DUFOMap: Efficient Dynamic Awareness Mapping

Daniel Duberg*[1], Qingwen Zhang*[1], MingKai Jia[2], and Patric Jensfelt[1]

***Abstract*— The dynamic nature of the real world is one of the main challenges in robotics. The first step in dealing with it is to detect which parts of the world are dynamic. A typical benchmark task is to create a map that contains only the static part of the world to support, for example, localization and planning. Current solutions are often applied in post-processing, where parameter tuning allows the user to adjust the setting for a specific dataset. In this paper, we propose DUFOMap, a novel dynamic awareness mapping framework designed for efficient online processing. Despite having the same parameter settings for all scenarios, it performs better or is on par with state-of-the-art methods. Ray casting is utilized to identify and classify fully observed empty regions. Since these regions have been observed empty, it follows that anything inside them at another time must be dynamic. Evaluation is carried out in various scenarios, including outdoor environments in KITTI and Argoverse 2, open areas on the KTH campus, and with different sensor types. DUFOMap outperforms the state of the art in terms of accuracy and computational efficiency. (See https://kth-rpl.github.io/dufomap for more details.)**

I. Introduction

Point clouds are widely used in robotics and other domains like surveying and architecture. Many core robotics components assume a static environment, but when this assumption is violated, it can lead to issues in path planning, mapping, and localization. Dynamic awareness is crucial for robust operation. An example from surveying illustrating the problems caused by dynamic objects is shown in Fig. 1. A point cloud model of a built environment, created using a 3D laser scanner (Leica-RTC360) and often used as ground truth for SLAM [1], [2], [3], can be severely compromised by moving people, as seen in the top right of the figure.

In this work, we propose DUFOMap, a dynamic awareness method based on UFOMap [4]. The framework accumulates point clouds into voxel maps. During integration, ray casting is used to identify the so-called *void* regions that at some time were empty. The classification of dynamic points can then be done by looking for points that fall into these void regions. Special care is required to account for localization errors and sensor noise. We present extensive experimental validation across multiple datasets, sensors, and scenarios, showing the generality, computational efficiency, and broad usability of DUFOMap. Our approach is open-source at https://github.com/KTH-RPL/dufomap. The main contributions of our work:

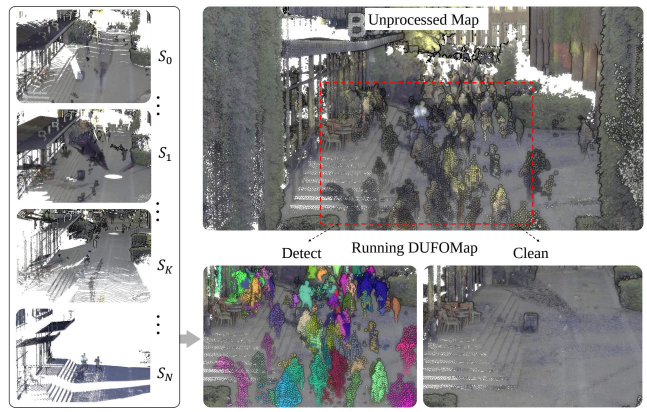


Fig. 1: The mapping pipeline integrates all point clouds into a global map, which initially contains numerous dynamic points. The unprocessed map is shown in the upper right. After processing with DUFOMap, the algorithm effectively detects and removes dynamic points, resulting in a clean and refined map suitable for downstream tasks.

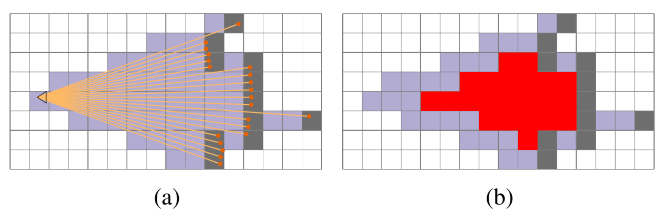


Fig. 2: Example of point cloud integration. (a) Ray casting is performed for each point from the sensor position, the triangle to the left. All cells intersecting a ray are marked as intersected (purple), and the cells where a point falls within are marked as hit (gray). (b) Cells that are intersected and surrounded exclusively by other intersected or hit cells are classified as void regions (red).

- We propose a method for detecting dynamics by finding parts of space that has been observed as free taking into account sensor noise and localization errors.
- Our method achieves state-of-the-art performance in both offline and online scenarios across different scenarios and sensors.
- We demonstrate that our method generalizes in experiments on datasets with five different sensors using the same setting for the method's three parameters.

II. Method

The proposed method, DUFOMap discretizes the world into voxels, each containing a flag i_{void} indicating whether it

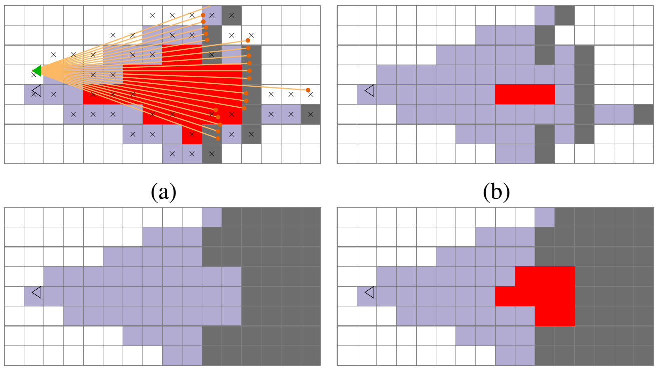
*: Daniel Duberg and Qingwen Zhang are co-first authors. Corresponding author: Qingwen Zhang. (qingwen@kth.se)

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

[2]Author is with Robotics Institute, Hong Kong University of Science and Technology, Hong Kong SAR, China.

TABLE I: Quantitative comparison of dynamic points removal in point cloud maps [5]. The best results are shown in **bold** and the second best results are shown in underlined. Results are in percentage.

	KITTI small town (00)			KITTI highway (01)			Argoverse 2 big city			Semi-indoor		
Methods	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑
Removert [6]	99.44	41.53	64.26	97.81	39.56	62.20	98.97	31.16	55.53	99.96	12.15	34.85
ERASOR [7]	66.70	98.54	81.07	98.12	90.94	<u>94.46</u>	77.51	99.18	87.68	94.90	66.26	79.30
OctoMap [8]	68.05	99.69	82.37	55.55	99.59	74.38	69.04	97.50	82.04	88.97	82.18	<u>85.51</u>
Dynablox [9]	96.76	90.68	93.67	96.33	68.01	80.94	96.08	92.87	**94.46**	98.81	36.49	60.05
DUFOMap (Ours)	97.96	98.72	**98.34**	98.09	94.20	**96.12**	96.67	88.90	<u>92.70</u>	99.64	83.00	**90.94**



(a) (b)

(c) (d)

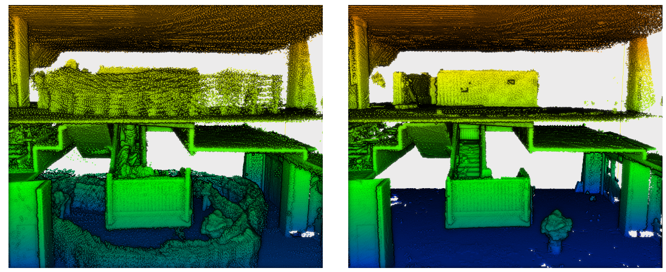
Fig. 3: Example of integration with larger localization errors. (a) Point cloud with the real sensor position (green) offset one cell up compared to Fig. 2(a), showing that some cells are now incorrectly classified (x). (b) We account for a localization error of up to two voxels in any direction. This leads to a more conservative classification of void regions compared to Fig. 2(b). (c) Extending the hits away from the sensor to allow classification of void regions next to obstacles (shown in (d)).

has been observed as empty. The key insight is that *points observed in previously empty regions must be dynamic*.

The DUFOMap inputs pairs of sensor poses and clouds and then continuously updates the void region.

1) Ray-casting from sensor positions to points in the scan.
2) Classifying voxels as hit, intersected, or unknown.
3) Identifying void voxels by confirming all 26 surrounding voxels (or 8 in 2D) are also intersected or hit.

In the real world, sensor noise and localization errors become a problem. If the sensor pose is offset from the true pose, the hits and intersected voxels would also be offset, and result in the set of classified void voxels being incorrect (Fig. 3(a)). We propose to look not only at the direct neighbors of a voxel but also at the surrounding voxels at a Chebyshev distance of d_p away; From Fig. 3(b), it is observed that it is now impossible to classify voxels next to hits as void. To deal with this, anything after a hit is also considered a hit (see Fig. 3(c)). This is implemented by extending the ray casting by inserting hits from where the original ray casting ended. The voxels classified as void after this can be seen in Fig. 3(d). Lastly, we consider the problem of sensor noise and modeled it by marking voxels



(a) Raw (Unclean) Map (b) Cleaned Map

Fig. 4: DUFOMap dynamic points removal performance in a complex, two-floor, structure. The color indicates different heights. For clearer visualization, parts of the walls have been removed.

at a distance d_s ahead of the hit along the ray as hits.

Points are classified as dynamic if they fall into void voxels $i_{\text{void}} = \texttt{true}$), otherwise static. The primary computational effort lies in classifying void regions, which is performed once for each new point cloud. Point classification, in contrast, requires only a quick map lookup and can be executed at any time. This approach allows for the efficient detection of dynamic objects in 3D environments while accounting for sensor and localization uncertainties.

III. Results

Table I shows that DUFOMap achieves the best performance in KITTI [10] and semi-indoor [5] dataset and comparable results in Argoverse 2 [11]. Figure 4 shows a scene from a two-floor building that challenges methods that make assumptions about the height, the ground level, etc. DUFOMap is able to effectively remove dynamic points. The output of our method in survey data is depicted in Fig. 1 (lower right) when applied to the raw data (upper). In the detection results (lower left), we naively clustered the points so that different objects stand out. More quantitative results can be found in our project page https://kth-rpl.github.io/dufomap.

Real-time: We performed a test on a low-power computer (a NUC with an Intel Core i7-8559U)on the semi-indoor dataset. Both Dynablox and DUFOMap reduced the range to 20 m. DUFOMap maintained a frequency of 20 Hz on the 4-core CPU compared with less than 10 Hz in Dynablox.

References

[1] L. Zhang, M. Helmberger, L. F. T. Fu, D. Wisth, M. Camurri, D. Scaramuzza, and M. Fallon, “Hilti-oxford dataset: A millimeter-

accurate benchmark for simultaneous localization and mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 408–415, 2023.

[2] J. Jiao, H. Wei, T. Hu, X. Hu, Y. Zhu, Z. He, J. Wu, J. Yu, X. Xie, H. Huang *et al.*, "Fusionportable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2022, pp. 3851–3856.

[3] N. Thien-Minh, Y. Shenghai, N. Thien Hoang, Y. Pengyu, C. Haozhi, X. Lihua, W. Maciej, J. Patric, T. Marko, Z. Justin, and B. Noel, "Mcd: Diverse large-scale multi-campus dataset for robot perception," in *Conference on Computer Vision and Pattern Recognition*, 11 2024. [Online]. Available: https://mcdviral.github.io/

[4] 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.

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

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

[7] 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.

[8] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.

[9] 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.

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

[11] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.