Step 1).

3) Each object instance of the given frame, generated with the previous steps, is examined, and a binary decision is made on the object level. If the object contains more than a given percent of points with a 'moving' label prediction, then all the object's points should be categorized as 'moving.' (The threshold percent was 40% in our experiments.)

To summarize, our temporal instance segmentation algorithm iterates over clusters of points belonging to a specific category (e.g., cars), through multiple frames (if not one sweep pipeline is used). For each cluster, it calculates the ratio of points labeled as 'moving' to the total number of points in the cluster. If this ratio exceeds a predefined threshold, all points within the cluster are categorized as 'moving'. This approach allows for robust object-level decisions, even in cases where individual point-level predictions may be noisy or uncertain.

IV. RESULTS

In this section, our test results are presented. In LiDAR-based moving object segmentation, the most commonly used benchmark is the SemanticKITTI [13], [14], so we also evaluated on this dataset. The LiDAR data of SemanticKITTI is point wisely annotated from the KITTI Odometry dataset with the same splits. We report the results for the validation dataset (unseen during the training). containing 4071 LiDAR scans (about 500 million points). The performance measure of the benchmark is applied:

$$IoU_{MOS} = \frac{TP}{TP + FP + FN} \tag{5}$$

where FN, TP, and FP are the numbers of False Negatives, True and False Positives of moving points, respectively.

A. QUANTITATIVE RESULTS

Table 1 reports results related to our most efficient, ego-poseless, one-frame pipeline for the SemanticKITTI validation dataset. Here, all the alternative methods, except one, utilize also only one frame for solving the MOS problem. Most of these are semantic segmentation frameworks re-trained for MOS task. The one exception is the 4DMOS [23] method, one of the highest-performing MOS solutions nowadays. This specific case of 4DMOS (reported by [23]) is included in our comparison as it also does not utilize ego-poses. In the training and inference, 5 consecutive frames are merged (using different local coordinate systems).

One can see that our proposed solution significantly outperforms all the other networks with single frame prediction and several with multiple frame ones from Table 2.

To evaluate our two-sweep solution, we provide Table 2 with the reported results of the competitors. In the first part of Table 2, we listed alternatives where only two consecutive frames are used as input of the model. The poses utilized in these tests were the database-provided ground-truth ones. 4DMOS [23] was the closest to our performance. However, one can see that in Table 1 4DMOS [23] performs very severely in poseless cases. Ultimately, our framework performed the best among these solutions.

We also listed (in the second part of Table 2) other state-of-the-art solutions from the LiDAR MOS field, which utilizes more scans (still not long sequences as offline methods). They have the disadvantage of computational inefficiency because processing several frames is required.

Our proposed method offers a state-of-the-art solution with real-time run compared to alternatives that do not use additional datasets for the training. Also, by applying test time augmentation, the best overall performance is achieved without using additional data.

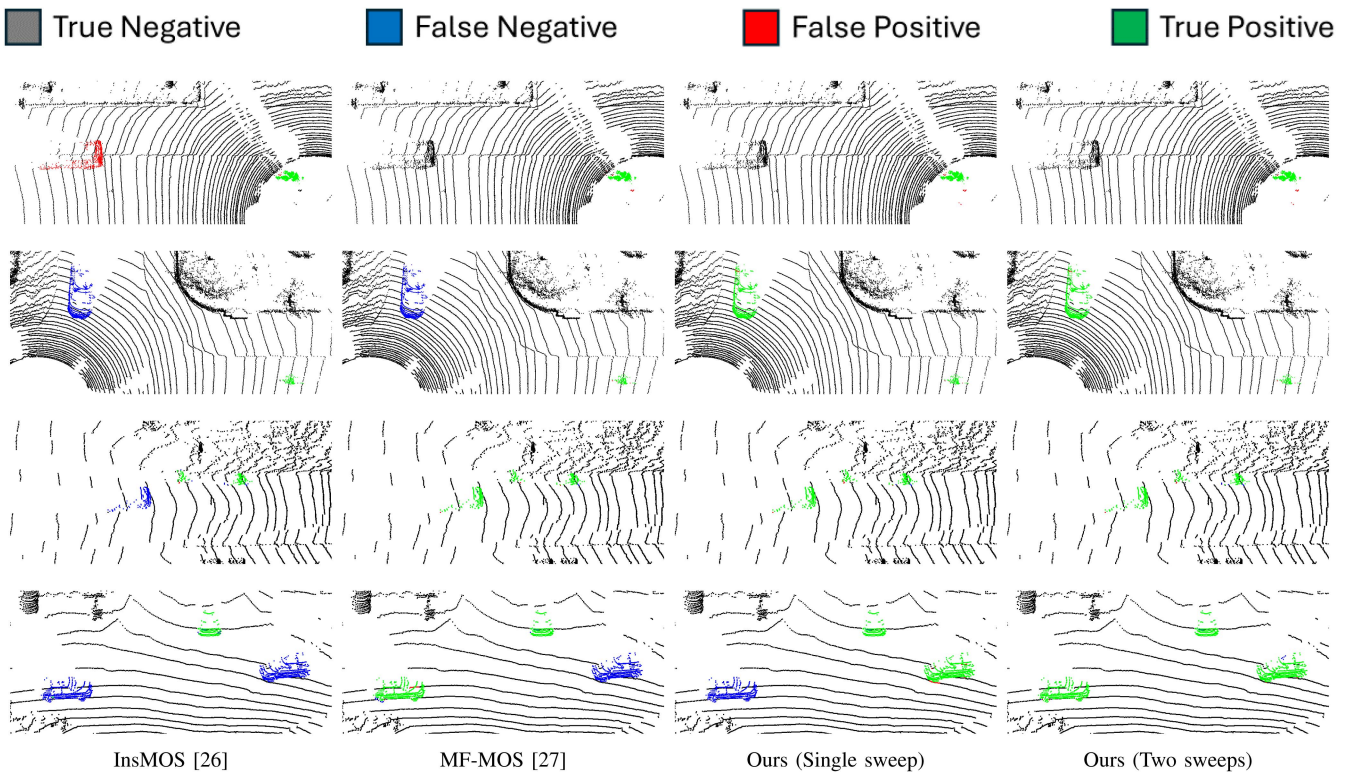


FIGURE 3. Qualitative examples of moving object predictions.

Besides, alternative methods are very sensitive to the decreased value of N. For example, InsMOS [26] (the currently published second best-performing solution on the SemanticKITTI test dataset in the MOS problem according to the leaderboard) provided data about the influence of N . The method achieves the best performance using 10 frames. However, if the frame number is decreased to 5in training, the authors of [26] found that the IoU values significantly drop to the level that is outmatched even by our single frame prediction. Our two frames-based, real-time estimation surpasses this model by a large margin (more than 10%). A comparison of how competitors perform in the case of different scan numbers is introduced in Section V, Fig. 4.

We have executed experiments to analyze the proposal's performance in the case of the most frequent semantic categories of moving objects. In the case of the parameter settings $TTA = 12$, $K = 2$, and $N = 2$, we have got the following IoU values for the SemanticKITTI validation set: Moving car: **76.1**, Moving person: **61.7** and Moving cyclists: **94.4**. One can see that the car (most frequent category) is about the same as the value for general moving objects. Cyclists' segmentation score is almost 100% . Person categories are the most troublesome; the reason for that is that even the general concept of someone moving or not is not obvious.

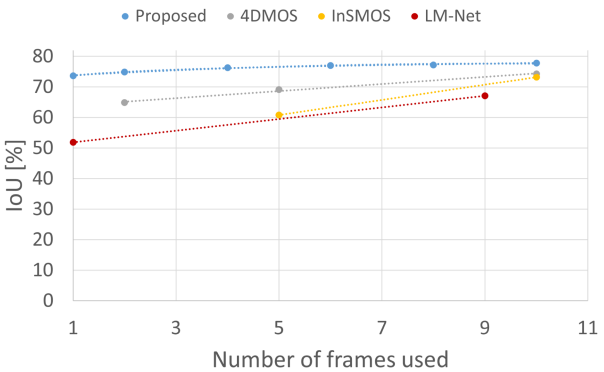


FIGURE 4. IoU as a function of different numbers of sweeps (K) in our instance segmentation process compared to IoU performance variation in case of applying different numbers of sweeps in competitor methods.

B. QUALITATIVE RESULTS

In the following, qualitative examples illustrate our tests evaluated in Tables 1 and 2. In Fig. 3, some typical example is visualized where the proposal's advantage can be compared to alternatives. All the figure rows show an example from the SemanticKITTI dataset with our results compared to the currently best performing other two methods. In the first row, it is visible that car parking on the road is falsely detected as moving one by [26], while both of our predictions are correct. The other three examples show the most frequently appearing error of other methods: moving car points are categorized as static. The reason behind that could be that other methods use many more sweeps for the predictions than us; that is why their model anticipates significant

TABLE 3. Comparison on Apollo Dataset

Method	No. of input frames used by the model (N)	IoU
LM-net [9]	9	16.9
LM-net (fine-tuned) [9]	9	65.9
MF-MOS [27]	8	49.9
MF-MOS (fine-tuned) [27]	8	70.7
4DMOS [23]	10	73.1
InsMOS [26]	10	78.0
Proposed ($TTA = 1$, $K = 2$)	2	**80.6**

movement, and slowly moving objects are assumed to be static.

V. DISCUSSION

In this section, a further study of our proposed framework is presented. First, we investigate the generalization capabilities of our 2DPASS-MOS; next, ablation studies of different components are presented. Finally, runtimes are discussed.

A. GENERALIZATION ANALYSIS

We demonstrate that our proposal generalizes well to different scenes by conducting experiments on the second most frequent MOS database, the Apollo [16] dataset. We follow the standard setup of [10] and use (the highly dynamic frames selected by them) sequence 2 and sequence 3 for testing. To achieve the reported performance, we only trained on the training set of SemanticKITTI and evaluated our framework on the Apollo dataset without modifying any settings or fine-tuning.

In Table 3, the accuracy of the proposed method with parameter set achieving the highest performance and still working in real-time is reported. One can see that some state-of-the-art alternative methods only provide acceptable performance when they are fine-tuned on the new dataset. Our system proved to be the most efficient without fine-tuning and using significantly fewer scans than others.

B. ABLATION STUDIES

In order to investigate the effect of different components of our system, we report an ablation study of the constructing elements of our approach in the SemanticKITTI validation dataset and one for the Apollo dataset.

As the baseline model already incorporates multi-modal learning, our focus here is on the incremental gains achieved by our proposed approach. A detailed ablation study on instance segmentation variants is presented in Tables 4 and 5. A dedicated analysis of TTA is presented in Fig. 5, highlighting its positive impact on the system's accuracy.

TABLE 4. Influence of Different System Components to the IoU in SemanticKITTI Dataset

Model		Semantics		
Base ($N = 1$)	Multi-frame ($N = 2$) extension	Instance segmentation ($K = 1$)	Multi-frame ($K = 2$) instance segmentation	IoU
✓	x	x	x	65.6
✓	x	✓	x	66.0
x	✓	x	x	73.2
x	✓	✓	x	73.6
x	✓	x	✓	74.9

TABLE 5. Influence of Different System Components to the IoU in Apollo Dataset

Model		Semantics		
Base ($N = 1$)	Multi-frame ($N = 2$) extension	Instance segmentation ($K = 1$)	Multi-frame ($K = 2$) instance segmentation	IoU
✓	x	x	x	49.1
✓	x	✓	x	52.0
x	✓	x	x	79.1
x	✓	✓	x	80.0
x	✓	x	✓	80.6

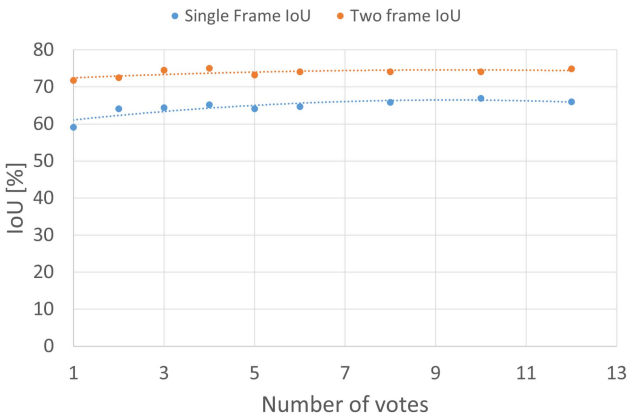


FIGURE 5. Effect of different number of votes (TTA) in the test time augmentation for our Single ($N = 1$, $K = 1$) and Two Frame ($N = 2$, $K = 2$) predictions.

In Table 4, the checkmark indicates that the given component (they are introduced in Section III) is used, and x means it is not.

In Tables 4 and 5, 'Base' means the 2DPASS model retrained for the MOS task in the SemanticKITTI training set and applying TTA = 12. One can see that all of our contributions raised the performance. Comparing Tables 4 and 5, it becomes apparent that the system generalizes learned

TABLE 6. Runtime Comparison in *ms* on SemanticKITTI

Method	LM-net [9]	InsMOS [26]	MF-MOS [27]	Proposed ($N = 1, K = 1$)	Proposed ($N = 2, K = 2$)
Pre-processing	72	0	514	0	0
Pose estimation	50	50	50	0	50
Inference	24	255	106	82	136
Post-processing	11	16	207	30	39
Total	157	321	877	**112**	225

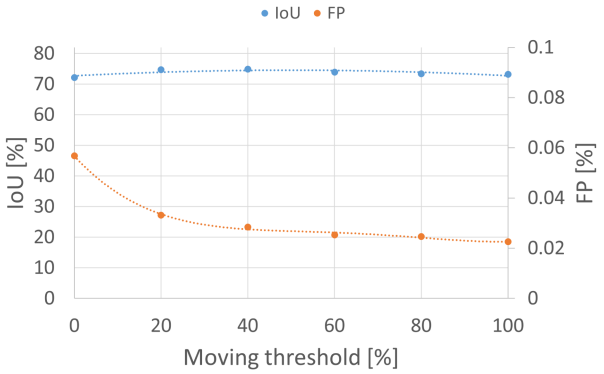


FIGURE 6. Impact of the moving threshold to our temporal moving instance segmentation ($TTA = 12$, $N = 2$, $K = 2$) in case of SemanticKITTI dataset.

temporal and velocity information more easily than semantic information. While the contribution of semantic information was less important in the SemanticKITTI (the dataset, the model was trained), it played the most significant role in the Apollo dataset (new environment).

In Tables 4 and 5 $K = 1$ and $K = 2$ cases are presented. To further investigate the number of frames used in the instance segmentation part of our pipeline (parameter K), we introduce Fig. 4.

C. SENSITIVITY ANALYSES

In this subsection, three analyses are provided, each of which highlights the effect of different hyperparameters of our system through the SemanticKITTI validation dataset. First, K, then the number of votes, and finally, $moving_t reshold$ impact is investigated. Fig. 4 (parameters $N = 2$ and $TTA = 12$ are fixed) depicts the relationship between the number of sweeps and the performance of our and other methods.

The performance of other methods significantly drops when the number of input frames is reduced. The performance of our 2DPASS-MOS remains high. The overall best performance we measured (by $K = 10$) is 77.8in the case of the SemanticKITTI validation set.

Note: There is a trade-off between accuracy (through increasing input frame number) and computational burden. Inspecting Fig. 4 and Table 6, it is observable that increasing the K value in our algorithm is computationally efficient. (In our tests, increasing K by 1 resulted in a maximum 9 ms run time increment in the post-processing.)

We also provide an analysis for the effect of different numbers of test time augmentation in Fig. 5. The runtime of TTA is directly proportional to the number of votes. That is why, for real-time at a configuration similar to our test one, $TTA = 2$ is proposed for $N = 1$ and $TTA = 1$ for $N = 2$.

In Fig. 6 the performance of the proposed method is investigated through different moving threshold values. Besides the IoU, False positives are also indicated normalized with the number of all the 'care' points of the dataset.

It is observable that the IoU optimum is around 40% (the value we used in our experiments), which results in just a bit higher FP than we would get without the temporal instance segmentation (moving threshold equal to 100%).

D. RUNNING TIME ANALYSIS

Here, we provide a runtime comparison to alternatives.

In Table 6, besides our single and two frame-based prediction cases, running time values are reported for the two currently best-performing competitors (InsMOS [26] and MF-MOS [27]) and also for LM-net [9], which provides the fastest implementation among the SemanticKITTI MOS leaderboard currently. Our test configuration was the following: AMD Ryzen 7 6800H with Radeon Graphics 3.20 GHz processor, 32 GB RAM, NVIDIA GeForce RTX 3070 GPU.

Preprocessing is required in the case of LM-net and MF-MOS corresponds to the residual image generation. MF-MOS needs significantly more of them minimally than LM-net, and this operation is computationally intensive; a real-time run is not possible for that method in the test hardware.

Pose estimation is necessary for all methods except in our one frame-based solution, but most of the competitors do not provide pose estimation implementation. Thus, the running time of KISS-ICP [33] in our test configuration is indicated for all these cells.

Postprocessing means refinement in the case of InsMOS and MF-MOS, using semantics in the case of LM-net and our proposal. LM-net assumed that the semantic segmentation could run parallel with the moving object segmentation; we used the same assumption for a fair comparison. Notably, the proposed method with $N = 1$ and $K = 1$ demonstrates significant efficiency gains in preprocessing and pose estimation, as these steps are not required. However, the inference and post-processing stages introduce additional overhead compared to LM-net. InsMOS and MF-MOS exhibit high inference and post-processing costs. MF-MOS requires the most runtime in all components except for inference, where InsMOS is the slowest.

Altogether, it is visible in Table 6 that our one frame solution is the fastest one among the presented solutions in our test configuration (introduced earlier), and only LM-net

is faster among the alternatives than our two-frame solution. However, its IoU performance (see Table 2) is not comparable to our proposal. LiDAR sensors typically operate at frequencies ranging from 5 to 20 Hz. Given this range, a processing time of approximately 10 Hz is well within the real-time requirements of most LiDAR-based applications. Furthermore, the KITTI dataset, which we employed for our evaluation, also features a 10 Hz LiDAR frame rate. It's worth noting that the reported processing times were obtained using standard, off-the-shelf hardware. We anticipate that with more specialized hardware, such as high-end GPUs, it would be feasible to achieve even faster processing speeds. In addition we have experienced that point cloud subsampling can lead to further significant speedups.

VI. CONCLUSION

A novel framework for moving object segmentation in LiDAR point clouds has been proposed. The proposed method performs state-of-the-art moving object segmentation and state-of-the-art generalization to new datasets.

We demonstrated that our proposal is less influenced than alternatives by the number of input sweeps; even one scan is sufficient for high-performance moving object segmentation. This is achieved by utilizing a multimodal learning model and training on LiDAR point clouds and camera images (and single modal inference only from LiDAR point clouds). Using fewer frames than other methods results in better computational efficiency. The potential application of our approach in autonomous vehicles and ADAS is significant.

In the future, we plan to investigate quantitatively the method's robustness against occlusion and also against generalization to unknown categories and a broader range of application domains. Besides, we would like to enhance LiDAR MOS by applying knowledge transfer from other autonomous sensors.

ACKNOWLEDGMENT

The authors are grateful for the possibility to use HUN-REN Cloud (see [39]; https://science-cloud.hu/) which helped us achieve the results published in this paper.

REFERENCES

[1] H. Lee, J. Yoon, Y. Jeong, and K. Yi, "Moving object detection and tracking based on interaction of static obstacle map and geometric model-free approach for urban autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3275–3284, Jun. 2021.

[2] C. Jiang, D. P. Paudel, D. Fofi, Y. Fougerolle, and C. Demonceaux, "Moving object detection by 3D flow field analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 1950–1963, Apr. 2021.

[3] L. Kovács, M. Kégl, and C. Benedek, "Real-time foreground segmentation for surveillance applications in NRCS LiDAR sequences," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 45–51, 2022.

[4] M.-N. Chapel and T. Bouwmans, "Moving objects detection with a moving camera: A comprehensive review," *Comput. Sci. Rev.*, vol. 38, 2020, Art. no. 100310.

[5] J. An, B. Choi, H. Kim, and E. Kim, "A new contour-based approach to moving object detection and tracking using a low-end 3 dimensional laser scanner," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7392–7405, Aug. 2019.

[6] Y. Cai, B. Li, J. Zhou, H. Zhang, and Y. Cao, "Removing moving objects without registration from 3D LiDAR data using range flow coupled with IMU measurements," *Remote Sens.*, vol. 15, no. 13, 2023, Art. no. 3390.

[7] Z. Rozsa, M. Golarits, and T. Sziranyi, "Localization of map changes by exploiting SLAM residuals," in *Advanced Concepts for Intelligent Vision Systems*, J. Blanc-Talon, P. Delmas, W. Philips, D. Popescu, and P. Scheunders, Eds. Cham, Switzerland: Springer, 2020, pp. 312–324.

[8] A. Efraim and J. M. Francos, "On minimizing the probability of large errors in robust point cloud registration," *IEEE Open J. Signal Process.*, vol. 5, pp. 39–47, 2024.

[9] X. Chen et al., "Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6529–6536, Oct. 2021.

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

[11] Z. Rozsa and T. Sziranyi, "Object detection from a few LIDAR scanning planes," *IEEE Trans. Intell. Veh.*, vol. 4, no. 4, pp. 548–560, Dec. 2019.

[12] Z. Rozsa, M. Golarits, and T. Sziranyi, "Immediate vehicle movement estimation and 3D reconstruction for mono cameras by utilizing epipolar geometry and direction prior," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 23548–23558, Dec. 2022.

[13] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9297–9307.

[14] J. Behley et al., "Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI dataset," *Int. J. Robot. Res.*, vol. 40, no. 8/9, pp. 959–967, 2021.

[15] 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.*, 2012, pp. 3354–3361.

[16] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, "L3-Net: Towards learning based LiDAR localization for autonomous driving," in *Proc. 2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6382–6391.

[17] J. Schauer and A. Nüchter, "The peopleremover—removing dynamic objects from 3-D point cloud data by traversing a Voxel occupancy grid," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1679–1686, Jul. 2018.

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

[19] M. Arora, L. Wiesmann, X. Chen, and C. Stachniss, "Mapping the static parts of dynamic scenes from 3D LiDAR point clouds exploiting ground segmentation," in *Proc. 2021 IEEE Eur. Conf. Mobile Robots*, 2021, pp. 1–6.

[20] 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, Apr. 2021.

[21] 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.*, vol. 8, no. 8, pp. 5180–5187, Aug. 2023.

[22] P. Pfreundschuh, H. F. C. Hendrikx, V. Reijgwart, R. Dubé, R. Y. Siegwart, and A. Cramariuc, "Dynamic object aware LiDAR SLAM based on automatic generation of training data," in *Proc. 2021 IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11641–11647.

[23] 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, Jul. 2022.

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

[25] 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, Jul. 2022.

[26] N. Wang, C. Shi, R. Guo, H. Lu, Z. Zheng, and X. Chen, "InsMOS: Instance-aware moving object segmentation in LiDAR data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 7598–7605.

[27] J. Cheng et al., "MF-MOS: A motion-focused model for moving object segmentation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 12499–12505.

[28] C. Ma, X. Shi, Y. Wang, S. Song, Z. Pan, and J. Hu, "MosViT: Towards vision transformers for moving object segmentation based on LiDAR point cloud," *Meas. Sci. Technol.*, vol. 35, no. 11, 2024, Art. no. 116302.

[29] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[30] Y. Hao, X. Jin, and D. Du, "Multi-dimensional geometric feature-based calibration method for LiDAR and camera fusion," in *Proc. 2024 IEEE Int. Conf. Acoust. Speech Signal Process.*, 2024, pp. 8406–8410.

[31] X. Yan et al., "2DPASS: 2D priors assisted semantic segmentation on LiDAR point clouds," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2022, pp. 677–695.

[32] M. Berman, A. Triki, and M. B. Blaschko, "The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proc. 2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Los Alamitos, CA, USA: Jun. 2018, pp. 4413–4421.

[33] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In defense of point-to-point ICP–simple, accurate, and robust registration if done the right way," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 1029–1036, Feb. 2023.

[34] X. Zhu et al., "Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9939–9948.

[35] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, AAAI Press, 1996, pp. 226–231.

[36] S. Li, X. Chen, Y. Liu, D. Dai, C. Stachniss, and J. Gall, "Multi-scale interaction for real-time LiDAR data segmentation on an embedded platform," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 738–745, Apr. 2022.

[37] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet : Fast and accurate LiDAR semantic segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4213–4220.

[38] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy, "SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds," in *Proc. Adv. Vis. Comput.: 15th Int. Symp., ISVC 2020, San Diego, CA, USA, Oct. 5–7, 2020, Proc., Part II*. Berlin, Heidelberg: Springer-Verlag, 2020, pp. 207–222.

[39] M. Heder et al., "The past, present and future of the ELKH cloud," *Informacios Tarsadalom*, vol. 22, no. 2, Aug. 2022, Art. no. 128.