COB-2025-0946
2D LiDAR SLAM for Dynamic Environments: Real-Time Detection and Filtering of Moving Objects

Sarah Dias
Sabrina Seba
Renan Moreira Pinto
Universidade de São Paulo, Avenida Trabalhador são-carlense, nº 400, Centro. CEP 13566-590 – São Carlos SP.
sarah.dias@usp.br
sabrinaseba18@usp.br
renanmoreira@usp.br
João Carlos Virgolino Soares
Dynamic Legged Systems Lab - Istituto Italiano di Tecnologia, Via S. Quirico, 19d, 16163 Genova GE - Italy
joao.virgolino@iit.it
Marcelo Becker
Universidade de São Paulo, Avenida Trabalhador são-carlense, nº 400, Centro. CEP 13566-590 – São Carlos SP.
becker@sc.usp.br

Abstract. *Autonomous navigation in dynamic environments remains a significant challenge, particularly for systems relying exclusively on 2D LiDAR data. Traditional SLAM algorithms assume static scenes, making them vulnerable to moving objects that can corrupt map consistency and localization accuracy. This paper presents a lightweight and heuristic-based approach that detects, tracks, and filters dynamic objects using only 2D LiDAR scans, without requiring 3D models, visual input, or supervised learning. The method integrates spatial segmentation and frame-to-frame comparison to isolate moving agents, which are then excluded from the scan before mapping. The filtered data is processed by the Hector SLAM algorithm to produce cleaner, more stable maps. Experiments in a simulated environment with multiple moving obstacles show a notable improvement in map quality and reduced localization error when compared to unfiltered baselines. Quantitative metrics, including Intersection over Union (IoU) and Weighted Root Mean Square Error (RMSE), validate the effectiveness of the approach. The results highlight the system's potential for deployment in real-time robotics applications operating in partially dynamic settings.*

Keywords: *Dynamic SLAM, Moving object detection, Dynamic object filtering*

1. INTRODUCTION

Navigation in dynamic environments remains one of the major challenges for autonomous systems that rely on LiDAR as the primary perception sensor. Most classic SLAM (Simultaneous Localization and Mapping) approaches assume a static scenario, which does not reflect real-world applications such as urban, rural or industrial areas where the presence of mobile objects - like pedestrians, vehicles or collaborative robots - is constant. This incorrect assumption can compromise the consistency in the generated maps, degrading the robots position estimation and path generation and, in critical cases, jeopardizing safe navigation.

Several studies have proposed robust solutions for autonomous navigation using only LiDAR sensors, but still focusing mainly on static surroundings. For instance, Thomas *et al.* (2021) explore self-supervised learning for point cloud segmentation in indoor scenarios, while Iqbal *et al.* (2020) and Affonso *et al.* (2025) propose navigation systems applicated to agriculture and crop phenotyping, all based on the assumption of predictable and still scenes. Although efficient in those contexts, such approaches are not directly applicable to environments with significant dynamic activity.

In parallel, camera-based approaches have gained traction due to the advancement of computer vision techniques, and with that alternatives to integrate semantic information to handle moving objects have emerged. DS-SLAM (Yu *et al.*, 2018) and (Virgolino Soares *et al.*, 2023), for example, combine visual SLAM with semantic segmentation to mask dynamic regions. DynaVINS (Song *et al.*, 2022) applies a similar technique but with visual-inertial sensors, and the review by Peng *et al.* (2024) presents SLAM moving object filtering with LiDAR 3D. Despite the success of those methods, most require visual sensors, stereo depth or supervised segmentation, which complicates their application in scenes with variable lighting or reduced visibility, where LiDAR is more reliable.

This study presents a novel solution based exclusively on 2D LiDAR scan data obtained from a mobile LiDAR sensor operating within a controlled simulation environment populated by multiple moving agents. In contrast to existing methods that depend on complex 3D models or supervised semantic segmentation techniques, the proposed approach utilizes only heuristic analysis of 2D LiDAR scans. By employing a method grounded in spatial segmentation and frame-to-frame comparison, the system effectively eliminates the need for computationally intensive deep learning processes. Consequently, the resulting framework is lightweight and efficient, making it more suitable for real-time applications in mobile robotics. The primary contributions of this work are threefold: (i) it demonstrates the feasibility of reliably detecting and tracking dynamic objects using solely 2D LiDAR data; (ii) it introduces a filtering mechanism for removing tracked dynamic objects; and (iii) it integrates the filtered data into a simultaneous localization and mapping (SLAM) system, thereby enhancing the robustness of autonomous navigation in dynamic environments.

2. METHOD

2.1 Method Overview

The proposed methodology integrates mobile object detection, dynamic obstacle filtering, and SLAM mapping into a unified pipeline tailored for dynamic environments. At its core, the system leverages DATMO framework (Konstantinidis *et al.*, 2020) to identify and track moving objects using consecutive 2D LiDAR scan. Building on this information, a filtering module processes LiDAR scan messages and exclude points corresponding to dynamic obstacles. The resulting filtered scan is then fed into the Hector SLAM algorithm for the map generating. By eliminating dynamic points prior to mapping, the system prevents the inclusion of transient artifacts into the occupancy grid, significantly improving localization accuracy and map quality. Figure 1 illustrates the overall SLAM pipeline with dynamic object filtering.

The system begins with raw LiDAR data acquisition, which is processed in two parallel paths. The first stage applies clustering techniques to organize the point cloud into spatial groupings and uses the Detection and Tracking of Moving Objects (DATMO) framework (Konstantinidis *et al.*, 2020) to monitor their movement over time. Points corresponding to dynamic objects are removed from the scan, resulting in a refined dataset. This filtered scan is then used by the Hector SLAM algorithm to align successive scans and build an occupancy grid without relying on external motion estimation, producing a more stable and accurate map.

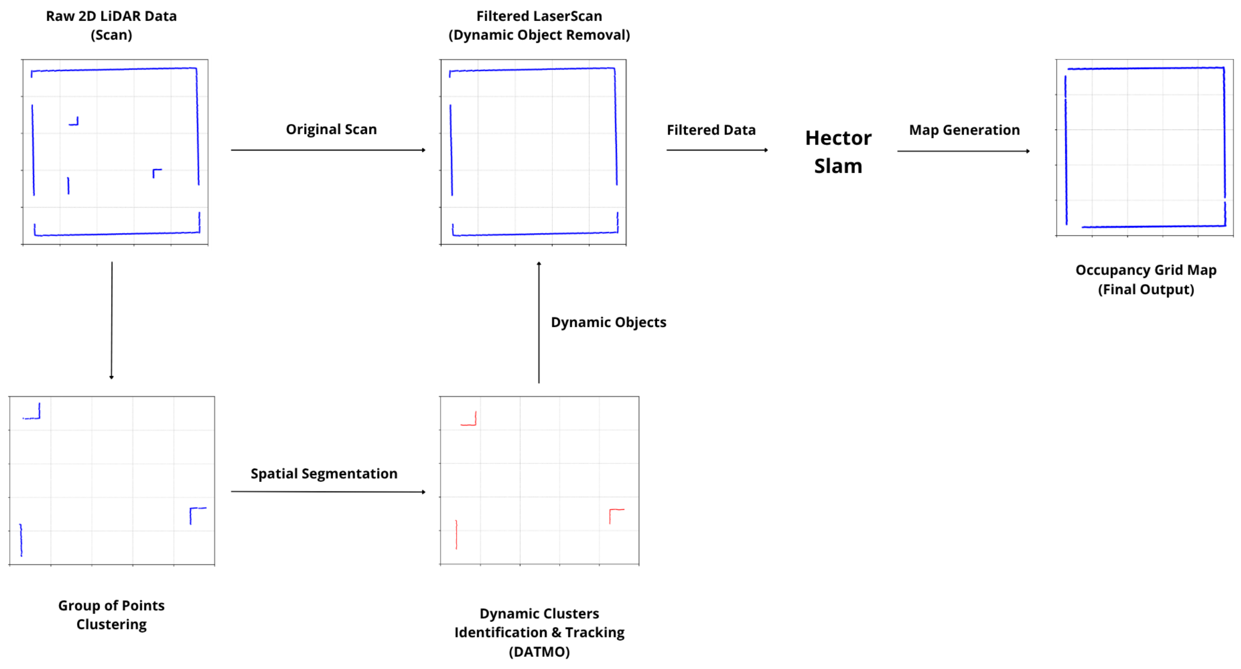


Figure 1: Overview of the proposed SLAM pipeline with dynamic object filtering.

2.2 Object Detection

Mobile object detection plays a crucial role in dynamic SLAM by distinguishing between static elements and moving agents, preventing the latter from compromising the mapping process. To achieve this, the DATMO method was adopted, relying on the comparison of consecutive LiDAR point cloud frames (Konstantinidis *et al.*, 2020). By registering each new frame p_t at time step t with the previous one p_{t-1}, the system detects changes in spatial point distributions. Motion is identified when the displacement $\Delta p = p_t - p_{t-1}$ for a frame exceeds a predefined threshold, suggesting the presence of mobile objects.

To enhance detection robustness, the method performs spatial segmentation of the raw point cloud using Euclidean clustering. Frames p_i and p_j belong to the same cluster if their Euclidean distance satisfies:

$$||p_i - p_j|| < \theta(r) \tag{1}$$

where $\theta(r) = \alpha r + \beta$ is an adaptive threshold that grows with the radial distance r from the sensor, accounting for point cloud sparsity and noise - α and β are constants. This thresholding approach is derived from the Adaptive Breakpoint Detector (Borges and Aldon, 2004), and includes error modeling based on the sensor's range variance to improve clustering accuracy across varying depths.

Following segmentation, each cluster undergoes a geometric approximation step. The algorithm fits rectangles to the clusters by evaluating multiple orientations and selecting the best match based on edge-point alignment. The resulting data is encapsulated in a state vector used in the filtering stage:

$$\mathbf{x} = [p_x,\ p_y,\ \theta,\ l,\ w]^\top \tag{2}$$

where p_x and p_y are the object's position coordinates, θ its orientation, and l and w represent the length and width of the rectangle.

A set of dynamic objects is constructed from this information, representing the successfully detected and tracked entities. Only the elements in this set, denoted by $\mathcal{M}$, are used in the filtering process. To be included, an object must exceed a predefined velocity threshold, ensuring that only dynamic objects are considered for filtering, as defined in Eq. 3.

$$\mathcal{M} = \{\mathcal{O}_i \mid \|\mathbf{v}_i\| > \mathbf{v}_{\text{min}}\} \tag{3}$$

where $\mathcal{M}$ is the set of dynamic objects, $\mathcal{O}_i$ represents an individual object containing the state vector $\mathbf{x}_i$, $\mathbf{v}_i$ is the velocity vector of the object, and $\mathbf{v}_{\text{min}}$ is the minimum velocity threshold used to classify an object as dynamic.

2.3 LiDAR Filter

To reduce interference caused by moving objects during autonomous mapping and navigation, a filtering module was developed as a post-processing stage applied to raw 2D LiDAR data. The proposed method combines low-level laser measurements with spatial and kinematic information from the object detection module to selectively remove reflections caused by dynamic obstacles. This process preserves the static structure of the environment, allowing mapping and localization systems to operate more robustly in partially dynamic scenarios.

The process of filtering dynamic objects, based on the set $\mathcal{M}$ begins with transforming the LiDAR scan into Cartesian coordinates, as shown in Eq. 4:

$$\mathbf{s}_i = (p_{x_i}, p_{y_i}) = (r_i \cos\alpha_i, r_i \sin\alpha_i) \tag{4}$$

where $\mathbf{s}_i$ is the i-th scan measurement in Cartesian coordinates, α_i is the emission angle, and r_i is the corresponding range value.

Prior to the application of the filtering step, the state vector of each dynamic object is employed to construct a transformation matrix that maps coordinates from the sensor frame to the object's local reference frame:

$$T_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i & p_{x_i} \\ -\sin\theta_i & \cos\theta_i & p_{y_i} \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

where T_i denotes the homogeneous transformation matrix corresponding to the dynamic object $\mathcal{O}_i$.

To delineate the exclusion region associated with each dynamic object, the filtering process generates a set of axis-aligned square bounding boxes $\mathcal{B}_i$, each of which corresponds to an object $\mathcal{O}_i \in \mathcal{M}$. The side length of each bounding box is defined by scaling the greater of the object's width w_i and length l_i using a margin factor λ. This results in a half-extent d_i, computed as:

$$d_i = \max(w_i, l_i) \cdot \frac{\lambda}{2} \tag{6}$$

Each bounding box $\mathcal{B}_j$ is centered at the position of its corresponding object and defines a square region in the 2D plane, within which scan points are considered to originate from dynamic elements and are therefore excluded. This region is formally defined in Eq. 7:

$$\mathcal{B}_i = \left\{(p_x, p_y) \in \mathbb{R}^2 \;\middle|\; |p_x| \le d_i,\ |p_y| \le dij\right\} \tag{7}$$

The resulting set of bounding boxes is utilized during the filtering stage to identify and eliminate LiDAR measurements associated with dynamic objects, thereby retaining only the static elements of the environment.

To perform this exclusion, each scan point $\mathbf{s}_i$ is transformed into the local reference frame of every dynamic object $\mathcal{O}_j \in \mathcal{M}$ by applying the inverse of the corresponding transformation matrix T_j^{-1}. A point is discarded if, in any transformed frame, it falls within the spatial extent of the augmented bounding box $\mathcal{B}_j$. This filtering condition is formally defined as:

$$\tilde{\mathbf{s}}_i = \begin{cases} \varnothing, & \text{if } \exists\, \mathcal{O}_j \in \mathcal{M} \text{ such that } T_j^{-1}\mathbf{s}_i \in \mathcal{B}_j \\ \mathbf{s}_i, & \text{otherwise} \end{cases} \tag{8}$$

2.4 SLAM

The filtered point cloud is subsequently processed using Hector SLAM, a 2D mapping and localization algorithm that operates solely on LiDAR data to construct maps and estimate the robot's position within the environment (Kohlbrecher *et al.*, 2011). The algorithm employs a scan matching technique that aligns consecutive laser scans using Gauss-Newton optimization, enabling real-time integration of new data into the existing map. A key advantage of Hector SLAM is that it does not require odometry or inertial measurements, making it particularly suitable for platforms with unreliable or unavailable motion sensing—such as wheeled robots on low-friction surfaces, aerial vehicles, or lightweight mobile systems.

However, despite its computational efficiency and high accuracy in static settings, Hector SLAM is notably sensitive to the presence of dynamic objects. Transient entities—such as pedestrians, vehicles, or other mobile agents—within the sensor's field of view may be erroneously interpreted as static features and integrated into the map. This misclassification results in the formation of ghost artifacts and compromises the accuracy of both the mapping and localization processes. Over time, these errors can accumulate, leading to a gradual degradation in overall map quality.

3. RESULTS

In this section, we present the evaluation of the proposed system through tests. The methodology was validated in a controlled simulation environment to assess the effectiveness of the proposed system in generating consistent maps in scenarios with dynamic obstacles. The results include both qualitative and quantitative analyses.

The simulations were conducted in a Gazebo environment. Three distinct situations were configured to represent different levels of complexity. In all cases, a robot equipped with a LiDAR sensor performed scanning, while making an oscillatory motion. The data was processed to build the map using the Hector SLAM algorithm.

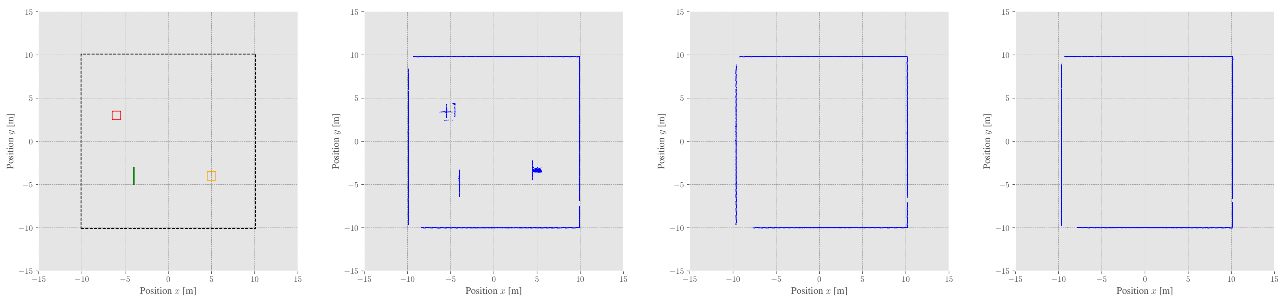
The environment features three simultaneously moving boxes: one follows a smaller square trajectory, another traces a larger square path, and the third moves along an L-shaped route. Two of the boxes have dimensions of $1 \times 1 \times 1$, while the third is elongated, measuring $0.1 \times 2 \times 0.1$.

The testes were structured into three distinct configurations, each progressively increasing in complexity and adaptability. In the first configuration, the real map dimensions are used as a baseline, and an initial result is conducted in which the raw set of 2D LiDAR readings is directly passed to the Hector SLAM system without any filtering. Since this setup includes both static and dynamic elements, the resulting map exhibits considerable distortion. The motion of dynamic objects introduces significant artifacts, manifested as black streaks and irregularities within the occupancy grid, which ultimately compromise the structural consistency and accuracy of the generated map.

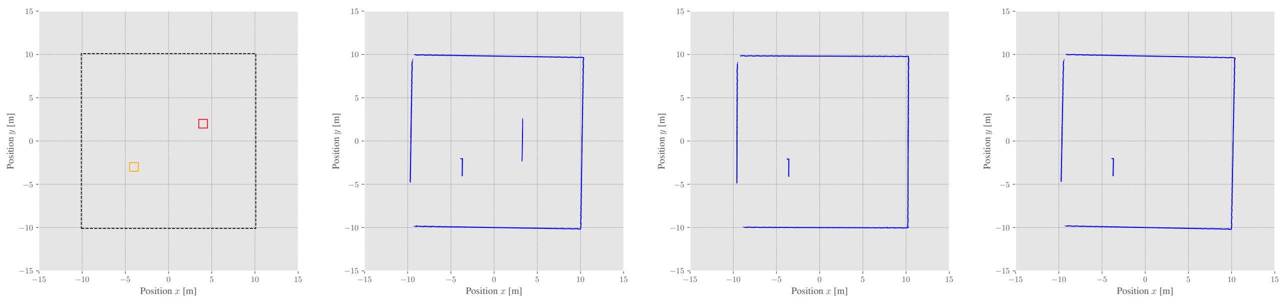
In the second configuration, a filtering module is introduced, incorporating kinematic and spatial criteria to exclude dynamic elements. Filtering is activated only when exactly three moving objects $\mathcal{O}_i \in \mathcal{M}$ are detected, each with a velocity magnitude satisfying $\|\mathbf{v}_i\| > \mathbf{v}_{\min}$, where $\mathbf{v}_{\min} = 0{,}5\,\text{m/s}$. The exclusion region associated with each object $\mathcal{O}_i$ is defined by a square bounding box $\mathcal{B}_i$, computed using a fixed base dimension $d_{\text{ref}} = 1{,}5\,\text{m}$ and a margin factor $\lambda = 2{,}0$, yielding a half-side length of $d_i = \frac{d_{\text{ref}} \cdot \lambda}{2} = 1{,}5\,\text{m}$. Each scan point $\mathbf{s_i}$ is transformed into the local reference frame of the object via T_i^{-1} and discarded if it lies within the exclusion region $\mathcal{B}_i$. Despite the use of fixed parameters, this approach demonstrates robustness in handling minor uncertainties in object dimension estimation, resulting in effective filtering.

In the third configuration, a more adaptive strategy is employed in which the dimensions of the exclusion regions are dynamically computed based on the estimated properties of each detected object. Rather than relying on a constant reference size, the half-side of the exclusion box is calculated as $d_i = \frac{\max(w_i, l_i) \cdot \lambda}{2}$, with $\lambda = 1{,}0$, allowing each bounding box $\mathcal{B}_i$ to scale according to the object's largest dimension (either width w_i or length l_i). This approach preserves the geometric proportions of each obstacle and eliminates the need for external hyperparameters such as d_{ref} or the number of expected objects. The point exclusion mechanism remains unchanged: each scan point $\mathbf{s_i}$ is transformed using T_i^{-1} and excluded if it is located within any of the computed exclusion regions $\mathcal{B}_i$. This adaptive configuration yields improved accuracy in environments with varying obstacle geometries and enhances the generalizability of the filtering process.

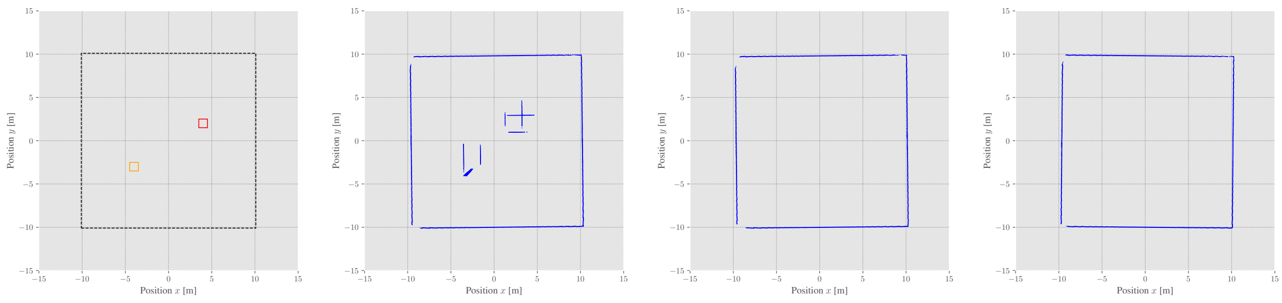
Finally, Fig. 2 summarizes the four maps side by side, including the reference map (ground truth), allowing a direct visual comparison between the different strategies.



(a) Ground truth map outline. (b) Without dynamic object filtering. (c) With dynamic object filtering. (d) With parameter-based filtering.



(e) Ground truth map outline. (f) Without dynamic object filtering. (g) With dynamic object filtering. (h) With parameter-based filtering.



(i) Ground truth map outline. (j) Without dynamic object filtering. (k) With dynamic object filtering. (l) With parameter-based filtering.

Figure 2: Comparison of map outlines across different filtering methods and trials.

Complementing the analysis, Table 1 presents quantitative metrics used to evaluate the maps, including Intersection over Union (IoU) and Weighted Root Mean Square Error (Weighted RMSE).

The IoU is mathematically defined as the ratio between the intersection and the union of two sets. Given a reference set A (e.g., the occupied pixels in the ground truth map) and a predicted set B (the occupied pixels in the generated map), the IoU is calculated as:

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{9}$$

where $|A \cap B|$ represents the number of elements common to both sets (intersection), and $|A \cup B|$ represents the total number of distinct elements present in at least one of the sets (union). The IoU value ranges from 0 to 1, with 1 being the ideal value, indicating a perfect overlap between the generated map and the reference map.

The Weighted RMSE is a quantitative metric that assesses the average deviation between corresponding values, with greater emphasis placed on larger errors through the use of squared differences. It is particularly useful in scenarios where certain regions of a map or measurements carry varying degrees of importance or reliability. The metric is defined as:

$$\text{Weighted RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \sigma_i (x_i - y_i)^2}{\sum_{i=1}^{n} \sigma_i}} \tag{10}$$

where x_i and y_i represent the reference and predicted values, respectively, σ_i denotes the weight associated with each pair of values, and n is the total number of data points considered. The weighting factor σ_i can encode the relative importance of different spatial regions or reflect the confidence level in specific measurements. A lower Weighted RMSE indicates that the predicted values closely align with the ground truth, offering a reliable measure of local accuracy in mapping tasks.

Analyzing the results, the intersection over union metric exhibits variability depending on the map scenario, indicating that the boundaries of the generated maps—both filtered and unfiltered—are generally well-aligned. However, the analysis of the weighted RMSE provides additional insight; the presence of dynamic objects that persist in the unfiltered map leads to higher local discrepancies, resulting in poorer performance. While the overall borders of the maps remain relatively consistent across both methods, the numerical error introduced by residual dynamic objects confirms their negative impact on map quality. This highlights the importance of effective filtering in improving local accuracy and reducing distortion in dynamic environments.

Table 1: Weighted RMSE and IoU for filtered and unfiltered maps

	Unfiltered		**Filtered**	
	IoU	**RMSE**	**IoU**	**RMSE**
1	0,2234	0,0362	0,1503	0,0115
2	0,1133	0,1763	0,1044	0,1675
3	0,1787	0,0303	0,1223	0,0205

4. CONCLUSION

This paper presented a LiDAR-based SLAM approach for dynamic environments, integrating real-time detection, tracking, and filtering of moving objects into the mapping process. Results demonstrated that the proposed system performs effectively across a range of simulated scenarios, including environments with multiple moving objects following varied paths. The comparison between filtered and unfiltered maps confirmed a substantial improvement in mapping quality and robustness. Furthermore, the results show that the filtering module remains effective even when no predefined parameters are provided, reinforcing the adaptability of the system in less constrained and more unpredictable settings.

As future work, the system will be extended into a complete navigation framework. With the current filtering pipeline providing a reliable environmental representation in most scenarios, the next objective is to leverage this map for safe and efficient global path planning. By incorporating motion prediction for tracked dynamic objects, a local planning strategy can be employed to adjust the global path for obstacle avoidance. This integration would enable the system to support autonomous navigation in partially dynamic environments, thereby enhancing the overall practicality and autonomy of mobile robots in real-world applications.

5. ACKNOWLEDGEMENTS

This work was carried out with the support of Petrobras, using resources from the R&D clause of the ANP, in partnership with the University of São Paulo (USP) and the Fundação de Apoio à Física e Química (FAFQ), under Cooperation Agreements No. 2023/00016-6 and 2023/00013-7.We also thank Ph.D. student Francisco Affonso for the support during our method validation.

6. REFERENCES

Affonso, F., Tommaselli, F.A.G., Capezzuto, G., Gasparino, M.V., Chowdhary, G. and Becker, M., 2025. “Crow: A self-supervised crop row navigation algorithm for agricultural fields”. *Journal of Intelligent & Robotic Systems*, Vol. 111, No. 1, p. 28.

Borges, G.A. and Aldon, M.J., 2004. “Line extraction in 2d range images for mobile robotics”. *Journal of intelligent and Robotic Systems*, Vol. 40, pp. 267–297.

Iqbal, J., Xu, R., Sun, S. and Li, C., 2020. “Simulation of an autonomous mobile robot for lidar-based in-field phenotyping and navigation”. *Robotics*, Vol. 9, No. 2, p. 46.

Kohlbrecher, S., Von Stryk, O., Meyer, J. and Klingauf, U., 2011. “A flexible and scalable slam system with full 3d motion estimation”. In *2011 IEEE international symposium on safety, security, and rescue robotics*. IEEE, pp. 155–160.

Konstantinidis, K., Alirezaei, M. and Grammatico, S., 2020. “Development of a detection and tracking of moving vehicles system for 2d lidar sensors”. *Delft Universitu of Technology*.

Peng, H., Zhao, Z. and Wang, L., 2024. “A review of dynamic object filtering in slam based on 3d lidar”. *Sensors*, Vol. 24, No. 2, p. 645.

Song, S., Lim, H., Lee, A.J. and Myung, H., 2022. “Dynavins: A visual-inertial slam for dynamic environments”. *IEEE Robotics and Automation Letters*, Vol. 7, No. 4, pp. 11523–11530.

Thomas, H., Agro, B., Gridseth, M., Zhang, J. and Barfoot, T.D., 2021. “Self-supervised learning of lidar segmentation for autonomous indoor navigation”. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 14047–14053.

Virgolino Soares, J.C., Medeiros, V.S., Abati, G.F., Becker, M., Caurin, G., Gattass, M. and Meggiolaro, M.A., 2023. “Visual localization and mapping in dynamic and changing environments”. *Journal of Intelligent & Robotic Systems*, Vol. 109, No. 4, p. 95.

Yu, C., Liu, Z., Liu, X.J., Xie, F., Yang, Y., Wei, Q. and Fei, Q., 2018. “Ds-slam: A semantic visual slam towards dynamic environments”. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 1168–1174.

7. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.