IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

PGP-DOR: A Point-Grid-Point Scheme for Efficient Dynamic Object Removal

Shuo Wang[1], Zhenping Sun[1], Hanzhang Xue[2], Bokai Liu[1], Hao Fu[1] and Luo Yinfu[1]

***Abstract*—In the field of autonomous driving, constructing high-precision maps, typically represented as 3D point cloud maps or bird's-eye view (BEV) grid maps, is essential for both offline and online applications. However, the presence of dynamic objects within a scene can introduce artifacts and noise that significantly degrade the quality of these maps. To address this challenge, we propose a method in this paper that can accurately identify those dynamic objects in both online and offline settings. Our approach fully exploits the spatio-temporal attributes of BEV grid maps and utilizes a point-grid-point (PGP) scheme to identify moving objects at both the 3D point cloud level and the 2D BEV grid level. Experimental results from public datasets, as well as a self-collected dataset, demonstrate that our method consistently outperforms state-of-the-art approaches in dynamic object removal in both online and offline contexts. The code and the newly introduced dataset will be made publicly available at: https://github.com/MichealRW/PGP-DOR .**

***Index Terms*—Mapping, Object Detection, Segmentation and Categorization, Probabilistic Inference.**

I. Introduction

GENERATING point cloud maps and bird's-eye view (BEV) maps from LiDAR data is a fundamental task in autonomous driving. These maps not only enhance situational awareness but also provide a crucial foundation for subsequent path planning tasks.

However, moving objects, such as pedestrians or vehicles, introduce artifacts and noise into point cloud and BEV maps (as shown in Figure 1). Consequently, the efficient identification and removal of dynamic objects are essential for enhancing map quality and ensuring robust autonomous navigation.

Existing dynamic object removal methods can be broadly classified into two categories: learning-based and rule-based methods. Learning-based approaches [1]–[5], particularly those utilizing deep learning techniques, have demonstrated exceptional performance in dynamic object removal. However, these methods are heavily dependent on large-scale annotated datasets, the construction of which are often time-consuming endeavors. Furthermore, these learned models usually overfit to specific datasets, which restricts their generalizability to other datasets created with different LiDAR types.

Manuscript received: July 3, 2025; Revised August 30, 2025; Accepted October 19, 2025.

This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments.

[1]The authors are with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China wangshuo198@nudt.edu.cn *(Shuo Wang and Zhenping Sun are co-first authors.) (Corresponding author: Shuo Wang.)*

[2]Hanzhang Xue is with the Test Center, National University of Defense Technology, Xi'an 710106, Shanxi, China

Digital Object Identifier (DOI): see top of this page.

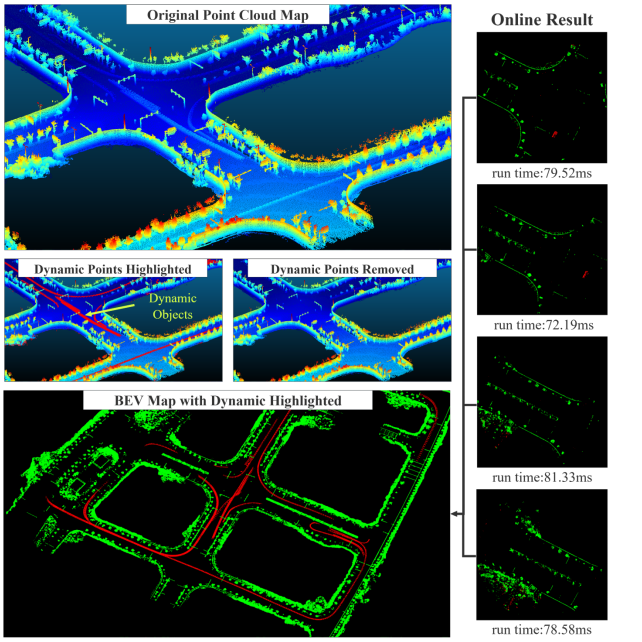


Fig. 1. This figure shows the map-level dynamic removal results (left figure) and the frame-level dynamic detection results (right figure) generated by the PGP-DOR method proposed in this paper. Dynamic objects are highlight in red.

Alternatively, rule-based methods [6]–[10] identify dynamic objects by leveraging geometric and temporal cues in LiDAR data. While these methods can effectively remove moving objects, they struggle in complex environments due to LiDAR sparsity and suffer from high computational costs, making them less suitable for real-time applications.

To address these limitations, we propose a novel and efficient dynamic object removal method in this paper. This method leverages both the efficient 2D BEV grid representation and the accurate 3D point cloud representation, combining the advantages of both through a point-grid-point scheme. In this scheme, dynamic attributes are first estimated at the point level. Subsequently, the point estimation results are fused at the BEV grid map level through a point projection step. At the BEV grid level, a Bayesian Generalized Kernel Inference (BGKI) step [11] is introduced for dense attribute estimation. The inferred states are then transformed back to the point level for further refinement.

The key contributions of this paper are as follows:

- We propose a bidirectional Point-Grid-Point update strategy for moving object detection. This method fully utilizes the efficiency of the 2D BEV grid map representation and the accuracy of the 3D point cloud

representation.
- At the 2D BEV level, we design a spatio-temporal inference method and also a BGKI module for dense dynamic attribute analysis. This approach effectively captures spatial dependencies while suppressing noise artifacts.
- The proposed method is evaluated in both the SemanticKITTI dataset and a dataset that we collected with a different type of LiDAR. State-of-the-art results were achieved in both datasets, confirming the applicability of our approach to different types of LiDARs.

The remainder of this paper are organized as follows: Section II reviews related work, Section III describes the proposed method in detail, Section IV and Section V presents the experimental evaluation of our method, Section VI concludes the paper.

II. Related Work

A. Map Representation Methods

An efficient map representation method is important for both real-time online perception tasks and offline map construction tasks in autonomous driving. Existing map representation approaches include 3D point cloud map, 2D grid map [12], mesh map [13]–[16], neural implicit map [17], etc.

3D point cloud map is a lossless representation of the environment. To process this map, approaches such as PointNet have been proposed, which can achieve high semantic segmentation accuracy [13]. However, these methods often suffer from high computational costs, hindering their real-time application.

Compared to raw 3D point cloud map, 3D voxel map or 2D grid map significantly reduce storage requirements. 3D voxel-based representation methods, such as Octree [17] provides an efficient spatial partitioning scheme. Zhu et al. [14] proposed i-Octree, a framework designed for large-scale scenarios. Hornung et al. [15] introduced OctoMap, a probabilistic occupancy grid mapping approach based on Octree. Duberg et al. [16] extended OctoMap by introducing UFOMap, which efficiently encodes three distinct states within the octree structure, allowing for fast insertion and rapid occupancy state queries.

In practical autonomous driving implementations, 2D BEV grid map have emerged as the dominant representation method due to its computational efficiency. While BEV grid maps can effectively model the surroundings of autonomous vehicles, they struggle to capture overhanging objects. Meng et al. [12] introduced a dual-layer BEV representation method, in which one layer records the maximum terrain elevation value, while the other stores the minimum height of overhanging objects. Inspired by this approach, we also develop a specific method in this paper to handle these overhanging objects.

B. Dynamic Object Removal Methods

Dynamic object removal methods can be broadly categorized into learning-based and rule-based approaches. Learning-based methods leverage labeled datasets to train deep neural networks. Mersch et al. [1] utilized sparse 4D convolutions for spatio-temporal feature extraction. Sun et al. [2] introduced a spatio-temporal fusion framework that uses range image representation to generate per-point dynamic probabilities. Wang et al. [3] proposed a dual-branch fusion strategy for motion and instance prediction. Zhang et al. [5] presented DeFlow, which estimates scene flow in LiDAR data to detect and remove dynamic objects. Despite their ability to handle complex motion patterns, these methods require extensive labeled datasets and usually overfit to a specific type of LiDAR.

On the contrary, rule-based methods typically explore the geometric attributes and handcrafted rules for dynamic object removal. Pagad et al. [6] used log-odds probability and a voxel traversal algorithm to estimate the dynamic state of a voxel. OctoMap-based approaches [15] employ ray-casting operator to mark occupied regions. Building on this work, DUFOMap [7] optimizes the octree insertion step to further enhance efficiency. Additionally, some approaches utilize range image representation to reduce computational loads, such as Removert [8], ERASOR [9], and ERASOR2 [10]. However, these methods are designed for offline usage, and cannot be used in online settings.

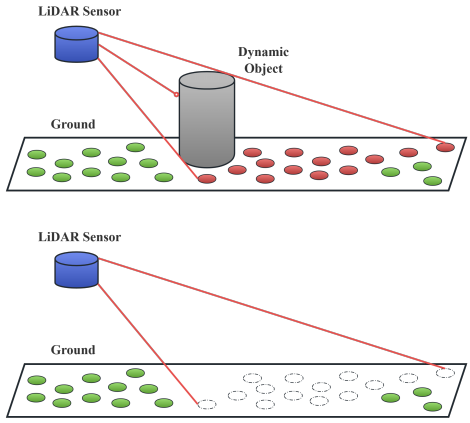


Fig. 2. The limitation of rule-based approaches is the tendency to remove ground points that are occluded by moving objects.

Another limitation of these rule-based methods is that they tend to generate lots of false positive detections in ground regions, as shown in Fig. 2. Therefore, some methods introduce ground segmentation as a pre-processing step to help eliminate these false positives. For example, Fu et al. [18] introduced a terrain modeling step to classifiy the LiDAR points into terrain points and non-terrain points, and dynamic objects are only found in these non-terrain points. Arora et al. [19] leverage an offline OctoMap framework to construct a reference ground model for enhanced dynamic filtering. Inspired by LeGO-LOAM [20], Yuan et al. [21] employ a ground segmentation strategy, augmented with spatio-temporal consistency checks for dynamic detection. However, until now, there is no perfect ground segmentation algorithm. Errors that occur during the ground segmentation step could not be remedied in the subsequent dynamic object detection steps. To overcome this limitation, we introduce a point-grid-point

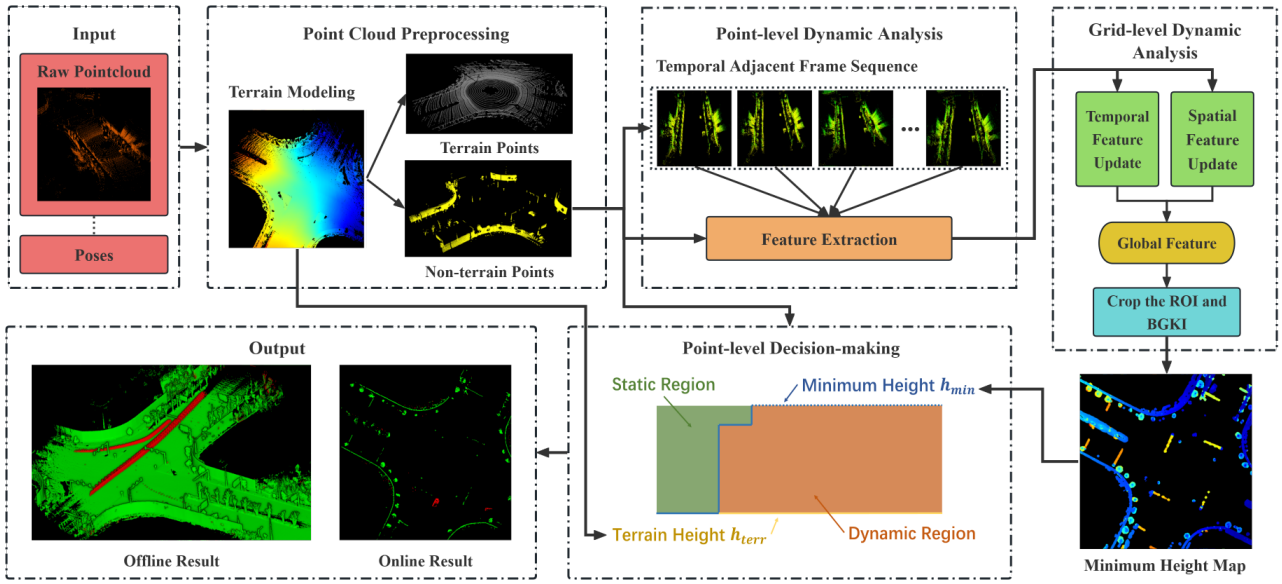


Fig. 3. The pipeline of the proposed PGP-DOR method. This method takes sequential point clouds and their poses as input, processes them through point cloud preprocessing, point-level dynamic analysis, grid-level dynamic analysis and point-level decision making. The outputs include both frame-level results for online applications and map-level results for offline applications.

update scheme in this paper that will re-evaluate all points' dynamic state after the BEV grid-level processing.

III. Method

In this work, we make a structural assumption tailored to autonomous driving scenarios: dynamic objects can only appear above the terrain surface and remain below the lowest occupied height of surrounding obstacles. Based on this assumption, we first prioritize terrain modeling to remove the terrain and all points beneath it, thereby isolating the above-ground point cloud. The remaining points are then subjected to dynamic analysis to distinguish moving objects from static ones. As illustrated in Fig. 3, the proposed framework is organized into four modules: point cloud preprocessing, point-level dynamic analysis, grid-level dynamic analysis, and point-level decision-making.

A. Point Cloud Preprocessing

Following the structural assumption that dynamic objects appear only above the terrain and below the lowest occupied height of surrounding obstacles, we first construct a BEV terrain height model, denoted as h_{terr}, to separate terrain points from above-ground points (non-terrain points). This process corresponds to the Terrain Modeling module illustrated in Fig. 3.

In the offline setting, a seed-growing strategy [18] is employed, leveraging all frames along the vehicle trajectory to expand reliable terrain regions outward from known vehicle paths. In the online setting, only recent frames are considered [11]; terrain heights for each grid cell are estimated from low-variance observations, followed by a Bayesian smoothing step to improve robustness.

Once h_{terr} is obtained, incoming LiDAR frames $\mathbf{P}^t$ are filtered to remove points below the terrain, leaving only above-ground points for dynamic analysis. The remaining points are then mapped to a rolling BEV grid G of size $W \times H$, with each cell covering $w \times w$ meters. Each point $\tilde{p}^t$ is rotated into a rotation-invariant frame and translated relative to the grid center at $(L_x^{t_0}, L_y^{t_0})$, producing grid coordinates (U, V):

$$\begin{cases} U = \dfrac{W}{2} + \left\lfloor \dfrac{\tilde{x}^t + L_x^t - L_x^{t_0}}{w} \right\rfloor \\ V = \dfrac{H}{2} - 1 - \left\lfloor \dfrac{\tilde{y}^t + L_y^t - L_y^{t_0}}{w} \right\rfloor \end{cases}. \tag{1}$$

Observed cells are fused with accumulated results to produce a dense 2D BEV terrain map, where each cell stores the estimated terrain height h_{terr}. This preprocessing ensures that the subsequent point-level dynamic analysis operates only on non-terrain points, simplifying dynamic object detection and improving robustness.

B. Point-level Dynamic Analysis

For efficient dynamic attribute analysis, we store a sequence of N temporal adjacent frames. Each point $\tilde{p}^t$ is firstly projected to all the N temporal adjacent frames, and in each frame, we judge if this point can find a match, i.e. a nearby point within a small distance threshold. If the point cannot find a match, it increases its chance of being a dynamic point, and vice versa.

We use the following formula to combine the N comparison results into a single dynamic confidence score S^t:

$$S^t = \sum_{k=1}^{N} \omega^k \cdot \mathbb{I}_k\left(\tilde{p}^t\right), \tag{2}$$

where $\mathbb{I}_k(\cdot)$ is an indicator function indicating that if point $\tilde{p}^t$ can find a match in the previous k-th frame. ω^k is a weight function of the k-th frame. We prefer to assign a larger weight to old frames than recent frames. As a simple case, ω^k can be defined as $N-k$.

We then use log-odds ratio to transform S^t into a zero-centered value β^t:

$$\beta^t = \ln\left(\frac{S^t}{S_m - S^t}\right), \tag{3}$$

where S_m denotes the upper bound of the dynamic confidence score, defined as the summation of weights across all temporally adjacent frames.

The dynamic score β^t now becomes a real value: A positive value indicates that the point is more biased toward being static, while a negative value suggests a stronger tendency toward being dynamic.

C. Grid-level Dynamic Analysis

The point-level dynamic analysis module proposed in the previous section provides preliminary motion estimates but still suffers from noise, especially in occluded regions where the computationally expensive ray casting operator is not applied. To enhance robustness, we aggregate point-level results at the BEV grid level, where evidence can be accumulated and noise suppressed.

For a grid cell G_c, we use two complementary features to represent its spatial structure and temporal property: the spatial feature f_s and the temporal feature f_t. f_s encodes the vertical distribution of points, while f_t characterizes how such structures evolve across time. Together, they provide a compact and robust representation of the environment.

Spatial feature. We first select dynamic points with $\beta^t > 0$ in G_c and compute the mean and variance of their height values to form a Gaussian distribution. This models both the typical elevation and its uncertainty, distinguishing flat terrain from vertical obstacles. To mitigate outliers and sparsity, we further fuse Gaussians across frames using the Normal Distributions Transform (NDT) [22], obtaining a compact joint distribution $f_s = \mathcal{N}(\mu, \Sigma)$.

Temporal feature. The temporal feature f_t is designed to capture the persistence of dynamic evidence. For each frame, we extract the maximum confidence score S^t_{MAX} and log-odds ratio β^t_{MAX}, as maximum values highlight the strongest evidence and avoid dilution by weak signals. The frame-level f_t is then calculated via a bidirectional decay module:

$$f_t = \begin{cases} S^t_{MAX} \cdot \exp(\alpha) & \text{if } \beta^t_{MAX} > 0 \\ (S^t_{MAX} - S_m) \cdot \exp(\alpha) & \text{otherwise} \end{cases}, \tag{4}$$

where α denotes the duration that the sign of β_{MAX} remains unchanged in preceding frames. This formulation emphasizes long-term consistency while suppressing transient fluctuations.

Due to point cloud sparsity, many grid cells may lack valid f_s. To achieve spatial continuity, we adopt the Bayesian Generative Kernel Inference (BGKI) method [11], which interpolates features while weighting them by temporal stability. Given $\{f_s^c = \mathcal{N}(\mu_c, \Sigma_c)\}$ for observed cells, the inferred feature at target location G_* is

$$\begin{cases} \mu_* = \dfrac{\sum_{c=1}^{N_o} \frac{\mu_c}{\Sigma_c} \delta_c k(G_c, G_*) + \frac{\mu_0}{\Sigma_0}}{\sum_{c=1}^{N_o} \frac{1}{\Sigma_c} \delta_c k(G_c, G_*) + \frac{1}{\Sigma_0}} \\ \Sigma_* = \dfrac{1}{\sum_{c=1}^{N_o} \frac{1}{\Sigma_c} \delta_c k(G_c, G_*) + \frac{1}{\Sigma_0}} + \Sigma_* \end{cases}, \tag{5}$$

where $k(\cdot,\cdot)$ is the kernel function, and μ_0, Σ_0 are prior parameters. The temporal weight

$$\delta_c = \exp\left(-\left(\frac{f_t^*}{f_t^c}\right)^2\right), \tag{6}$$

increases the contribution of stable cells with large $|f_t|$, ensuring that consistent dynamics dominate the interpolation.

Finally, we combine f_s and f_t to estimate the minimum dynamic height h_{min} for a grid cell:

$$h_{min} = \begin{cases} \mu - 3\sqrt{\Sigma} & \text{if } f_t > 0 \\ +\infty & \text{otherwise} \end{cases}, \tag{7}$$

Here, $\mu - 3\sqrt{\Sigma}$ serves as a conservative lower bound for structures above terrain. Thus, each grid cell has two height values: the terrain height h_{terr} and the minimum height h_{min}. Since $h_{min} > h_{terr}$, dynamic objects are constrained to appear within the interval $h_{terr} < h < h_{min}$, achieving a principled separation between static terrain and dynamic structures.

D. Point-level Decision-making

As shown in Figure 4, the blue line represents the h_{min} value and the yellow line represents the terrain height h_{terr} calculated in section III-A.

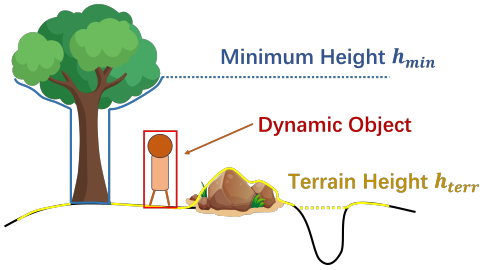


Fig. 4. Illustration of minimum height and terrain height. Solid lines indicate the minimum height at corresponding positions, blue dashed lines denote regions with infinite minimum height (i.e., $f_t < 0$)

Based on these two values, we could easily judge the dynamic state of a point by comparing its height value z with h_{terr} and h_{min} by:

$$\text{State} = \begin{cases} \text{dynamic} & \text{if } h_{terr} < z < h_{min} \\ \text{static} & \text{otherwise} \end{cases}. \tag{8}$$

In Fig.4, the h_{min} and h_{terr} values for the tree part are both small, and the tree will be considered as a static object. In contrast, the h_{min} and h_{terr} value for the rock part could either be a small value beneath the rock or a relatively large value corresponding to the surface of the rock. In either of these cases, the rock will be considered a static object according to (8).

IV. Experimental Setup

We devise two experimental settings to evaluate the algorithm: the online setting and the offline setting. In the offline setting, the algorithm can process the whole as a dataset in batch, while in the online setting, it can only process the frames sequentially. In the offline experiments, we compare the performance with Removert, ERASOR, OctoMap, and Deflow. For the online experiments, we compare with Dynablox and DUFOMap. PGP-DOR denotes the offline version of our method, where both h_{terr} and h_{min} are computed using all frames [18]. PGP-DOR* refers to the online mode, in which both h_{terr} and h_{min} are estimated from local historical frames [11]. PGP-DOR+ is a variant of the online setting, where h_{terr} is precomputed with the method in [18], while h_{min} is still obtained online.

Ablation studies assess the contributions of the bidirectional decay module described in (4) and the BGKI module described in (5).

All experiments were performed on a laptop with an Intel Core i9-13900H processor.

A. Datasets

We evaluate the proposed method using both the SemanticKITTI dataset [23] and a self-collected dataset.

The SemanticKITTI dataset, derived from the KITTI odometry data, utilizes a Velodyne HDL-64E LiDAR [24]. It encompasses urban, residential, and highway scenes, providing point-wise semantic labels and pose information for dynamic objects such as vehicles, bicycles, and pedestrians. We perform experiments on sequences 00 and 01. Following [7], experiments are only performed in a portion of the dataset where many dynamic objects exist.

To further validate our method across different LiDAR types, we collected data using a Robosense Ruby Plus-128 LiDAR. The self-collected dataset consists of two scenarios: an urban traffic scene (DOR-Urban) and an unstructured off-road scene (DOR-Rural). We manually annotate dynamic objects in this dataset. The dynamic object categories include cars, vans, and pedestrians.

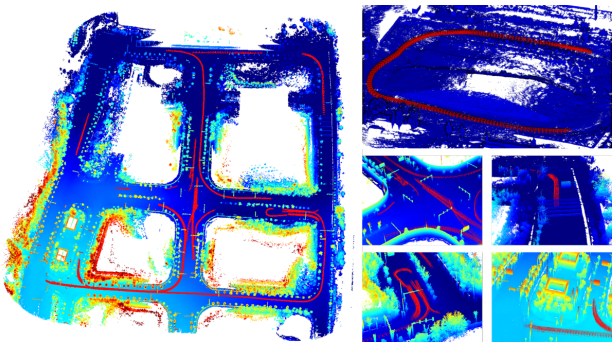


Fig. 5. The left figure shows DOR-Urban, and the top right figure shows DOR-Rural. The bottom right shows the four local views.

B. Metric

Following [25], we adopt Static Accuracy (SA %), Dynamic Accuracy (DA %), and Associated Accuracy (AA %) to evaluate the map-level dynamic removal performance. SA quantifies the proportion of correctly retained static structures, while DA reflects the correctly removed dynamic objects. The composite metric $AA = \sqrt{SA \times DA}$ is a balanced measure between SA and DA.

For frame level analysis, we employ accuracy (ACC %), false positive rate (FPR ‰), and F1-score. All metrics are computed on the original resolution without any downsampling operator.

V. Experiments

A. Quantitative results

1) Accuracy: As shown in Table I, Removert achieves a high SA score but a relatively low DA score. In contrast, ERASOR achieves superior DA but exhibits instability in SA, as demonstrated by significant variations between the KITTI 00 and KITTI 01 datasets. This suggests that ERASOR's SA is suboptimal in confined environments, such as small towns. Similarly, although OctoMap achieves the highest DA score, its SA is comparatively lower, particularly on KITTI 01, where the prevalence of dynamic obstacles leads to a marked decline in static accuracy. Deflow shows competitive overall accuracy in offline scenarios, but its DA performance degrades on highway and rural scenes, limiting its generalization capability. In comparison, our PGP-DOR algorithm, benefiting from its flexible update mechanism, consistently delivers the best overall performance across all datasets.

Regarding the online methods, Dynablox employs a conservative strategy, leading to a lower DA on the DOR-Rural dataset. In contrast, DUFOMap exhibits balanced performance across all datasets. Although the online version of our algorithm (PGP-DOR*) experiences a slight reduction in SA compared to its offline counterpart, its DA remains robust. By incorporating information from adjacent frames, the PGP-DOR+ variant obtains a remarkable performance comparable to the offline version.

2) Efficiency: Figure 7 presents a runtime comparison of various algorithms. Offline methods generally exhibit longer runtimes, with OctoMap being the slowest due to its unoptimized octree updates. DUFOMap benefits from the optimized octree insertion of UFOMap, thereby reducing its overall runtime. As a learning-based approach, Deflow shows the shortest average runtime across both datasets, though with higher variance mainly stemming from initialization overheads. In contrast, PGP-DOR achieves a balanced trade-off between efficiency and accuracy by employing a BEV grid-based representation for dynamic analysis. Moreover, PGP-DOR+ further reduces runtime by directly utilizing the precomputed h_{terr} values, making it more efficient while retaining robustness.

B. Qualitative results

Figure 6 shows the three-dimensional reconstruction of the test scene, the ground truth, and the dynamic detection results obtained by comparative algorithms. Several observations can be made from these results: Removert exhibits numerous missed detections; ERASOR produces lots of false-positive regions; OctoMap suffers from excessive removal of ground

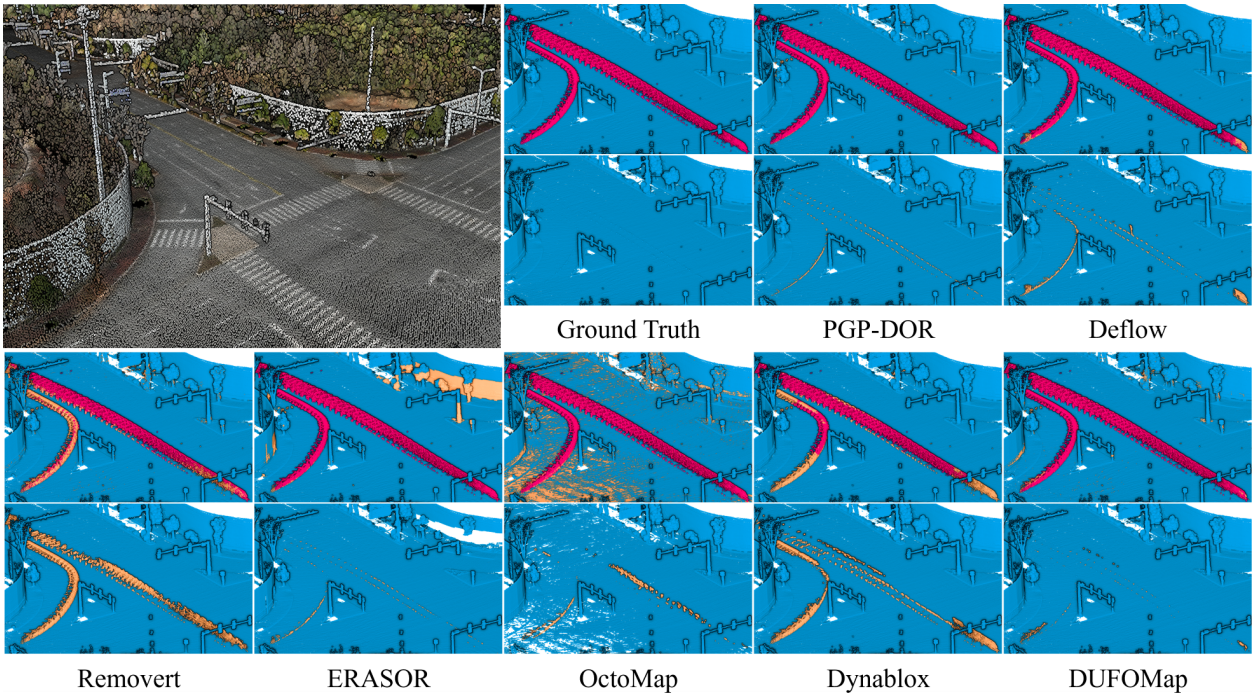


Fig. 6. Comparison of dynamic object removal results across different algorithms. For each method, the top row presents the segmentation results of all points, while the bottom row shows the scene after removing the detected dynamic points. In the visualization, red indicates correctly classified dynamic objects, blue represents correctly classified static objects, and yellow denotes misclassified regions.

TABLE I
PERFORMANCE COMPARISON OF VARIOUS ALGORITHMS FOR REMOVING DYNAMIC POINTS

	KITTI Small Town (00)			KITTI Highway (01)			DOR-Urban (Ours)			DOR-Rural (Ours)		
Methods	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑
Removert [8]	99.44	41.53	64.26	97.81	39.56	62.20	99.89	31.10	55.74	99.78	54.36	73.65
ERASOR [9]	66.70	98.54	81.07	98.12	90.94	94.46	87.22	98.64	92.75	92.70	44.73	64.40
OctoMap [15]	68.05	99.69	82.37	55.55	99.59	74.38	95.03	91.91	93.46	94.91	96.61	95.76
Deflow [5]	99.91	92.74	96.25	99.82	82.30	90.64	99.97	96.03	97.98	99.97	61.67	78.52
PGP-DOR(Ours)	98.63	97.60	**98.11**	99.39	93.05	**96.17**	99.91	98.16	**99.03**	99.43	99.39	**99.41**
Dynablox [4]	96.76	90.68	93.67	96.33	68.01	80.94	99.99	61.92	78.69	99.95	25.62	50.60
DUFOMap [7]	98.37	92.37	95.31	98.48	81.34	89.50	99.78	97.24	98.50	99.45	96.34	97.88
PGP-DOR*(Ours)	98.41	97.60	98.00	91.58	94.31	92.93	96.46	99.70	98.07	99.22	99.39	99.30
PGP-DOR+(Ours)	98.62	97.59	**98.10**	98.19	92.46	**95.28**	98.99	99.02	**99.00**	99.34	99.38	**99.36**

[1] DUFOMap*, PGP-DOR*, and PGP-DOR+ all utilize the latest online scanning results per frame as input to ensure real-time execution. PGP-DOR+ specifically stores precomputed h_{terr} values in advance.
[2] The best results are shown in **bold** and the second best results are shown in underlined. Results are in percentage.

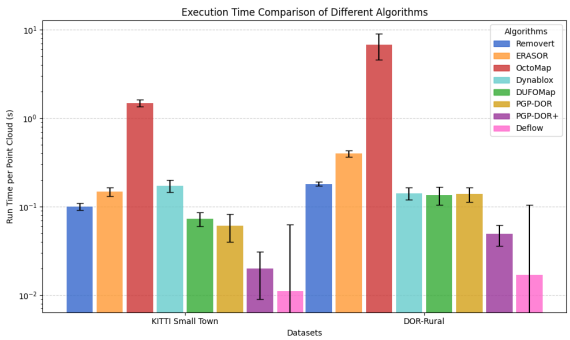
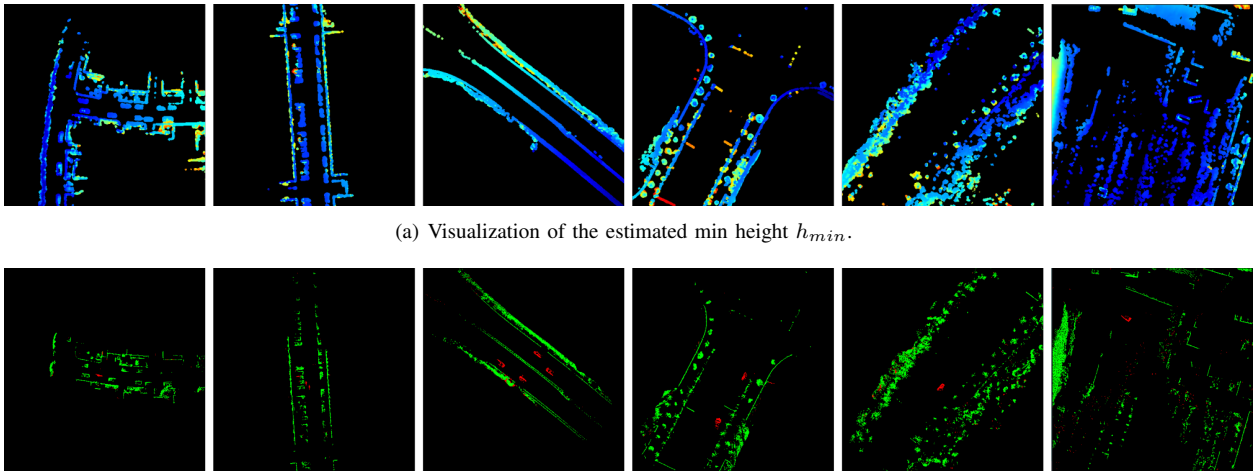


Fig. 7. Processing time of different algorithms.

points and a significant number of missed detections. Deflow shows relatively good static preservation, but its dynamic segmentation deteriorates in distant or boundary regions due to perception range limitations and observation constraints. The Dynablox method, owing to its conservative nature, fails to entirely remove occluded vehicles. While DUFOMap generally performs well, a few missed detections still occur. In contrast, our PGP-DOR method demonstrates superior overall performance, with almost no noticeable missed detections, and exhibits excellent suppression of false positives.

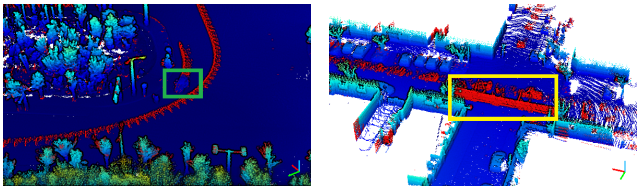
Figure 9(a) illustrates a scenario in which a dynamic vehicle suddenly stops. Figure 9(b) shows a scene with large dynamic vehicles. In both of these two scenes, our method performs consistently well, demonstrating its ability to swift changes and processing unseen objects.



(a) Visualization of the estimated min height h_{min}.

(b) Real-time dynamic detection Results in the BEV map.

Fig. 8. Illustration of minimum height (top) and real-time dynamic detection results in the BEV map (bottom). Colors range from blue (low elevation) to red (high elevation), with red in BEV indicating dynamic grids and green representing static grids.



(a) A scene where a dynamic vehicle suddenly stopped.

(b) Large vehicles moving at a relatively slow speed.

Fig. 9. Visualization of dynamic detection results: (a) A scene where a dynamic vehicle suddenly stopped. (b) A scene with a slow-moving large vehicle. Dynamic objects are colored in red. The other points are colored by elevation.

In summary, our algorithm demonstrates robust dynamic detection performance across a variety of testing scenarios, exhibiting robustness to occlusions and generating few false positives.

C. Ablation study

In this section, we evaluate the impact of the BGKI module m_β and the bidirectional decay module m_τ on the overall performance of the algorithm. In table II, we present the performance of the PGP-DOR algorithm after the removal of each module.

TABLE II
ABLATION STUDY OF PGP-DOR

Parameter settings	SA [%] ↑	DA [%] ↑	AA [%] ↑
w/o m_β, m_τ	44.28	96.50	65.37
w/o m_β	43.72	99.99	66.12
w/o m_τ	99.59	95.26	97.40
Final PGP-DOR	99.43	99.39	**99.41**

From Table II, it can be observed that the BGKI module significantly enhances static accuracy (over 55%), as it primarily functions to smooth and expand the spatial features of the grid cells. Meanwhile, the bidirectional decay module is designed to reduce false positives caused by dynamic state transitions, primarily enhancing the dynamic accuracy (approximately 5%).

From the results presented in the table, it is evident that both of these two modules help improve the algorithm's performance.

D. BEV experiment

In this section, we evaluate the performance of the PGP-DOR algorithm at the BEV grid level. Fig. 8(a) and Fig. 8(b) present visualizations of the h_min value and the BEV segmentation results. It shows that h_min can effectively represent the static environment and our PGP-DOR algorithm can effectively detect the dynamic objects.

TABLE III
COMPARISON OF BEV RESULTS FOR ONLINE ALGORITHMS

Methods	KITTI Small Town (00)		
	ACC [%] ↑	FPR [‰] ↓	F1 [%] ↑
Dynablox [4]	99.97	0.11	68.75
DUFOMap [7]	99.51	4.83	20.69
PGP-DOR(Ours)	99.94	0.52	77.09
Methods	**DOR-Urban (Ours)**		
	ACC [%] ↑	FPR [‰] ↓	F1 [%] ↑
Dynablox [4]	99.94	0.23	59.34
DUFOMap [7]	99.82	1.59	39.04
PGP-DOR(Ours)	99.96	0.39	79.56

Table III summarizes the BEV-level performance of various algorithms. Overall, DUFOMap exhibits a relatively high false positive rate and lower precision, leading to a reduced overall F1-score. In contrast, Dynablox employs a more conservative strategy, which effectively reduces the false positive rate. However, due to its lower recall rate, its F1-score remains suboptimal. The PGP-DOR algorithm successfully balances recall and precision while maintaining a moderate false positive rate, ultimately achieving the highest F1-score among the evaluated methods.

VI. Conclusion

This paper introduces PGP-DOR, a dynamic object removal method based on the Point-Grid-Point scheme. Extensive evaluations against state-of-the-art methods across various dynamic scenes demonstrate its superior real-time performance. Additionally, further ablation studies and BEV-level experiments validate the effectiveness of its modular design, adaptability, and robustness.

Despite its advantages, the grid-level dynamic analysis relies on multi-frame accumulation, making it susceptible to false positives in early observations, which adversely affects SA in real-time scenarios. Additionally, failures in terrain modeling can impede the accurate characterization of near-ground dynamic objects, thereby reducing DA. Future improvements may involve integrating learning-based methods or scene flow estimation techniques, as well as incorporating segmentation and spatio-temporal fusion strategies to enhance overall performance.

References

[1] 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 Robotics and Automation Letters*, vol. 8, no. 8, pp. 5180–5187, 2023.

[2] 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 *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 11 456–11 463.

[3] N. Wang, C. Shi, R. Guo, H. Lu, Z. Zheng, and X. Chen, "Insmos: Instance-aware moving object segmentation in lidar data," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 7598–7605.

[4] L. Schmid, O. Andersson, A. Sulser, P. Pfreundschuh, and R. Siegwart, "Dynablox: Real-time detection of diverse dynamic objects in complex environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6259–6266, 2023.

[5] Q. Zhang, Y. Yang, H. Fang, R. Geng, and P. Jensfelt, "Deflow: Decoder of scene flow network in autonomous driving," 2024.

[6] S. Pagad, D. Agarwal, S. Narayanan, K. Rangan, H. Kim, and G. Yalla, "Robust method for removing dynamic objects from point clouds," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 765–10 771.

[7] D. Duberg, Q. Zhang, M. Jia, and P. Jensfelt, "Dufomap: Efficient dynamic awareness mapping," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5038–5045, 2024.

[8] G. Kim and A. Kim, "Remove, then revert: Static point cloud map construction using multiresolution range images," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 758–10 765.

[9] 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 Robotics and Automation Letters*, vol. 6, no. 2, pp. 2272–2279, 2021.

[10] 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 *Robotics: Science and Systems (RSS 2023)*, 2023.

[11] H. Xue, H. Fu, L. Xiao, Y. Fan, D. Zhao, and B. Dai, "Traversability analysis for autonomous driving in complex environment: A lidar-based terrain modeling approach," *Journal of Field Robotics*, vol. 40, no. 7, pp. 1779–1803, 2023.

[12] X. Meng, N. Hatch, A. Lambert, A. Li, N. Wagener, M. Schmittle, J. Lee, W. Yuan, Z. Chen, S. Deng, *et al.*, "Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation," *arXiv preprint arXiv:2303.15771*, 2023.

[13] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, " PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation ," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, July 2017, pp. 77–85.

[14] J. Zhu, H. Li, Z. Wang, S. Wang, and T. Zhang, "i-octree: A fast, lightweight, and dynamic octree for proximity search," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 12 290–12 296.

[15] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, pp. 189–206, 2013.

[16] D. Duberg and P. Jensfelt, "Ufomap: An efficient probabilistic 3d mapping framework that embraces the unknown," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6411–6418, 2020.

[17] D. Meagher, "Geometric modeling using octree encoding," *Computer graphics and image processing*, vol. 19, no. 2, pp. 129–147, 1982.

[18] H. Fu, H. Xue, and G. Xie, "Mapcleaner: Efficiently removing moving objects from point cloud maps in autonomous driving scenarios," *Remote Sensing*, vol. 14, no. 18, p. 4496, 2022.

[19] M. Arora, L. Wiesmann, X. Chen, and C. Stachniss, "Static map generation from 3d lidar point clouds exploiting ground segmentation," *Robotics and Autonomous Systems*, vol. 159, p. 104287, 2023.

[20] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765.

[21] Z. Yuan, X. Wang, J. Wu, J. Cheng, and X. Yang, "A fast dynamic point detection method for lidar-inertial odometry in driving scenarios," *arXiv e-prints*, p. 2407, 2024.

[22] J. Saarinen, H. Andreasson, T. Stoyanov, J. Ala-Luhtala, and A. J. Lilienthal, "Normal distributions transform occupancy maps: Application to large-scale online 3d mapping," in *2013 IEEE international conference on robotics and automation*, 2013, pp. 2233–2238.

[23] 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 *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9296–9306.

[24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[25] Q. Zhang, D. Duberg, R. Geng, M. Jia, L. Wang, and P. Jensfelt, "A dynamic points removal benchmark in point cloud maps," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 608–614.