Moving Object Segmentation in Point Cloud Data using Hidden Markov Models

Vedant Bhandari, Jasmin James, Tyson Phillips and P. Ross McAree

***Abstract*— Autonomous agents require the capability to identify dynamic objects in their environment for safe planning and navigation. Incomplete and erroneous dynamic detections jeopardize the agent's ability to accomplish its task. Dynamic detection is a challenging problem due to the numerous sources of uncertainty inherent in the problem's inputs and the wide variety of applications, which often lead to use-case-tailored solutions. We propose a robust learning-free approach to segment moving objects in point cloud data. The foundation of the approach lies in modelling each voxel using a hidden Markov model (HMM), and probabilistically integrating beliefs into a map using an HMM filter. The proposed approach is tested on benchmark datasets and consistently performs better than or as well as state-of-the-art methods with strong generalized performance across sensor characteristics and environments. The approach is open-sourced at `https://github.com/vb44/HMM-MOS`.**

I. INTRODUCTION

Detecting motion in the workspace is a crucial capability for autonomous agents. Agents employ sensors such as cameras and Light Detection and Ranging (LiDAR) to image their environment. The Moving Object Segmentation (MOS) problem involves categorizing the pixels in an image or the points in a LiDAR scan as static or dynamic. A key challenge is to provide consistent detection across environments, platform dynamics, and sensor characteristics. There is a need for a solution that offers generalized and accurate dynamic detection. To address this, we propose a learning-free MOS approach demonstrating strong generalized performance.

II. RELATED WORK

Learning-free approaches to solving the MOS problem are generally categorized as scan-based or map-based.

Scan-based methods compare successive observations to highlight discrepancies in the environment. Underwood et al. [1] detect changes in scans by identifying discrepancies in the observed space, with points labelled dynamic if they are greater than a distance from previously registered points. Yoon et al. [2] use a similar idea with dynamic detection relying on a window size that allows sufficient displacement of the object - a characteristic differing between object classes. Mersch et al. [3] demonstrate state-of-the-art performance with 4DMOS using sparse 4D spatio-temporal convolutions for segmenting dynamic points in scan-to-scan comparisons.

Map-based methods construct a representation of the environment and query changes in occupancy. Octomap by Armin et al. [4] clamp the occupancy probabilities to evolve beliefs in dynamic environments. Methods using similar forgetting policies described by Yguel et al. [5] cannot adapt to different object classes without compromising the mapping quality and introducing false positives. Dynablox by Schmid et al. [6] integrates temporal properties in a Truncated Signed Distance Field map, demonstrating generalized performance across diverse dynamic objects. Mersch et al. [7] extend 4DMOS with a volumetric approach to retain a memory of spaces that can be occupied by moving objects, increasing the detection rate.

The University of Queensland, 4072, Australia `v.bhandari@uq.edu.au`

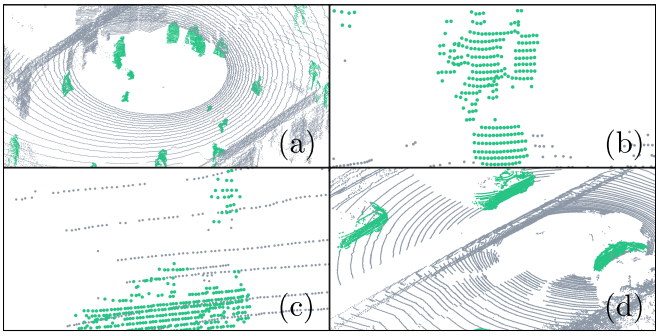

Fig. 1: HMM-MOS accurately detects moving objects using the same configuration in all scenarios, including (a) a shopping centre, (b) a person jumping over a moving ball, (c) a pedestrian walking alongside a car, and (d) multiple cars on a highway.

III. PROPOSED APPROACH

The general HMM framework applied to identify dynamic objects in point cloud data is adapted from [8]. This section provides a brief overview of the three stages of the algorithm.

1) Voxel Representation: A map frame, $\mathcal{M}$, is defined to indicate the environment's origin. At time k, the map, $M_{\mathcal{M},k}$, is discretized using voxels, v, of a user-configured size Δ. The voxels are augmented with temporal attributes. Without uncertainty, detecting dynamic objects is as simple as updating voxel occupancy with new observations, with occupancy changes suggesting dynamic objects. As the state of each voxel is not directly interpretable due to the associated uncertainty, an HMM is used to represent each voxel's occupancy, similar to [9], [10]. Using the notation from [11], each voxel is represented using an HMM with three states ($n = 3$), $S = \{unobserved, occupied, free\}$. Let the i-th voxel's state vector, $\hat{\mathbf{x}}_{i,k} \in \mathbb{R}^{n\times 1}$, denote the probability of being in each state at time k, with the initial state is given by $\hat{\mathbf{x}}_{i,0} = [1, 0, 0]^{\mathrm{T}}$. The state transition matrix, $\mathbf{A} \in \mathbb{R}^{n\times n}$, has large self-transition probabilities for each state, based on the belief that a voxel requires sufficient confidence before transitioning. The likelihood of the i-th voxel being in each state at time k given the sensor observation is encoded in the measurement conditional densities, $\mathbf{B}_{i,k} \in \mathbb{R}^{n\times n}$. Once defined, a voxel's state is efficiently updated using the recur-

Accepted to the IEEE IROS 2024 workshop on Long-Term Perception for Autonomy in Dynamic Human-shared Environments: What Do Robots Need?

sive HMM filter [12], $\hat{\mathbf{x}}_{i,k} = \eta_{i,k}\mathbf{B}_{i,k}\mathbf{A}\hat{\mathbf{x}}_{i,k-1}$, where $\eta_{i,k}$ is a normalization that ensures $\hat{\mathbf{x}}_{i,k}$ is a probability. Voxels outside the sensor's maximum range and those unobserved in the global window, w_g, are removed from the map.

2) Map Update: A point cloud at time k in the sensor frame, $P_{\mathcal{S},k}$, is transformed by the current sensor pose estimate, $\hat{T}_{\mathcal{M}\to\mathcal{S},k}$, to locate the scan points in the map frame, $P_{\mathcal{M},k}$. The scan in map frame, $P_{\mathcal{M},k}$, is discretized at a voxel resolution of Δ, to form a voxelized scan, $P^{'}_{\mathcal{M},k}$. A raycast is performed using [13] to find all observed voxels. All observed voxels are saved in $P^{'obs}_{\mathcal{M},k}$.

The measurement conditional densities of the i-th voxel being in a particular state given an observation is, $\mathbf{B}_{i,k} = \text{diag}(0, \mathcal{L}^o_{v_i}, 1 - \mathcal{L}^o_{v_i})$, where $\mathcal{L}^o_{v_i}$, is the likelihood of the voxel being occupied. An observed voxel is likely to be occupied if it is close to a voxel in the voxelized scan, $P^{'}_{\mathcal{M},k}$, and free otherwise. This is captured using the scan's Euclidean Distance Field [14]. The EDF value for the i-th observed voxel, d_i, is used to calculate the occupancy likelihood by evaluating an unnormalized Gaussian at d_i, $\mathcal{L}^o_{v_i} = \exp\left(-d_i^2/2\sigma_o^2\right)$, where σ_o is a user-configured standard deviation to capture uncertainty in the estimate. A voxel's state is updated when the state's probability surpasses a predefined threshold, p_{min}.

3) Dynamic Point Identification: A voxel's occupancy transition seeds the detection of dynamic objects. The first step is to identify voxels from the current voxelized scan, $P^{'}_{\mathcal{M},k}$, that changed state in the voxel map, $M_{\mathcal{M},k}$, captured in $P^{'chg}_{\mathcal{M},k}$. The change detection allows for likely dynamic voxels to be identified, however, changes in the voxel's neighbourhood are not examined. A spatiotemproal (4D) convolution is performed to identify missed detections and suppress noisy detections. For each voxel, $v_i \in P^{'}_{\mathcal{M},k}$, the likelihood of being dynamic, $\mathcal{L}^{dyn}_{v_i}$, is calculated by summing state changes in the voxel's local neighbourhood over a local window size of w_l. A kernel, $K_m \in \mathbb{R}^{m\times m\times m}$, is convolved with each voxel in $P^{'}_{\mathcal{M},(k-w_l)\to k}$ to compute $\mathcal{L}^{dyn}_{v_i}$.

A voxel's dynamic likelihood depends on the voxel size, the convolution kernel, the scan sparsity, and the object. Hence, manually thresholding to extract dynamic voxels based on their likelihood is challenging. Otsu's automatic thresholding [15] is applied to the convolution scores to extract the set of dynamic voxels, $P^{'dyn}_{\mathcal{M},k}$. High-confidence dynamic voxels from the previous scan are preserved in the current scan and saved in a temporal dynamic occupancy map of size w_d scans to assist with future detections. A nearest neighbour dilation is applied to $P^{'dyn}_{\mathcal{M},k}$ to grow the dynamic detection results into neighbouring regions.

IV. RESULTS

We evaluate the proposed algorithm using the DOALS [16], Sipailou Campus [17], and HeLiMOS [18] benchmark datasets to test generalized performance. All tests use the same configuration with an uncertainty equal to the voxel size, $\sigma_o = \Delta$, $p_{min} = 0.99$, a convolution kernel size of $m = 5$, $otsu_{min} = 3$, and scan windows of $w_l = 3$, $w_d = 100$, and $w_g = 300$ to demonstrate generalized behaviour. The full testing conditions, sensor poses, sample videos, and all results are linked on our open-source page. All tests estimate sensor pose using [19], unless stated otherwise.

The results displayed in Tables I-III demonstrate the strong generalization capabilities of the proposed algorithm in comparison to state-of-the-art methods. The DOALS dataset is recorded with a handheld OS1-64 in environments with diverse dynamic objects predominantly consisting of pedestrians, the Sipailou Campus dataset is recorded using a Livox Avia mounted to an unmanned ground vehicle as it traverses a university campus, whereas the new HeLiMOS dataset is recorded with four different LiDARs mounted to a vehicle in dynamic urban environments. We demonstrate consistent performance, on par with or performing better than state-of-the-art such as Dynablox [6], 4DMOS [3] and MapMOS [7]. To evaluate generalized performance, we compare the HeLiMOS benchmark results with the methods trained on the SemanticKitti dataset. When the benchmark approaches are trained on the new data, they outperform the proposed approach, see [18]. The algorithm's performance metrics are hindered by the ground truth labelling process as we detect movement only, and not if the object has moved throughout the scan sequence. This severely decreases the recall. The proposed algorithm is computationally expensive and only provides real-time results within a 20-50m range depending on the point cloud density. There is ongoing work to achieve real-time results for larger detection ranges.

TABLE I: Evaluation on the DOALS dataset with best results in bold. Results for other methods are as documented by [6].

Method	ST	SV	HG	ND
4DMOS [3]	38.8	50.6	71.1	40.2
LMNet [20] (Refit)	19.9	18.9	27.4	40.1
Dynablox [6]	**86.2**	**83.2**	84.1	**81.6**
This paper (online), $\Delta = 0.20m$	82.7	80.8	**85.9**	81.4
LC Free Space [21] (20 m)	48.7	31.9	24.7	17.7
Dynablox [6] (20 m)	87.3	**87.8**	86.0	83.1
This paper (online), $\Delta = 0.20m$ (20 m)	**88.9**	84.7	**87.3**	**83.5**

TABLE II: Evaluation on the Sipailou Campus dataset with best results in bold. Results for other methods are as reported by [17].

Method	IoU Validation	IoU Test
MotionSeg3D [22]	6.83	6.72
4DMOS [3]	78.54	82.30
Motion-BEV-h [17]	70.94	71.51
This paper (online), $\Delta = 0.25m$	**85.60**	**87.00**

TABLE III: Evaluation on the HeLiMOS dataset with best results in bold. Results for other methods are as documented by [18].

Method	L	A	O	V	Avg
4DMOS, online [3]	52.1	54.0	64.2	4.7	43.7
4DMOS, delayed [3]	59.0	58.3	70.4	5.4	48.3
MapMOS, Scan [7]	58.9	63.2	81.4	4.3	52.0
MapMOS, Volume [7]	**62.7**	66.6	**82.9**	5.8	54.5
This paper (online), $\Delta = 0.25m$	51.3	69.8	75.0	35.0	57.8
This paper, delayed, $\Delta = 0.25m$	57.6	**70.0**	73.4	**53.9**	**63.7**

V. CONCLUSIONS

This paper presents a learning-free solution to the MOS problem. The significance of the work is that it is robust and generalizes to a range of datasets without reconfiguration, producing comparable or better results than state-of-the-art.

Accepted to the IEEE IROS 2024 workshop on Long-Term Perception for Autonomy in Dynamic Human-shared Environments: What Do Robots Need?

REFERENCES

[1] J. P. Underwood, D. Gillsjö, T. Bailey, and V. Vlaskine, "Explicit 3d change detection using ray-tracing in spherical coordinates," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4735–4741.

[2] D. Yoon, T. Tang, and T. Barfoot, "Mapless online detection of dynamic objects in 3d lidar," in *2019 16th Conference on Computer and Robot Vision (CRV)*, 2019, pp. 113–120.

[3] B. Mersch, X. Chen, I. Vizzo, L. Nunes, J. Behley, and C. Stachniss, "Receding moving object segmentation in 3d lidar data using sparse 4d convolutions," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7503–7510, 2022.

[4] 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.

[5] M. Yguel, O. Aycard, and C. Laugier, *Update Policy of Dense Maps: Efficient Algorithms and Sparse Representation*. Springer Berlin Heidelberg, 2008, pp. 23–33.

[6] 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.

[7] 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.

[8] J. James, J. J. Ford, and T. L. Molloy, "A framework for bayesian quickest change detection in general dependent stochastic processes," *IEEE Control Systems Letters*, vol. 8, pp. 790–795, 2024.

[9] D. Meyer-Delius, M. Beinhofer, and W. Burgard, "Occupancy grid models for robot mapping in changing environments," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 26, no. 1, pp. 2024–2030, Sep. 2012. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/8377

[10] Z. Wang, R. Ambrus, P. Jensfelt, and J. Folkesson, "Modeling motion patterns of dynamic objects by iohmm," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1832–1838.

[11] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[12] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov models: estimation and control*. Springer Science & Business Media, 2008, vol. 29.

[13] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[14] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart, "Signed distance fields: A natural representation for both mapping and planning," in *RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics*. University of Michigan, 2016.

[15] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

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

[17] B. Zhou, J. Xie, Y. Pan, J. Wu, and C. Lu, "Motionbev: Attention-aware online lidar moving object segmentation with bird's eye view based appearance and motion features," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8074–8081, 2023.

[18] H. Lim, S. Jang, B. Mersch, J. Behley, H. Myung, and C. Stachniss, "Helimos: A dataset for moving object segmentation in 3d point clouds from heterogeneous lidar sensors," *arXiv preprint arXiv:2408.06328*, 2024.

[19] V. Bhandari, T. G. Phillips, and P. R. McAree, "Minimal configuration point cloud odometry and mapping," *The International Journal of Robotics Research*, vol. 0, no. 0, p. 02783649241235325, 0.

[20] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss, "Moving object segmentation in 3d lidar data: A learning-based approach exploiting sequential data," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6529–6536, 2021.

[21] J. Modayil and B. Kuipers, "The initial development of object knowledge by a learning robot," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 879–890, 2008.

[22] 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.