IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

BeautyMap: Binary-Encoded Adaptable Ground Matrix for Dynamic Points Removal in Global Maps

Mingkai Jia*[1], Qingwen Zhang*[2], Bowen Yang[1], Jin Wu[1], Ming Liu[1], Patric Jensfelt[2]

***Abstract*— Global point clouds that correctly represent the static environment features can facilitate accurate localization and robust path planning. However, dynamic objects introduce undesired *'ghost'* tracks that are mixed up with the static environment. Existing dynamic removal methods normally fail to balance the performance in computational efficiency and accuracy. In response, we present *'BeautyMap'* to efficiently remove the dynamic points while retaining static features for high-fidelity global maps. Our approach utilizes a binary-encoded matrix to efficiently extract the environment features. With a bit-wise comparison between matrices of each frame and the corresponding map region, we can extract potential dynamic regions. Then we use coarse to fine hierarchical segmentation of the z-axis to handle terrain variations. The final static restoration module accounts for the range-visibility of each single scan and protects static points out of sight. Comparative experiments underscore BeautyMap's superior performance in both accuracy and efficiency against other dynamic points removal methods. The code is open-sourced at https://github.com/HKUSTGZ-IADC/BeautyMap.**

I. Introduction

Point clouds are widely used in various robotics applications due to their effectiveness in supporting essential functions and tasks [1], [2], [3], [4]. Existing Simultaneous Localization and Mapping (SLAM) approaches [5], [6] simply fuse point clouds without intentionally removing dynamic points that may affect the accurate representation of static environment features. Including dynamic points in the map can result in ghost points, as illustrated in red in Fig. 1. These ghost points will also affect the performance of downstream tasks. In the localization task, ghost points may reduce robustness. For global path planning, the presence of ghost points can lead to suboptimal path selection.

Existing methods of dynamic point removal can be grouped into three categories based on their logic of pipeline: freespace-based, differential-based, and data-driven. Freespace-based methods [7], [8], [9] identify free regions if they are frequently passed through during ray-casting. However, the partial detection problem [10] and the high computation demands limit their applications. Differential-based approaches [11], [12] classify dynamic points by comparing differences between scans and maps. However, their lack of a suitable global map representation complicates the restoration of mistakenly removed static points. Data-driven methods [13], [14], [15], [16], [17] rely on learning from dataset and train a feature extraction network to detect dynamic points directly. Although these methods effectively identify moving objects, domain adaptation is a big challenge if the train and test datasets are from different sensor setups.

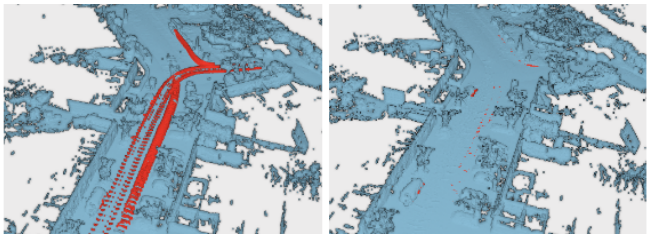


Fig. 1: The results of dynamic points removal in SemanticKITTI dataset, utilizing our proposed approach called BeautyMap. **Red:** all of the dynamic points (left) and after removed (right). **Blue:** static points.

In this paper, we introduce our novel dynamic points removal method for open-world environments, referred to as the Binary-Encoded Adaptable groUnd maTrix for dYnamic points removal (BeautyMap). The proposed method can remove dynamic points from a global point cloud map efficiently and accurately. To enhance the efficiency, we represent point cloud data as 3D binary grids and encode them to 2D matrices. We mark all potential dynamic points by comparing matrices between frames and the global point cloud map using bitwise operations. This straightforward operation allows us to perform potential dynamic region detection more efficiently than traditional ray-casting methods.

For accurate dynamic points removal, we perform an adaptable ground adjustment process. In this process, we apply a coarse ground extraction module to filter outlier points below ground and a hierarchical resolution module to segment the upper surface of the ground and the dynamic points near the ground. Then we perform a static points restoration process to review the former potential dynamic regions, where we consider the sensor property and mask out of sight detections. We evaluate our method through several datasets and BeautyMap achieves state-of-art performance on the dynamic removal in point cloud maps benchmark [10]. Our approach is open-source at https://github.com/HKUSTGZ-IADC/BeautyMap.

Our main contributions are as follows:

- We represent a binary 3D grid map as a binary-encoded matrix to improve computational efficiency through bit manipulation and matrix operations.

M. Jia and Q. Zhang are co-first authors.

[1]Authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. Ming Liu is the corresponding author (e-mail: eelium@ust.hk).

[2]Authors are with the Division of Robotics, Perception, and Learning(RPL), KTH Royal Institute of Technology, Stockholm 114 28, Sweden.

Preprint submitted to IEEE Robotics and Automation Letters as Submission for RA-L only
Received March 29, 2024 02:51:42 Pacific Time

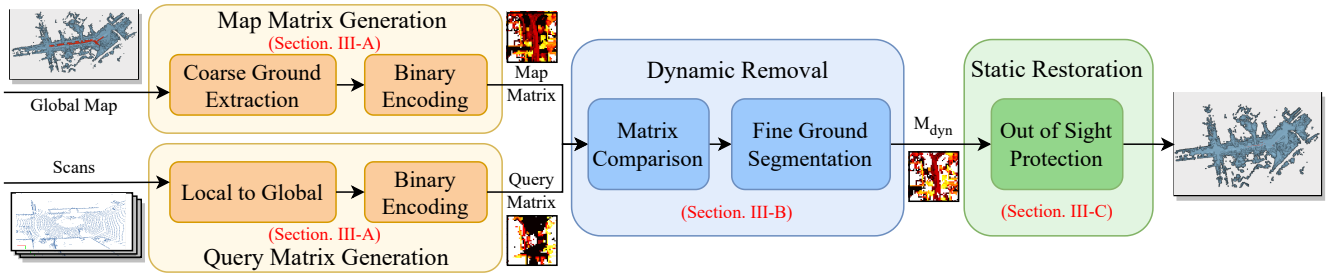


Fig. 2: The overview pipeline of our proposed method. The global map is encoded as a matrix, and processed by the adaptable ground adjustment. Scans are similarly encoded after being transformed into the global frame. To realize dynamic region recognition, the query matrix is compared with the corresponding map submatrix bitwise to get a potential dynamic matrix. Finally, incorrectly classified points that are actually static are restored.

- We introduce adaptive adjustment and hierarchical z-axis resolution for ground-level points and present a protection module for falsely detected points.
- We validate BeautyMap against state-of-the-art methods and demonstrate its superior performance through a comprehensive evaluation.

II. Related Works

This section reviews dynamic point removal methods, categorized into three types: freespace-based, differential-based, and data-driven. We primarily focus on the map representation structures of these methods and their different approaches to achieving map fidelity and efficiency.

A. *FreeSpace-based Removal Algorithms*

Most freespace-based removal algorithms utilize occupancy voxel maps. Various data structures and their extensions aim to compress memory and enhance efficiency for a faster traverse of all voxels. Octomap [7] uses ray casting, marking voxels with points in them as occupied, voxels along the ray as free, and the rest remains unknown. Its Octree structure offers hierarchical resolution by equally dividing a cube into eight segments. Duberg *et al.* [9], [18] expand the structure by explicitly storing the voxel status and introducing faster ray-casting techniques. Schmid *et al.* [19] adopt Voxblox [8], using the Truncated Signed Distance Function (TSDF) to construct the Euclidian Signed Distance Fields (ESDF). These methods focus on free spaces but struggle with spaces where occupancy is uncertain due to the sparseness of rays, especially when used with low-cost sensors. While these algorithms attempt to extend the coverage of free cells by leveraging inflating and clustering, certain obstacle points remain unaddressed in the global map.

B. *Differential-based Removal Algorithms*

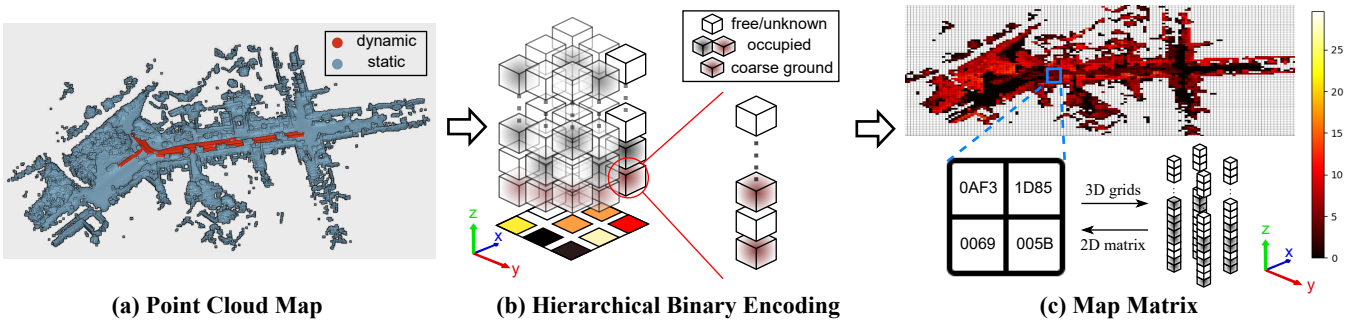
Differential-based removal algorithms, unlike freespace-based methods, detect dynamic points based on visibility, reducing computational overhead compared to ray-casting methods. Pomerleau *et al.* [20] use spherical coordinates for sparse point clouds and determine dynamic elements by dynamic probabilities of associated pairs of points. Kim *et al.* [11] use a range image-based approach to separate static and dynamic points in urban environments. Others, like Lim *et al.* [12], introduce height differences in volumes to incorporate cells and voxels. Recently, ERASOR2 [21] explore instant-aware mapping methods. Although these methods improve computational efficiency in recognizing dynamic points, their map representation structures are ego-centric, lacking global consistency, and necessitating additional transformations from global map points to the local frame. Different from these methods, we perform difference detection in consistent global coordinates. Furthermore, we leverage binary encoding to enable fast matrix operations for the difference calculations.

C. *Data-driven Methods*

Mersch *et al.* [22] employs sparse 4D convolutions to segment receding moving objects in 3D LiDAR data, efficiently processing spatiotemporal information using sparse convolutions. Sun *et al.* [23] develop a novel framework for fusing spatial and temporal data from LiDAR sensors, leveraging range and residual images as input to the network. Pfreundschuh *et al.* [13] design an unsupervised network using an end-to-end occupancy grid-based approach for dynamic object labeling. Milioto *et al.* [14] utilize range images combined with a neural network to address challenges in SqueezeNet [24], such as discretized displacement and blurred output. MotionSC [15] develop a real-time dense local semantic mapping algorithm leveraging 3D deep learning for dynamic object supervision. These methods face challenges due to the types of labels and training models, which limit the accurate perception of only classified objects. Scene flow methods [25], [17], [26], [27] provide a possible solution to estimate the velocity of points in a class-agnostic way. However, Domain adaption is still a crucial problem which means that the accuracy of methods decreased a lot if the train and test datasets were collected by different sensors [28].

III. Methodology

The BeautyMap framework is illustrated in Fig. 2. After turning point clouds into binary-encoded matrices (Section III-A), we mark potential dynamic regions by comparing matrices directly and use fine ground segmentation to extract



(a) Point Cloud Map **(b) Hierarchical Binary Encoding** **(c) Map Matrix**

Fig. 3: Implementation of our proposed map representation structure. (a) Global cluttered point cloud map. The red refers to dynamic points from ground truth only for visualization and no ground truth label will be used. (b) Map storage structure. Colored voxels mean they are occupied, and free voxels in white. Hierarchical resolutions on z-axis are applied to the ground grid cells shown in brown. (c) The binary-encoded 2D map matrix. The 2D matrix values of the zoom-in region are shown in hexadecimal, with the view of vertical occupancy data on the right. For a better view, each value is logarithmically mapped and visualized in heatmap format.

dynamic parts near the ground (Section III-B). We then employ static restoration to protect points out of sight from being removed (Section III-C). We run our method on each scan independently and remove detected dynamic points in the end. Further details are provided in the following.

A. Binary-encoded Matrix

1) Matrix Generation: The map representation approach is crucial for the dynamic points removal performance. Recent studies [11], [19], [29] have extensively employed range image maps to enhance the computational efficiency of classic voxel-based ray-casting methods. However, their operation within the ego-centric frame lacks global consistency, which means a large number of global map points need to be converted into local frames. Our method uses consistent voxel-based structures for both scans and the global map to enhance efficiency, so that all processes can be performed within the same global frame.

As shown in Fig. 3, our model retains a conventional grid structure. To facilitate efficient storage and retrieval, we compress the information into a 2D matrix, where each element is a binary number that contains the encoded vertical occupancy data. This structure necessitates only one-time processing of the global point cloud map, making the matrix immediately usable without any need for conversions.

To prepare the data within a unified coordinate system for matrix indexing, we apply a transformation to the point cloud, followed by compression into a matrix $\mathbf{M} \in \mathbb{R}^{N_1 \times N_2}$. This reduces the complexity of subsequent operations to the order of the matrix.

We then encode the occupancy status of each vertical column of voxels bit-wise as a binary number. We define the status of the vertical cell as 1 for *occupied* and 0 for others and perform bit operations based on the index subsets. For instance, if a specific matrix element has points in its third upward cell, the cell occupancy value will be set as 1. Any occupancy state beyond the maximum storable digits exceeding the height limit is considered part of the occupancy of the highest cell. The operation is also performed on each scan frame to obtain the query matrix within the unified coordinate system.

Algorithm 1 Binary-Encoding Algorithm

Initialization: For each i, j, set $\mathbf{M}_{i,j} \leftarrow 0$.
Set bit constraint N_{bit}
for each point $\mathbf{p} = \{p_x, p_y, p_z\}$ in pts **do**
  $i \leftarrow \text{floor}(p_x/\text{res}_g)$
  $j \leftarrow \text{floor}(p_y/\text{res}_g)$
  $k \leftarrow \text{floor}(p_z/\text{res}_h)$
  if $k < 0$ **then**
    Skip to the next iteration.
  else if $k \geq N_{bit}$ **then**
    $\mathbf{M}_{i,j} \leftarrow \mathbf{M}_{i,j} \vee (1 \ll (N_{bit} - 1))$
  else
    $\mathbf{M}_{i,j} \leftarrow \mathbf{M}_{i,j} \vee (1 \ll k)$
  end if
end for
Output: binary matrix $\mathbf{M}$

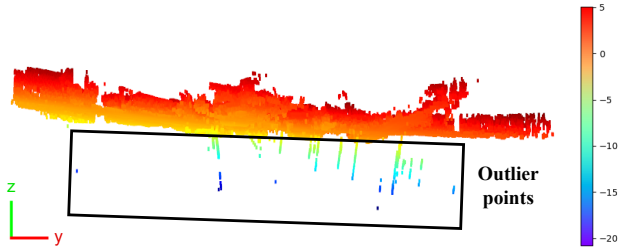


Fig. 4: Underground outlier points in the black square are apparently shown in the side-view of the global point cloud map. The color bar indicates the point heights in meters.

2) Coarse Ground Extraction: During range sensor scanning, surfaces with high reflectivity, such as glass or water, can cause rays to have multiple reflections. This often leads to dense outlier point clusters below ground in certain regions as illustrated in the side-view in Fig. 4. These outliers

will expand the boundaries and result in inefficient space utilization for traditional voxel structures and also affect the accuracy of the representation structure.

To identify outlier points below ground and perform adaptable ground extraction, we employ the Median Absolute Deviation (MAD) algorithm [30]. Superior in representing the central tendency and less influenced by outliers compared to other metrics, the MAD algorithm effectively identifies and handles outlier points in this situation. With the assumption that ground height changes are continuous, we determine ground height constraints by calculating the MAD value of the lowest points in a close range. Let $\mathcal{L}$ be the set of lowest point height values from the surroundings, then the process is represented by the following equations:

$$\mathrm{MAD}(\mathcal{L}) = \mathrm{median}\left(|l_i - \mathrm{median}(\mathcal{L})|\right) \ \forall l_i \in \mathcal{L} \quad (1)$$

$$\{l_{\min}, l_{\max}\} = \mathrm{median}(\mathcal{L}) \pm 3 \times \mathrm{MAD}(\mathcal{L}) \quad (2)$$

$$g_k = \min(\max(l_k, l_{\min}), l_{\max}) \quad (3)$$

where l_k is the lowest height value of the current vertical column, and l_i is an individual height value from the set $\mathcal{L}$. $l_{\min}, l_{\max}$ is the lower and upper bound respectively. We confine the ground height g_k of the coarse ground cell k to lie between $l_{\min}$ and $l_{\max}$. When encoding occupancy status into a 3D grid, g_k serves as the starting height of the lowest cell in each vertical column. As shown in Fig. 3(b) and Fig. 5(a), the 3d grid starts from different height cells, thereby better handling non-flat terrains. For the global map, we introduce a hierarchical structure only for coarse ground cells. To achieve the fine resolution structure, we encode points inside the ground cell into a 3D binary grid again as shown in Fig. 3 (b).

B. Dynamic Removal

In this part, we first detect dynamic part radically by directly comparing map and query matrices from the previous steps. Differences are marked as potential dynamic regions. Then, using the ground points in the static grid, we zoom in on the grid and use fine ground segmentation to further mark the missed dynamic points (see Fig. 5).

1) Matrix Comparison: Based on our binary-encoded matrix-based map representation, we identify a matrix $\mathbf{M}_{dyn}$ indicating potential dynamic regions by calculating a difference matrix $\mathbf{M}_{diff}$ between each query scan matrix $\mathbf{M}_{scan}$ and the map $\mathbf{M}_{map}$ as following:

$$\mathbf{M}_{dyn} = \mathbf{M}_{map} \wedge (\neg \mathbf{M}_{scan}), \quad (4)$$

where $\wedge$ denotes the AND operation and $\neg$ denotes the NOT operation.

2) Fine Ground Segmentation: Existing methods [12], [10], [31] utilize Principal Component Analysis or Sample Consensus segmentation to perform ground plane fitting, or estimate the normal direction to segment the ground plane. However, they still struggle with the considerable thickness of ground points in the global map which is caused by pose errors and severe terrain variations. The fixed resolution of these methods also makes it difficult to separate dynamic points close to the ground points distinctly.

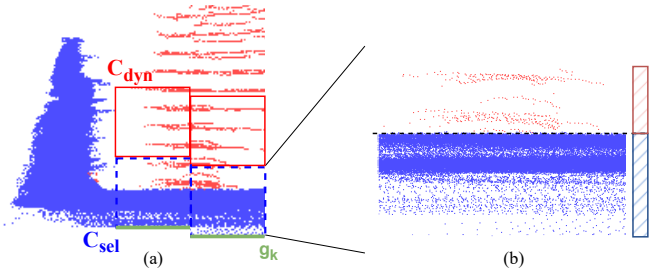


Fig. 5: Our fine ground segmentation process. The black box zooms in on C_{sel} showing the ground points number ratio segmentation along the black dashed line. Blue points for static and red points for dynamic. The red boxes are potential dynamic grids and the dashed blue boxes with both static and dynamic points are for internal calculation.

In our approach, as detailed in Section III-A.2, we first define g_k to identify coarse ground grid cells. These cells are initially determined based on their potential to be part of the ground. Since the cell with both dynamic points and the ground might be mistakenly detected as static, we introduce the concept of C_{sel}. A coarse ground grid cell is considered as C_{sel} only if the adjacent cell above it is potentially dynamic.

When building a global map, dynamic objects are moving, and ground points also accumulate through time. This results in different distribution of point numbers of the ground and dynamic points. Through this fact, we calculate the distribution of points in the coarse ground cell at higher resolution and apply a ground points number ratio (Fig. 5 (b)) to segment the ground that contains the most number of points.

C. Static Restoration Module

Despite the ground segmentation, many static regions out of sight of the current scan are falsely included in $\mathbf{M}_{dyn}$ from Eq. (4). Therefore, we implement further restoration strategies to balance dynamic point removal and static point retention. To restore these static points in the global map, we employ range and visibility-based methods according to the properties of sensors, producing mask matrices that refine our classification of dynamic regions. The results are shown in Fig. 6.

Some points in the map may be out of sight in the current scan's view while inside the potential dynamic $\mathbf{M}_{dyn}$. Considering the LiDARs' field of view, we restore cells above the highest detected cells of each vertical column. The slope k of the rays with the highest point is defined as:

$$k = \max_i \left(\frac{z_i}{\sqrt{x_i^2 + y_i^2}}\right), \quad (5)$$

where i is the index of a point and $\mathbf{p}_i = (x_i, y_i, z_i)$ is the point's position. And the highest height of each column can be calculated in the global frame by $\lceil k \times s \rceil$, where s is the 2D distance to the sensor. We then protect the higher regions from being classified as dynamic since they are outside the sight range.

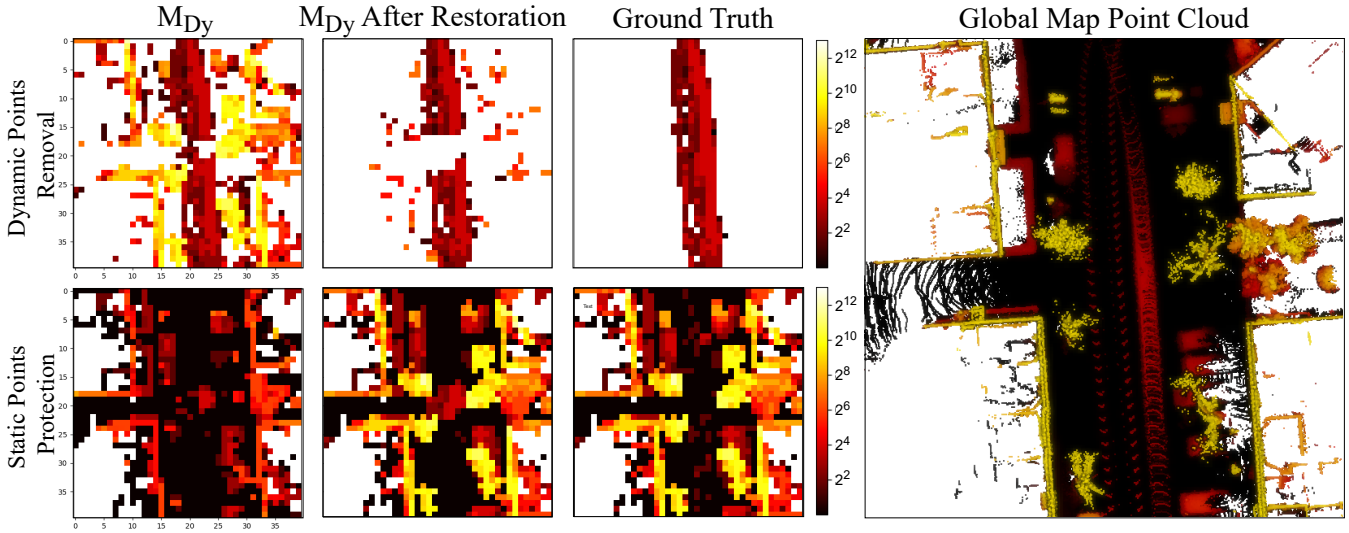


Fig. 6: Visulization of matrices with potential dynamic regions. The first row shows the potential dynamic point region matrix, the dynamic region matrix after performing our static restoration method which has a noticeable decrease in the mistakenly identified static regions, and the ground truth. The second row displays the remaining static region matrix, the matrix after static restoration, and the ground truth. The original point cloud of the global map is shown on the right.

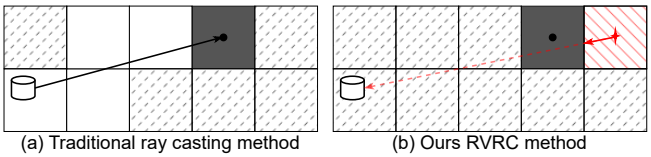


Fig. 7: Comparison of traditional RC and ours RVRC. Gray dashes refer to unknown spaces, white shows the free spaces, black solid is the occupied cell, and red slash represents our potential dynamic cell.

In dynamic removal tasks, ray casting means that the ray is projected from the sensor to detected points and helps identify free spaces along its path. Considering higher efficiency and effectiveness, we propose reverse virtual ray casting (RVRC). It casts virtual rays from each potential dynamic region back to the sensor. This reverse casting ends when meeting the first occupied cell along the virtual ray paths, indicating that this region is blocked and the potential dynamic cell is protected from being removed. Using grid-based ray casting methods, similar to accelerated methods [9], we ensure a balance between speed and accuracy. We make it more efficient by only considering newly detected potential dynamic grids in subsequent scans within the global frame as shown in Fig. 7.

IV. Experiment

We follow evaluation protocol from [10], the experiments dataset of different sensor types includes KITTI [32] (VLP-64) where dynamic ground truths are from Semantic-Kitti [33] and semi-indoor [10] datasets (VLP-16). We kept the same setting from [10], [12] that selected partial frames from the KITTI sequences 00, 01, 02, and 05 which present the most dynamics. All poses of datasets are from SLAM packages like SuMa [34] for KITTI and simple-ndt for semi-indoor [35]. The metrics of dynamic removal in maps are Static Accuracy (SA), and Dynamic Accuracy (DA) at the point level without downsampling the ground truth map to have an accurate and fair evaluation. Instead of using Associate Accuracy (AA) proposed in [10], we propose to use the Harmonic Accuracy (HA) as a comprehensive metric that combines both accuracies.

$$HA = \frac{2 \times SA \times DA}{SA + DA}$$

Beside methods provided in the public benchmark [10], we add three more data-driven methods: 4DMOS [22], MapMOS [16] and DeFlow [17] into our comparison. The training dataset of 4DMOS and MapMOS includes KITTI sequences from 0 to 10. No results are therefore presented for the KITTI sequences as these methods have seen ground truth data. DeFlow is trained on the Argoverse 2 *Sensor* dataset. The default parameters for BeautyMap are 1×1 m^2 cell size for all KITTI sequences and 0.5×0.5 m^2 cell size for the semi-indoor dataset. Table III shows an ablation study on the tradeoff between different cell sizes, performance, and running speed.

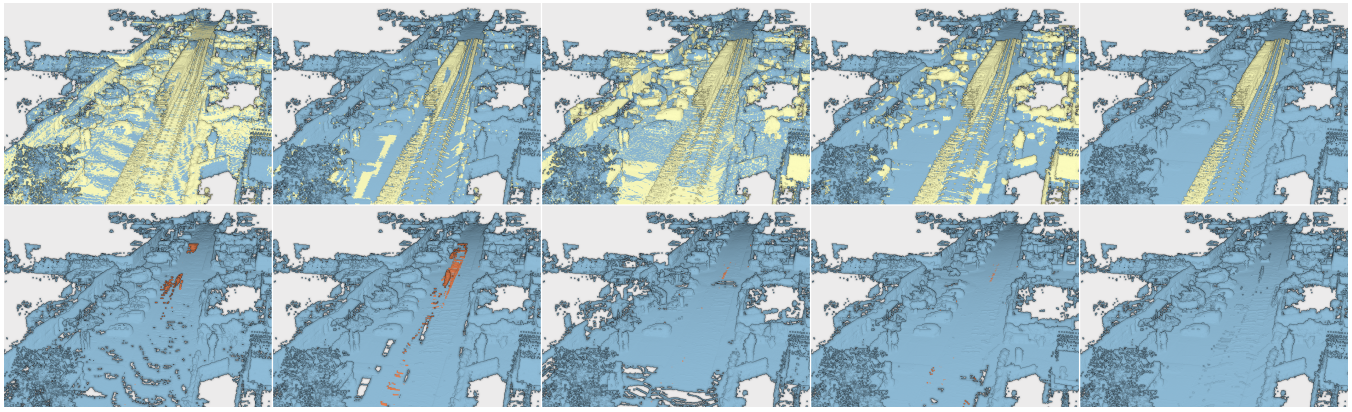
The computation cost is also compared between our binary-encoded structure-based method and others. All experiments are conducted on a desktop computer equipped with an Intel Core i9-12900KF processor.

A. Quantitative Results

The quantitative comparison results of dynamic points removal methods on several datasets are shown in Table I. Regarding differential-based methods, ERASOR shows its ability to remove dynamic points and achieve a high DA score, at the cost of reducing the SA score. Removert shows

TABLE I: Quantitative comparison of dynamic points removal methods, † means data-driven methods. The best results are shown in **bold** and the second best results are shown in underlined. Results are in percentage.

	KITTI sequence 00			KITTI sequence 01			KITTI sequence 05			Semi-indoor		
Methods	SA ↑	DA ↑	HA ↑	SA ↑	DA ↑	HA ↑	SA ↑	DA ↑	HA ↑	SA↑	DA ↑	HA ↑
4DMOS† [22]	-	-	-	-	-	-	-	-	-	99.99	10.60	27.59
MapMOS† [16]	-	-	-	-	-	-	-	-	-	99.99	4.75	9.07
DeFlow† [17]	99.43	81.68	89.69	99.19	81.25	89.33	99.48	50.85	67.30	99.99	2.02	3.95
Removert [11]	99.44	41.53	58.59	97.81	39.56	56.33	99.42	22.28	36.40	99.96	12.15	21.67
ERASOR [12]	66.70	98.54	79.55	98.12	90.94	94.39	69.40	99.06	81.62	94.90	66.26	78.04
Octomap [7]	68.05	99.69	80.89	55.55	99.60	71.28	66.28	99.24	79.48	88.97	82.18	85.44
Octomap w GF [10]	93.06	98.67	95.78	80.64	97.27	88.18	93.54	92.48	93.01	96.79	73.50	83.55
dynablox [19]	96.76	90.68	93.62	96.33	68.01	79.73	97.80	88.68	93.02	98.81	36.49	53.30
BeautyMap (Ours)	96.76	98.38	**97.56**	99.17	92.99	**95.98**	96.34	98.29	**97.31**	93.69	90.67	**92.16**



(a) Octomap [7] (b) dynablox [19] (c) ERASOR [12] (d) BeautyMap (Ours) (e) Ground Truth

Fig. 8: Qualitative results in KITTI Sequence 05 using one VLPC-64 LiDAR sensor. The first row represents the labeling result. The second row shows the cleaned output map from different approaches. In all the methods shown in the images, yellow represents points labeled as dynamic, and orange indicates dynamic points that are incorrectly labeled as static.

an opposite low score, which is mainly attributed to the attendance of incorrectly recovered dynamic points near the ground with the coarse range image, as mentioned in [11]. The dynamic points removal results of the semi-indoor dataset on 4DMOS, MapMOS and DeFlow reveal that these three data-driven methods struggle with 16-channel LiDAR. Since both are trained on datasets captured with LiDARs with many channels like KITTI (64-channel) and AV2 (32-channel), one possible way is to label and train on mixed datasets. The freespace-based method Octomap struggles to balance static point saving and dynamic removal because of noise and ground point as mentioned in [10]. The other two, Octomap w GF and dynablox, usually maintain good static and dynamic accuracy. The special case is on the semi-indoor dataset (16-channel), where 4DMOS, MapMOS, DeFlow, Removert, and dynablox perform poorly on dynamic accuracy. The reason behind it for Removert and dynablox is that they transfer LiDAR points to a range image and the resolution of this range image for LiDARs with few channels limits the detection ability. As a differential-based method, our method benefits from the new point cloud representation and the difference strategy we propose and outperforms all other methods in all four datasets in Table I.

B. Qualitative Results

The qualitative results for dynamic removal in point cloud maps on KITTI sequence 05 are presented in Fig. 8. The first row displays the labeling results, while the second row shows the cleaned output maps, with remaining dynamic points marked in orange. For freespace-based methods, Octomap (Fig. 8 (a)) exhibits a considerable number of false dynamic labels in the first row and misclassified dynamic points in the cleaned map in the second row. Dynablox (Fig. 8 (b)), on the other hand, is more conservative in identifying dynamic grids, as evident from the results in the first row. ERASOR (Fig. 8 (c)) produces a cleaner map compared to the other two methods, but it also detects more false dynamic points and inaccurately removes some static structures. The cleaned map of our BeautyMap most closely matches the ground truth, outperforming the other methods in terms of accuracy and dynamic point removal.

C. Computation Cost Comparison

We compare the runtime per frame of our proposed BeautyMap method with other dynamic point removal methods on KITTI sequence 01. As shown in Table II, our BeautyMap method outperforms the others, performing a runtime of 0.046 seconds per point cloud. This efficiency is

TABLE II: Runtime comparison of different methods running on KITTI sequence 01.

Methods	Runtime/point cloud [s] ↓
Removert [11]	0.134 ± 0.004
ERASOR [12]	0.718 ± 0.039
Octomap [7]	2.981 ± 0.952
Octomap w GF [10]	2.147 ± 0.468
Dynablox [19]	0.141 ± 0.022
BeautyMap (Python)	**0.046** ± 0.011

attributed to our binary-encoded matrix structure. The second fastest method Removert [11] utilizes range image and direct comparison between the map and scans to realize a shorter runtime. The Dynablox [19] has its runtime close to that of the Removert method, with the TSDF structure to identify the free spaces and decrease the complexity. The ERASOR method [12] follows the rank by encoding the point clouds into Volume of Interest (VOI) and performing bin-wise calculations based on this structure. The Octomap [7] and Octomap w GF [10], using traditional Octree structure and ray casting method for each point, are the slowest.

In exploring the tradeoffs between cell size, speed, and performance in our BeautyMap method, we present a comparison of different cell sizes in Table III using KITTI sequence 02. As shown in Table III, a smaller cell size (0.5 meters) leads to a slight increase in HA, the improvement is marginal compared to the significant increase in runtime. On the other hand, a larger cell size (2.0 meters) results in higher DA but lower SA, as it tends to remove more points. This default setting also offers the best runtime efficiency and well dynamic points removal performance. Comparing the results for BeutyMap in Table II and Table III with 1.0 meter cell size it can be seen that the runtime is consistently low in both datasets.

TABLE III: Runtime comparison of BeautyMap with different cell size on KITTI sequence 02. Runtime is on per point cloud.

Size [m]	SA ↑	DA ↑	HA ↑	Runtime/point cloud [s]
0.5	83.92	84.14	**84.03**	0.132 ± 0.034
1.0	83.40	82.41	82.90	0.031 ± 0.039
2.0	74.92	88.83	81.28	**0.018** ± **0.010**

D. Ablation Study

In our proposed method, matrix comparison is the main module to classify potential dynamic points. But other modules are equally important to improve the performance. Therefore, we carry out an ablation study on KITTI sequence 01 to further evaluate the contributions of adaptable ground adjustment modules and static restoration.

When performing matrix comparison without any other modules, our method achieves a relatively high DA score but a low SA score as shown in Table IV. This kind of map results in losing many static features and may harm localization accuracy.

TABLE IV: Ablation Study of Function Modules

Ground Module	Static Restoration	SA ↑	DA ↑	HA ↑
		59.55	96.47	75.79
✓		59.04	**99.01**	76.46
	✓	**99.38**	77.81	87.94
✓	✓	99.17	92.99	**96.03**

The ground module further includes dynamic points that are close to the ground proposed in Section III-B.2. We achieve higher dynamic accuracy with the ground module and static accuracy is still similar to the previous one, proving that points we remove mostly belong to dynamic objects.

However, both of them have relatively low static accuracy. That's where static restoration can help with protecting the out of sight points from being removed. With static restoration, we observe that SA is higher around 40% than the other two and near 100%.

The complete BeautyMap method with ground and static restoration modules together can achieve state-of-the-art results in dynamic point removal in a map. More than 90% points in the map are correctly classified and output a satisfied cleaned map with HA to 96%.

V. Conclusions

In this paper, we introduce 'BeautyMap', a novel methodology for dynamic points removal in global point cloud maps. Our approach leverages the binary-encoded matrix map representation structure, together with adaptable ground adjustment and static restoration modules, to become a considerable solution for global map maintenance.

Through comparative experiments, BeautyMap reveals its superior capabilities in both accuracy and efficiency against other traditional methods, showing the efficiency of our structure and methodology. It can also be generalized for diverse scenarios for dynamic points removal and is suitable for range sensors of various types.

Like many other methods, BeautyMap also faces the challenge of dealing with multiple ground levels at a specific position. A potential solution for our proposed method is to detect the ceiling and perform better protection.

In future works, we will explore further usage of the binary-encoded matrix structure, deal with challenging scenes like multiple ground levels, implement an online mode integrated with real-time odometry, and expand our methodology for long-term change detection and map update frameworks.

Acknowledgement

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

This work was supported by Guangdong Basic and Applied Basic Research Foundation (No. 2021B1515120032),

Guangzhou-HKUST(GZ) Joint Funding Program (No. 2024A03J0618), awarded to Prof. Ming Liu.

REFERENCES

[1] J. Jiao, H. Wei, T. Hu, X. Hu *et al.*, "Fusionportable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3851–3856.

[2] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu *et al.*, "Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint," *The International Journal of Robotics Research*, vol. 41, no. 3, pp. 270–280, 2022.

[3] T. Liu, Q. hai Liao, L. Gan, F. Ma *et al.*, "The role of the hercules autonomous vehicle during the covid-19 pandemic: An autonomous logistic vehicle for contactless goods transportation," *IEEE Robotics & Automation Magazine*, vol. 28, no. 1, pp. 48–58, 2021.

[4] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten *et al.*, "Elevation mapping for locomotion and navigation using gpu," 2022.

[5] T.-M. Nguyen, D. Duberg, P. Jensfelt, S. Yuan *et al.*, "Slict: Multi-input multi-scale surfel-based lidar-inertial continuous-time odometry and mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2102–2109, 2023.

[6] W. Xu, Y. Cai, D. He, J. Lin *et al.*, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

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

[8] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart *et al.*, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[9] 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.

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

[11] 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)*. IEEE, 2020, pp. 10 758–10 765.

[12] 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.

[13] P. Pfreundschuh, H. F. Hendrikx, V. Reijgwart, R. Dubé *et al.*, "Dynamic object aware lidar slam based on automatic generation of training data," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 641–11 647.

[14] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2019, pp. 4213–4220.

[15] J. Wilson, J. Song, Y. Fu, A. Zhang *et al.*, "Motionsc: Data set and network for real-time semantic mapping in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8439–8446, 2022.

[16] B. Mersch, T. Guadagnino, X. Chen, I. Vizzo *et al.*, "Building Volumetric Beliefs for Dynamic Environments Exploiting Map-Based Moving Object Segmentation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 8, pp. 5180–5187, 2023.

[17] Q. Zhang, Y. Yang, H. Fang, R. Geng *et al.*, "Deflow: Decoder of scene flow network in autonomous driving," *International Conference on Robotics and Automation (ICRA)*, 2024.

[18] D. Daniel, Q. Zhang, M. Jia, and P. Jensfelt, "Dufomap: Efficient dynamic awareness mapping," *arXiv preprint arXiv:2403.01449*, 2024.

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

[20] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale *et al.*, "Long-term 3d map maintenance in dynamic environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3712–3719.

[21] H. Lim, L. Nunes, B. Mersch, X. Chen *et al.*, "Erasor2: Instance-aware robust 3d mapping of the static world in dynamic scenes," in *Robotics: Science and Systems (RSS 2023)*. IEEE, 2023.

[22] B. Mersch, X. Chen, I. Vizzo, L. Nunes *et al.*, "Receding Moving Object Segmentation in 3D LiDAR Data Using Sparse 4D Convolutions," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 3, pp. 7503–7510, 2022.

[23] J. Sun, Y. Dai, X. Zhang, J. Xu *et al.*, "Efficient spatial-temporal information fusion for lidar-based 3d moving object segmentation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 456–11 463.

[24] B. Wu, X. Zhou, S. Zhao, X. Yue *et al.*, "Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 4376–4382.

[25] P. Jund, C. Sweeney, N. Abdo, Z. Chen *et al.*, "Scalable scene flow from point clouds in the real world," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1589–1596, 2021.

[26] K. Vedder, N. Peri, N. Chodosh, I. Khatri *et al.*, "Zeroflow: Fast zero label scene flow via distillation," *arXiv preprint arXiv:2305.10424*, 2023.

[27] X. Li, J. Zheng, F. Ferroni, J. K. Pontes *et al.*, "Fast neural scene flow," *arXiv preprint arXiv:2304.09121*, 2023.

[28] M. K. Wozniak, V. Kårefjärd, M. Hansson, M. Thiel *et al.*, "Applying 3d object detection from self-driving cars to mobile robots: A survey and experiments," in *2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2023, pp. 3–9.

[29] H. Wu, Y. Li, W. Xu, F. Kong *et al.*, "Moving event detection from lidar point streams," *Nature Communications*, vol. 15, no. 1, p. 345, 2024.

[30] P. J. Rousseeuw and C. Croux, "Alternatives to the median absolute deviation," *Journal of the American Statistical association*, vol. 88, no. 424, pp. 1273–1283, 1993.

[31] Z. Yan, X. Wu, Z. Jian, B. Lan *et al.*, "Rh-map: Online map construction framework of dynamic object removal based on 3d region-wise hash map structure," *IEEE Robotics and Automation Letters*, 2023.

[32] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[33] J. Behley, M. Garbade, A. Milioto, J. Quenzel *et al.*, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9297–9307.

[34] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments," in *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[35] Z. Qingwen and J. Mingkai, "Simple ndt slam," https://github.com/Kin-Zhang/simple_ndt_slam.git, 2022.