GJR Publication
OPEN ACCESS

Global Journal of Research in Engineering & Computer Sciences
ISSN: 2583-2727 (Online)
Volume 06 | Issue 03 | May-June | 2026
Journal homepage: https://gjrpublication.com/gjrecs/

Research Article

A Method for Dynamic Environmental Awareness Based on a Probabilistic Multi-Resolution Framework

Hao Tang[1], *Caigen Zhou[1], Yong Li[2], Licai Zhu[1], Hao Yang[1]

[1]College of Information Engineering, Yancheng Teachers University, Yancheng 224000, China
[2]College of Computer and Information Engineering, Nanjing Tech University, Nanjing 210000, China

DOI: 10.5281/zenodo.20311843 **Submission Date: 30 March 2026 | Published Date: 20 May 2026**

***Corresponding author: Caigen Zhou**
College of Information Engineering, Yancheng Teachers University, Yancheng 224000, China.

Abstract

To address the issues of laser point cloud mapping in dynamic scenes being susceptible to interference from moving objects and the generation of motion artifacts, this paper proposes a dynamic environment awareness method based on a probabilistic multi-resolution framework. This method incorporates a two-state posterior probability model and a log-odds recursive update mechanism to fuse multi-frame observation data as probabilistic evidence. Additionally, it constructs a multi-resolution voxel representation and designs an anti-overaccumulation mechanism to enhance the stability and robustness of distinguishing dynamic points from static points. Experimental results on the Semantic KITTI benchmark sequences and the Tree Scope dataset demonstrate that the proposed method can generate cleaner static maps.

Keywords: *Dynamic Environment; Dynamic Point Cloud Removal; Static Map Construction.*

Introduction

Orchard environments are typical “complex dynamic environments,” which place higher demands on robotic autonomous navigation[Error! Reference source not found.]. Unlike structured industrial environments, orchard scenarios are characterized by their dynamic nature. This includes rigid dynamic objects such as workers and other moving equipment, as well as non-rigid dynamic disturbances such as swaying branches and leaves and local occlusions.

LiDAR-based simultaneous localization and mapping (SLAM) technology is key to ensuring that robots can operate around the clock without being affected by lighting conditions[Error! Reference source not found.]. However, when multiple frames of point clouds collected by the robot are directly overlaid to create a map, swaying branches and moving obstacles can cause severe “ghosting” and “trailing” in the point cloud map, significantly compromising the map’s purity[Error! Reference source not found.]. Existing dynamic point cloud removal algorithms, such as Removert[Error! Reference source not found.], can effectively handle fast-moving rigid objects like vehicles in urban environments. However, the “deterministic binary decision” rules on which they rely are prone to misclassification when dealing with the constantly present and swaying branches and leaves in orchards. Either the branches and leaves are mistakenly identified as dynamic noise and filtered out, resulting in an incomplete tree model; or, due to their persistent nature, misclassify them as static structures, preventing the robot from planning precise operational paths near the fruit trees.

Some studies have adopted visibility-based methods to reduce computational load. This type of method associates query points with map points within a nearly identical narrow field of view (FOV)[Error! Reference source not found.]; if a query point is observed behind a previously acquired point in the map, the previously acquired point is considered dynamic. This method is simple and intuitive. Early research proposed visibility-based methods for detecting moving objects[Error! Reference source not found.], focusing on iteratively updating the state of map points by continuously fusing multiple measurements and checking visibility between the query scan and the map. In recent years, significant progress has been made by combining visibility principles with semantic segmentation techniques for point clouds. Kim et al. used SLAM technology to obtain a prior map of the entire environment and then filtered out moving target points from the point cloud based on visibility principles. Chen et al. proposed the LiDAR-MOS framework[Error! Reference source not found.], capable

of real-time detection and filtering of moving objects. Their method utilizes three classic point cloud segmentation networks—RangeNet++[Error! Reference source not found.], SalsaNext[Error! Reference source not found.], and MINet[Error! Reference source not found.]—to project the point cloud onto a range image and compute the residuals between the current and previous scans based on visibility. The residual image is then combined with the current range image and fed into a fully convolutional neural network. This network is trained using binary labels to effectively distinguish between moving and stationary objects. Compared to the method proposed by Kim et al., the advantage of this approach is that it does not rely on pre-constructed maps; instead, it operates online using only past LiDAR scans, thereby improving real-time performance. However, a potential drawback is that it may fail to detect some dynamic objects that are temporarily stationary for a certain period of time.

To address the need for static map construction in dynamic scenarios, this study investigates methods for dynamic environment perception. To address the limitations of traditional dynamic point removal methods—which are prone to pose errors, local occlusions, and non-rigid distortions in complex scenes—this paper proposes a dynamic environment perception method based on a probabilistic multi-resolution framework.

1. A Probabilistic Multi-Resolution Method for Dynamic Point Cloud Removal

1.1 Two-state posterior and log-odds update model

Remora Optimization Algorithm[Error! Reference source not found.] was proposed by Jia et al. in 2021, which focuses on simulating remora adsorbing on hosts of different sizes to complete the foraging process. This algorithm covers both the sailfish optimization algorithm and the whale optimization algorithm. Remoras have an elongated body with a flattened head that gradually transitions to a cylindrical shape at the back. It is characterized by a sucker on top that evolved from the first dorsal fin: remoras are known for their ability to attach themselves to whales or other marine creatures as well as to ocean-going vessels. This habit not only saves them energy, but also protects them from predators. Remoras are usually found in tropical waters but move with their hosts to cooler areas. They feed mainly on other fish and invertebrates. When they reach food-rich areas, remoras temporarily leave their hosts to feed, then attach themselves to new hosts and continue their migration to other areas.

1.1.1 State Variables and Probabilistic Representations

To perform dynamic point removal for each spatial element (voxel), we introduce a binary random variable $S_i \in \{static, dynamic\}$ to indicate whether the element is static. This binary state does not simply classify spatial elements as static or dynamic, but rather describes the long-term properties exhibited by a spatial element across multiple frames in probabilistic terms. If a spatial element is consistently and stably supported as static across consecutive observations, its static posterior probability will gradually increase; conversely, if a spatial element appears only briefly at localized moments or exhibits significant spatial inconsistency across different time points, its static posterior probability will gradually decrease. Through this approach, this paper does not maintain instantaneous state labels, but rather a static confidence metric that is continuously updated based on the observation history. We maintain the posterior probability of it being static: $p_i = P(S_i = static \mid Z)$, and use the log-odds to represent its confidence, as shown in Equation 1:

$$L_i = \log \frac{p_i}{1 - p_i} \tag{1}$$

The log-odds has the advantage of additive accumulation[Error! Reference source not found.], allowing multiple observation updates to be accumulated smoothly, thereby avoiding numerical instability in the probability values.

1.1.2 Construction of observational evidence

A depth image is generated for each frame of laser point cloud data by projecting it onto a spherical range image $I^Q_{k,r}(u,v)$, which is then compared with an existing map to produce the map's depth image $I^M_{k,r}(u,v)$. The depth difference is calculated as shown in Equation 2:

$$d_{k,r}(u,v) = I^Q_{k,r}(u,v) - I^M_{k,r}(u,v) \tag{2}$$

Based on the magnitude of the depth difference and the validity of the pixel, the status of each pixel is classified into the following four categories: (1) Support (static): $|d_{k,r}| \le \varepsilon_r$; (2) Conflict: $d_{k,r} > \tau_r$ (the depth difference between the

current frame and the map is significant, possibly indicating a dynamic object); (3) Occlusion: $d_{k,r} < -\varepsilon_r$ (the depth of the current frame is small, possibly indicating occlusion); (4) Unknown. I^Q or I^M is invalid (the pixel has not detected a point or is outside the field of view). Here, ε_r and τ_r represent the static support threshold and conflict threshold, respectively, with their values set based on the actual scene and point cloud density. To accommodate noise at long distances, the conflict threshold is determined adaptively; the adaptive algorithm is shown in Equation 3:

$$\tau_r(u,v) = \alpha_r \cdot I^M_{k,r}(u,v) + \tau_{0,r} \tag{3}$$

Here, $\tau_r(u,v)$ represents the collision detection threshold at pixel (u,v) in the resolution level r; α_r is a scale-dependent scaling factor used to control the rate at which the threshold increases with depth; and $\tau_{0,r}$ is the base threshold, which ensures that a minimum detection margin is maintained in close-range areas.

To illustrate more clearly the geometric relationship between current frame observations and existing maps, as well as the corresponding types of evidence, this paper presents a schematic diagram of the construction of observational evidence, as shown in Figure 1.

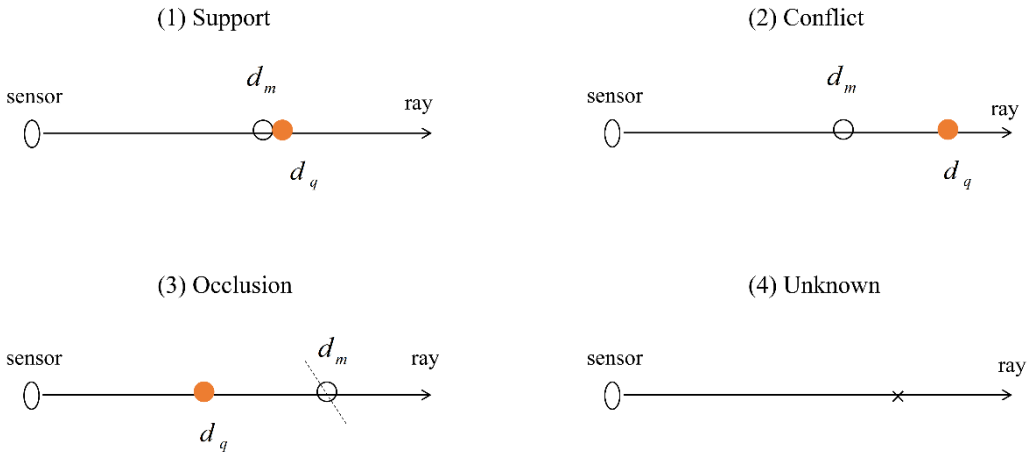

Figure 1: Schematic Diagram of Observational Evidence Construction.

1.1.3 Log-odds incremental update

For each observation event $z_{k,r}$ in each frame, we update the log-odds values of the spatial elements based on the type of evidence, L_i as shown in Equation 4:

$$L_i \leftarrow L_i + \Delta L(z_{k,r}) \tag{4}$$

In particular, the update magnitude for conflicting evidence is larger, while that for supporting evidence is smaller. The specific update rules are shown in Equation 5:

$$\Delta L(z_{k,r}) = \begin{cases} +\eta_r \cdot (1 - p_i), & \text{if } z_{k,r} = \text{support} \\ -\kappa_r \cdot g(d_{k,r}), & \text{if } z_{k,r} = \text{conflict} \\ 0, & \text{if } z_{k,r} \in \{occlusion, unknown\} \end{cases} \tag{5}$$

where $\Delta L(z_{k,r})$ denotes the log-odds increment of a spatial cell caused by an observation event at the k th frame and resolution level r; the η_r update rate of supporting evidence and the κ_r update rate of conflicting evidence, $\kappa_r > \eta_r$ is typically set. $g(d)$ is the conflict strength function, which is usually linear or sigmoid, is used to normalize the depth difference, as shown in Equation 6:

$$g(d) = \min(1, \frac{d - \tau_r}{\sigma_r}), d > \tau_r \quad (6)$$

Here, d represents the depth difference between corresponding positions, τ_r is the conflict threshold at the resolution level r, and σ_r is the scaling parameter for conflict intensity. This design ensures that even when dynamic objects have weak supporting evidence across multiple scales, they can still be effectively "pulled back" by strong conflicting evidence.

1.2 Multi-resolution Fusion and Anti-Overlapping Mechanism

1.2.1 The Need for Multi-Resolution Fusion

In practical applications, spatial representations at a single resolution often struggle to balance the preservation of local details with the stability of global decisions. Multi-resolution approaches can effectively address issues arising from pose errors, boundary effects, and rapid changes in dynamic objects. Different resolutions allow for higher accuracy at fine scales[Error! Reference source not found.] and improved computational efficiency at coarse scales. Fine resolution (smaller voxel size) is highly sensitive to minute positional changes, while coarse resolution (larger voxel size) can tolerate a certain degree of error and noise.

Therefore, adopting a voxel-based multi-scale fusion approach allows for the simultaneous acquisition of both detailed and global information across different scales, thereby enabling better identification and removal of moving objects in dynamic and complex environments.

In complex dynamic environments such as orchards, this multi-scale complementarity is particularly crucial. Major structural elements like tree trunks and road boundaries typically exhibit high stability at coarse scales, whereas branches, foliage, local occlusions, and moving obstacles rely more heavily on fine-scale representations for accurate differentiation. The multi-resolution fusion strategy adopted in this paper essentially involves evaluating the same spatial element simultaneously at different observation scales: the fine scale supplements local details, while the coarse scale provides overall consistency constraints. This approach not only avoids excessive sensitivity to local jitter at the fine scale but also helps reduce excessive loss of boundary information at the coarse scale, thereby achieving a more reasonable balance between dynamic point recognition and the preservation of static structures.

1.2.2 The Integration of Independent Evidence and Issues of Evidentiary Relevance

The issue of evidence correlation is a critical challenge that must be addressed in multi-resolution probabilistic fusion methods. Since evidence generated from the same frame and the same point cloud at multiple resolutions is highly correlated, direct summation may result in double counting. To avoid this problem, the following strategies to prevent excessive accumulation are proposed:

(1) Intra-frame fusion: First, fuse evidence from all resolutions within the same frame, then update the historical log-odds. Specifically: 1. Conflict Priority: If conflicting evidence arises at any scale, the update is based on the intensity of the conflict. 2. Support-Based Update: If there are no conflict and the supporting evidence at least one scale is strong, a small-scale update is applied. 3. Occlusion/Unknown: No update is performed for occluded or unknown regions.

(2) Incremental Clipping: Clip the total increment of updates for each frame to ensure that the update magnitude for a single frame does not become excessive. Specifically, limit the log-odds increment to within a specified range.

(3) Support Evidence Saturation and Forgetting Factor: To prevent long-term accumulation of supporting evidence from misclassifying dynamic objects as static, a support evidence saturation mechanism is adopted. The update mechanism is shown in Equation 7:

$$\Delta L^{\text{support}} = +\eta_r \cdot (1 - p_i) \quad (7)$$

At the same time, the introduction of a forgetting factor λ causes the historical evidence to gradually decay, as shown in Equation 8:

$$L_i \leftarrow (1 - \lambda) L_i + \Delta L_k \quad (8)$$

In particular, the forgetting factor is typically set to $10^{-3} \sim 10^{-2}$.

1.2.3 Dual-Threshold Decision-Making and Uncertain Outputs

Once the posterior probability p_i has been calculated, a two-threshold strategy is employed for decision-making, as shown in Equation 9:

$$static(i) = \begin{cases} static, \ p_i \geq \tau_{sta} \\ dynamic, \ p_i \leq \tau_{dyn} \\ uncertain, \ \tau_{dyn} < p_i < \tau_{sta} \end{cases} \quad (9)$$

Here, τ_{sta} and τ_{dyn} are the thresholds for static and dynamic states, respectively; typically, is set to 0.7 and to 0.3. For regions of uncertainty, this paper adopts a deferred decision strategy, temporarily refraining from classifying them as static or dynamic, and postponing classification until sufficient observational evidence has been accumulated.

2. Experimental Validation

2.1 Experimental Setup

To ensure the reproducibility of the dynamic point cloud removal experiments and the fairness of the comparisons, this paper uniformly sets the random seed to 7. Two datasets were selected for validation: the SemanticKITTI 00 sequence and the TreeScope orchard scene sequence UCM-0822-11.

The SemanticKITTI 00 sequence is input in a uniform format, with processing starting from frame 0. A total of 20 frames are selected, with a frame interval set to 1. Additionally, to control the computational scale, a maximum of 8,000 points are retained per frame. For TreeScope, a synthetic dynamic scene based on a BAG file was used as input. Processing began at frame 10, with a total of 20 frames selected, and the frame interval was also set to 1; the maximum number of points per frame was limited to 9,000. Considering the lack of dynamic targets in the original orchard scene data, this paper further introduced synthetic dynamic targets into the TreeScope dataset to enhance the validation rigor of the dynamic point cloud removal algorithm. Specifically, two dynamic targets were set up, each consisting of 260 points, with a movement speed of 0.8 m/s and a height of 1.72 m. Additionally, a moving vehicle target was included, comprising 420 points, with dimensions set at 1.8 m in length, 0.9 m in width, and 1.5 m in height.

Regarding algorithm parameter settings, this paper compares four methods: ERASOR[Error! Reference source not found.], OctoMap[Error! Reference source not found.], Removert, and the proposed method, PaperFusion. For the OctoMap method, the voxel support threshold is set to 4. For the ERASOR method, the voxel support threshold is set to 4, the mesh support threshold to 12, and the minimum ground clearance threshold to 0.8 m. For the Removert method, the voxel support threshold was set to 3, and the neighborhood support threshold was set to 60. For the method proposed in this paper, the voxel support threshold was set to 3, the neighborhood support threshold to 60, the grid support threshold to 12, and the minimum height-above-ground threshold to 0.6 m.

Regarding result visualization, to balance display quality and rendering efficiency, the maximum number of rendering points was set to 70,000, and the local region clipping size was set to 12.0 m. These parameter settings account for differences in point cloud density across datasets and the decision characteristics of each algorithm, providing a unified experimental foundation for subsequent comparisons of dynamic point removal performance and the presentation of local details.

In addition, the main parameters required for the method described in this paper are shown in Table 1:

Table 1: Experimental Parameter Settings for Probabilistic Multi-Resolution Dynamic Point Removal

Parameters	value
Multi-resolution angular resolution sequence	2.5°，2.0°，1.5°
Downsampled voxel size	0.05m

2.2 Evaluation Criteria

To evaluate the performance of dynamic point cloud detection and removal, two key metrics are primarily used: static point retention rate (PR) and dynamic point removal rate (RR). These metrics reflect the algorithm's ability to preserve the integrity of static points while removing dynamic points, as well as its effectiveness in detecting and removing dynamic points.

The static point retention rate reflects the algorithm's ability to protect static points during the process of removing dynamic points.

Its calculation formula is shown in Equation 10:

$$PR = \frac{P_{ps}}{P_{ts}} \tag{10}$$

Here, P_{ts} represents the total number of static points in the original point cloud, while P_{ps} represents the number of static points retained. A higher PR value indicates that the algorithm makes fewer erroneous deletions of static points during the removal of dynamic points, and thus demonstrates a stronger ability to retain static points.

The dynamic point removal rate is used to evaluate the algorithm's effectiveness in identifying and removing dynamic points. Its calculation formula is shown in 11

$$RR = \frac{p_{pd}}{P_{td}} \tag{11}$$

Here, P_{td} represents the total number of moving points in the original point cloud, while P_{pd} represents the number of moving points removed. A higher RR value indicates that the algorithm is more effective at identifying and removing moving points.

To comprehensively evaluate the performance of algorithms in dynamic point cloud detection and removal tasks, "score" is adopted as the evaluation metric. We calculate a composite score that reflects the algorithm's capabilities in both retaining static points and removing dynamic points. The score is defined as the harmonic mean of the static point retention rate and the dynamic point removal rate, thereby balancing the performance of both aspects. A higher score indicates that the algorithm can more effectively retain static points while removing dynamic points, thereby demonstrating greater robustness and accuracy. The calculation formula is shown in Equation 12:

$$score = 2 \times \frac{PR \times RR}{PR + RR} \tag{12}$$

2.3 Analysis of Dynamic Point Removal Experiments

As shown in Figure 2 and Table 2, in the SemanticKITTI 00 sequence, different algorithms exhibit significant differences in their ability to preserve static points and remove dynamic points. ERASOR achieved the highest static object retention rate of 97.01%, indicating strong protection of static objects; however, its dynamic object removal rate was only 42.92%, suggesting that the method lacks sufficient capability to remove dynamic objects, and many dynamic objects remain in the scene. Removert's static point retention rate and dynamic point removal rate are 88.04% and 37.04%, respectively, with a composite score of only 54.12, indicating that it not only mistakenly removes some static points but also fails to effectively remove dynamic points in this scenario. OctoMap achieves a dynamic point removal rate of 80%, demonstrating strong dynamic point removal capability; however, its static point retention rate is only 81.19%, indicating that while this method improves the dynamic point removal rate, it results in relatively significant false deletions of static objects. In contrast, on the SemanticKITTI dataset, the proposed method achieved static point retention and dynamic point removal rates of 94.28% and 71.50%, respectively, with a composite score of 81.32—the highest among the four algorithms. This indicates that the proposed method effectively removes dynamic points while preserving static points. As shown in Figure 2, our method effectively suppresses dynamic objects while maintaining the continuity of the scene's main contours, achieving a better balance between static structure preservation and dynamic object removal.

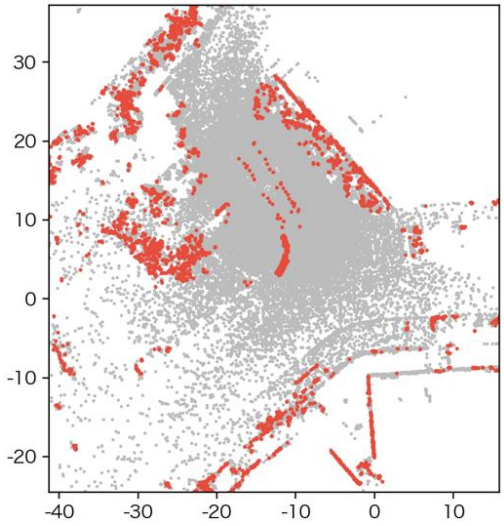

Figure 2(1) REASOR Visualization of Dynamic Point Removal

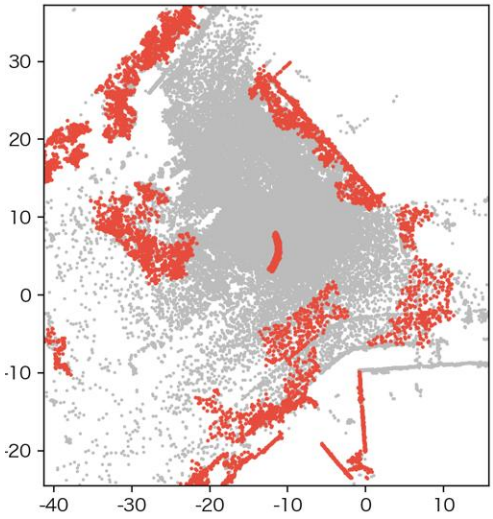

Figure 2(2) PaperFusion Visualization of Dynamic Point Removal

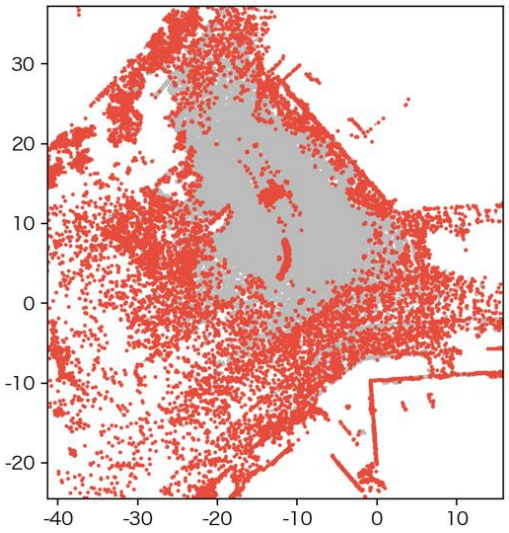

Figure 2(3) Removert Visualization of Dynamic Point Removal

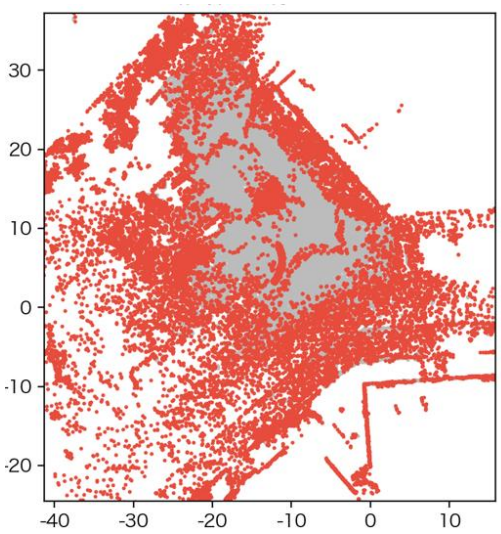

Figure 2(4) OctoMap Visualization of Dynamic Point Removal

Figure 2: SemanticKITTI 00 Sequence: Visualization of Dynamic Point Removal for 4 Algorithms

Figure 2 SemanticKITTI 00 Sequence: Visualization of Dynamic Point Removal for 4 Algorithms

Table 2 SemanticKITTI 00 Sequence: Point Count Results

Sequence	Method	PR (%)	RR (%)	score
SemanticKITTI	Removert	88.04	37.04	52.14
SemanticKITTI	OctoMap	81.19	80.00	80.59
SemanticKITTI	ERASOR	97.01	42.92	59.51
SemanticKITTI	PaperFusion	94.28	71.50	81.32

As shown in Figure 3 and Table 3, in the TreeScope orchard scene, the difficulty of dynamic point detection is further increased due to the presence of tree trunks, foliage obstruction, and highly repetitive semi-structured environmental features. ERASOR achieves a static point retention rate of 94.96% in this scene, demonstrating strong static point retention capabilities; however, its dynamic point removal rate is only 19.73%, resulting in a composite score of just 32.67. This indicates that while it rarely mis-removes static objects, it struggles to effectively remove dynamic objects in the orchard scene. Removert achieved a static point retention rate of 89.96% and a dynamic point removal rate of

50.93%, with a composite score of 65.04. While its overall performance improved compared to the urban road scenario, it still suffers from insufficient dynamic point removal. OctoMap achieves a dynamic point removal rate of 82.60%, indicating strong capabilities in identifying and removing dynamic points; however, its static point retention rate is 82.70%, suggesting significant mis-removal of static structures—such as tree structures—which can easily lead to the sparsification of the orchard's main structure. In contrast, the method proposed in this paper achieves static point retention, dynamic point removal, and a composite score of 93.94%, 76.47%, and 84.31, respectively, and also obtains the highest composite score. As shown in Figures 3, the method proposed in this paper effectively removes dynamic objects from the scene while preserving the continuity of the orchard's main trunk structure and inter-row contours, indicating that the proposed probabilistic multi-resolution fusion strategy is better suited for orchard scenes.

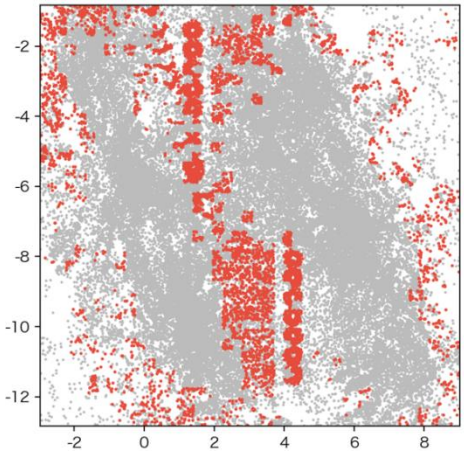

Figure 3(1) REASOR Visualization of Dynamic Point Removal

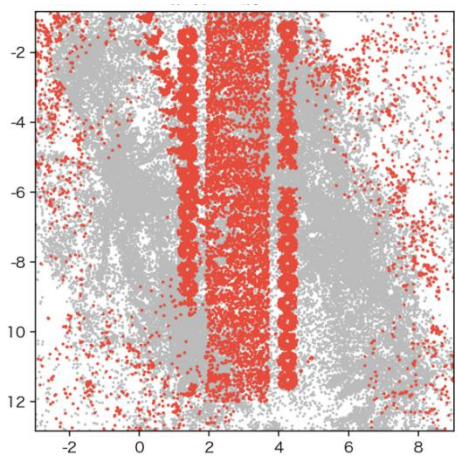

Figure 3(2) PaperFusion Visualization of Dynamic Point Removal

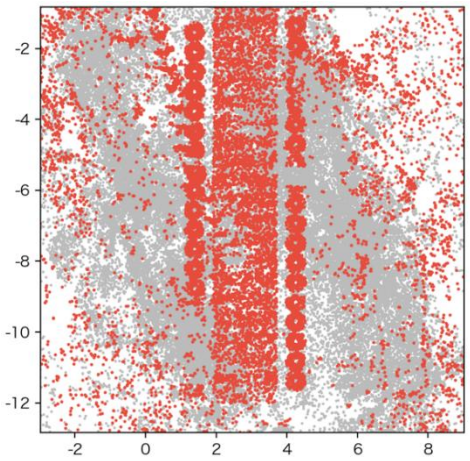

Figure 3(3) Removert Visualization of Dynamic Point Removal

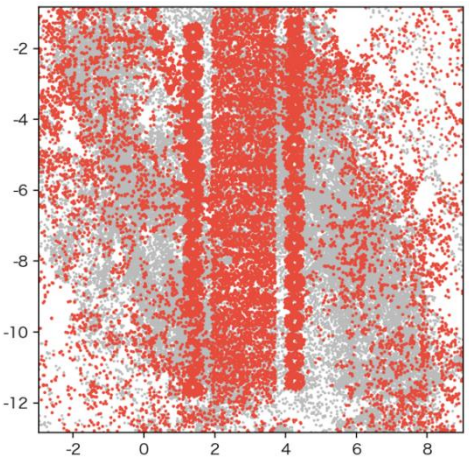

Figure 3(4) OctoMap Visualization of Dynamic Point Removal

Figure 3 TreeScope Series: Visualization of Dynamic Node Removal Using 4 Algorithms

Table 3 TreeScope: Point Statistics Results

Sequence	Method	PR (%)	RR (%)	score
TreeScope	Removert	89.96	50.93	65.04
TreeScope	OctoMap	82.70	82.60	82.65
TreeScope	ERASOR	94.96	19.73	32.67
TreeScope	PaperFusion	93.94	76.47	84.31

A comparison of the results from the two sets of experiments shows that, although our method did not consistently achieve the highest scores on individual metrics, it obtained the highest overall scores on both datasets, indicating that it strikes a better balance between static point retention and dynamic point removal. In the SemanticKITTI scenario, our method achieved overall score improvements of 29.18, 21.81, and 0.73 compared to Removert, ERASOR, and OctoMap,

respectively; in the TreeScope scenario, the improvements were 19.27, 51.64, and 1.66, respectively. These results indicate that the probability-based decision mechanism based on log-odds evidence accumulation, combined with a multi-resolution fusion strategy, can effectively mitigate the trade-off between dynamic point omission and static point mis-removal in traditional methods, thereby enhancing the stability and robustness of static map construction in dynamic environments.

3. CONCLUSION

To address the issue of laser point clouds in dynamic environments being easily disrupted by moving objects, which leads to dynamic ghosting in static maps, we propose a dynamic environment-aware method based on a probabilistic multi-resolution framework, building upon the traditional Removert method. This method is based on the estimation of two-state posterior probabilities for spatial elements. It employs a log-odds form to perform recursive updates on multi-frame observation evidence, extending the traditional binary classification approach to a continuous confidence expression, thereby enhancing the ability to describe uncertain information during dynamic point recognition. Additionally, by integrating multi-resolution voxel representations, the method fuses observation evidence across different scales, balancing the ability to capture local details with the stability of global classification.

To mitigate the issue of redundant accumulation caused by the correlation of multi-scale evidence, we further designed anti-overaccumulation mechanisms such as intra-frame fusion, incremental pruning, and support for evidence saturation and forgetting factors. Additionally, a dual-threshold strategy is adopted to output static, dynamic, and unstable states, thereby enhancing the method's robustness under conditions of repeated observations. Through these designs, the proposed method can better balance the relationship between static structure preservation and dynamic object removal during the dynamic point removal process.

In the experimental validation section, the proposed method was compared with methods such as Removert, ERASOR, and OctoMap on the SemanticKITTI road scene and the TreeScope orchard scene. The results indicate that although the proposed method is not optimal in individual metrics, it achieves the highest overall scores on both datasets. This demonstrates that the method achieves a better balance between static point retention and dynamic point removal, exhibiting good environmental adaptability and stability. In summary, the dynamic environment perception method proposed in this chapter provides a cleaner and more reliable environmental representation for subsequent path planning, laying the perceptual foundation for autonomous navigation of robots in dynamic environments.

Acknowledge

This work is supported by Natural Science Major Project of the Higher Education Institutions of Jiangsu Province of China under Grant No. 24KJA520008, 23KJA520014, Key Research and Development Program of Yancheng City (Social Development) under Grant No. YCBE202554, Applied Basic Research Program of Yancheng City under Grant No. YCBK2025076, Jiangsu Key R&D Program (Industry Foresight and Key Core Technology) Key Project, BE2021016, Key Technology Research and Development of Precise Intelligent Picking Robot System for Typical Fruit and Tea under Complex Environment.

REFERENCES

1. Dou, H. J., Chen, Z. Y., Zhai, C. Y., et al. (2024). Recent advances in autonomous navigation technology for intelligent orchard machinery. *Transactions of the Chinese Society of Agricultural Machinery, 55*(4), 1–22.
2. Placed, J. A., Strader, J., Carrillo, H., et al. (2023). A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Transactions on Robotics, 39*(3), 1686–1705.
3. Zhu, H., Kuang, X., Su, T., et al. (2022). Dual-constraint registration LiDAR SLAM based on grid maps enhancement in off-road environment. *Remote Sensing, 14*(22), 5705.
4. Kim, G., & Kim, A. (2020). Remove, then revert: Static point cloud map construction using multiresolution range images. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 10758–10765). IEEE.
5. Fan, T., Shen, B., Chen, H., et al. (2022). DynamicFilter: An online dynamic objects removal framework for highly dynamic environments. In *2022 International Conference on Robotics and Automation (ICRA)* (pp. 7988–7994). IEEE.
6. Pomerleau, F., Krüsi, P., Colas, F., et al. (2014). Long-term 3D map maintenance in dynamic environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3712–3719). IEEE.
7. Chen, X., Li, S., Mersch, B., et al. (2021). Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data. *IEEE Robotics and Automation Letters, 6*(4), 6529–6536.
8. Milioto, A., Vizzo, I., Behley, J., et al. (2019). RangeNet++: Fast and accurate LiDAR semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4213–4220). IEEE.
9. Cortinhal, T., Tzelepis, G., & Aksoy, E. E. (2020). SalsaNext: Fast, uncertainty-aware semantic segmentation of LiDAR point clouds. In *International Symposium on Visual Computing* (pp. 207–222). Springer.

10. Li, S., Chen, X., Liu, Y., et al. (2021). Multi-scale interaction for real-time LiDAR data segmentation on an embedded platform. *IEEE Robotics and Automation Letters, 7*(2), 738–745.
11. Ren, Y., Cai, Y., Zhu, F., et al. (2024). ROG-Map: An efficient robocentric occupancy grid map for large-scene and high-resolution LiDAR-based motion planning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 8119–8125). IEEE.
12. Li, Y., Na, L., Liang, X., et al. (2025). Indoor dynamic environment mapping based on semantic fusion and hierarchical filtering. *ISPRS International Journal of Geo-Information, 14*(7), 236.
13. Lim, H., Hwang, S., & Myung, H. (2021). ERASOR: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3D point cloud map building. *IEEE Robotics and Automation Letters, 6*(2), 2272–2279.
14. Hornung, A., Wurm, K. M., Bennewitz, M., et al. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots, 34*(3), 189–206.

CITATION

Tang, H., Zhou, C., Li, Y., Zhu, L., & Yang, H. (2026). A Method for Dynamic Environmental Awareness Based on a Probabilistic Multi-Resolution Framework. In Global Journal of Research in Engineering & Computer Sciences (Vol. 6, Number 3, pp. 1–10). https://doi.org/10.5281/zenodo.20311843