Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 6 Iss. 1 January 2026, pp: 395-403
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

LiDAR and Visual Perception-Based Indoor Semantic Mapping: Comparative Study of GMapping and SLAM Toolbox

Muhammad Salam Pararta Saragi[1*], Deden Pradeka[2], Anugrah Adiwilaga[3], Dyah Kusuma Dewi[4], Roni Permana Saputra[5]

[1,2,3]Program Studi Teknik Komputer, Universitas Pendidikan Indonesia, Indonesia
[4,5] Pusat Riset Mekatronika Cerdas, Badan Riset dan Inovasi Nasional, Indonesia

E-Mail: [1]salam.sen@upi.edu, [2]dedenpradeka@upi.edu, [3]anugrah.adiwilaga@upi.edu, [4]dyah010@brin.go.id, [5]roni.permana.saputra@brin.go.id

Received Dec 09th 2025; Revised Dec 30th 2025; Accepted Jan 17th 2026; Available Online Jan 31th 2026
Corresponding Author: Muhammad Salam Pararta Saragi

Abstract

This study investigates semantic embedding strategies for indoor mapping by comparing Trajectory-Based Payload Embedding (TPE) in GMapping and Pose-Based Payload Embedding (PPE) in SLAM Toolbox. A custom Turtlebot3 platform equipped with a 2D LiDAR and six RGB cameras was used in the Gazebo simulation to acquire geometric and visual data. Object segmentation results from YOLOv11 were integrated into occupancy grids using two distinct embedding workflows: scan-level batch attachment in TPE and point-level graph persistence in PPE. Performance evaluation employed two metrics: pixel-level accuracy and time cost under three varied velocity conditions, followed by a comparative analysis. Results show that PPE achieved higher accuracy (mean 86.83%) and lower variability, while maintaining negligible time cost (<0.5 ms). TPE, although simpler to implement, exhibited greater sensitivity to motion dynamics and higher computational variability (average 350.47 ms). These findings highlight a trade-off between accuracy and efficiency, suggesting PPE as the more suitable approach for real-time semantic SLAM, while TPE remains useful for lightweight integration scenarios. Beyond quantitative results, the study contributes methodological insights into how embedding granularity and persistence affect semantic consistency, offering guidance for future implementations in both simulated and real-world robotic navigation.

Keywords: Robot Navigation, Robotics, Semantic Mapping, Sensor Fusion, SLAM

1. INTRODUCTION

Simultaneous Localization and Mapping (SLAM), is a method used by autonomous mobile robots to construct or update a map of an unknown environment while simultaneously keeping track of their own location within that environment. It is crucial for autonomous navigation because it enables a robot to explore an area without prior knowledge and to localize itself based on the map it is creating in real time. SLAM techniques often use sensor data such as LiDAR or cameras to gather environmental information and generate occupancy grid maps that represent obstacles, free spaces, and robot positions [1], [2]. Mapping is the fundamental importance in mobile robot automation because it allows the robot to understand the spatial configuration of the environment. This environmental model serves as the basis for essential functions such as path planning and obstacle avoidance. Without mapping, a robot cannot plan feasible routes to its target because it lacks knowledge of possible obstacles or navigable pathways. SLAM provides continuous localization updates with respect to the generated map, enabling accurate position tracking and navigation [3].

Although SLAM frameworks provide reliable geometric maps, their representations are largely restricted to occupancy grids that describe free and occupied spaces. As indoor robotic applications become more complex and dynamic, this geometric perspective alone proves insufficient. Robots increasingly require semantic awareness such as the ability to identify objects, rooms, and contextual features using visual perception to navigate safely and adaptively, where recent studies highlight that semantic understanding is becoming a critical component in indoor navigation systems [4], [5]. Without such contextual reasoning, robots are prone to navigation errors, reduced situational awareness, and inefficient path planning.

Advances in visual perception and deep learning have created new opportunities to enrich SLAM systems with semantic information. This research enhanced panoramic cameras with YOLOv11 object

segmentation model to allow robots to detect and classify objects in real time [6], [7], [8]. Yet, despite their demonstrated effectiveness, these models are seldom incorporated directly into SLAM pipelines for robotic applications. When visual perception is integrated with SLAM, the resulting semantic-geometric maps provide a richer description of the environment, enabling robots to reason about both spatial layout and contextual elements. Such integration facilitates more informed decision-making and enhances navigation performance in complex indoor settings. Previous studies further indicate that semantic augmentation not only improves mapping accuracy but also supports task-oriented reasoning, including obstacle avoidance and goal-directed planning [5], [6].

In this study, semantic mapping is implemented through two distinct embedding strategies, as illustrated in Figure 1. The first approach, Trajectory-Based Payload Embedding (TPE), integrates semantic labels alongside the trajectory generated by GMapping. The second approach, Pose-Based Payload Embedding (PPE), leverages the pose-graph structure of SLAM Toolbox to attach semantic data to graph nodes. In both approaches, grid cell is annotated with label-specific information, visualized through distinct color encodings, and stored in a readable format. Both approaches also enables semantic information to be updated dynamically and maintained consistently across mapping scenarios. By comparing TPE and PPE, the study evaluates differences in semantic accuracy and computational efficiency.

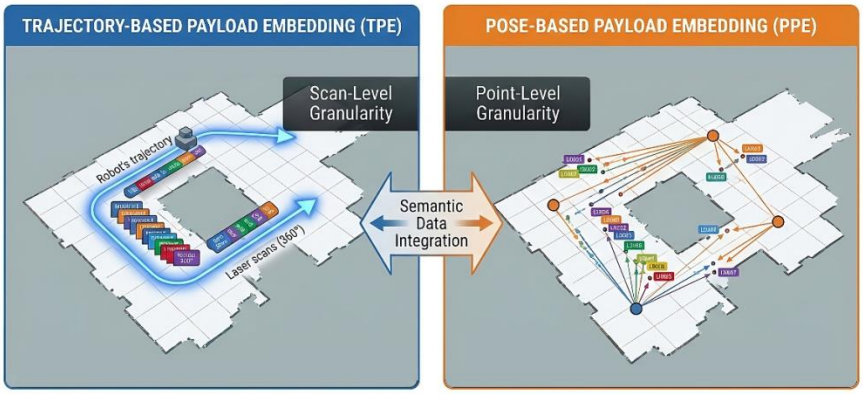

Figure 1. TPE and PPE Approach Overview

In line with the intended objectives and expected contributions of this research, the development and evaluation of semantic mapping are scoped within indoor environments characterized by moderate clutter and static elements. The experiments are conducted using the Turtlebot3 platform within the Gazebo simulation environment, with YOLOv11 employed for object segmentation. Performance evaluation focuses on pixel-level accuracy, cycle time, and semantic consistency across varied motion scenarios to ensure the applicability of both approaches in real-world indoor navigation tasks.

2. MATERIALS AND METHOD

To enable semantic mapping within SLAM systems, this study investigates two complementary embedding strategies that integrate object-level information into spatial representations. The approaches were selected to reflect distinct paradigms of semantic integration: one operating directly at the trajectory level, and the other at the pose-graph level. By comparing these strategies under controlled simulation scenarios, the study aims to highlight trade-offs in accuracy, efficiency, and consistency [9]. These comparisons provide insights into their applicability for real-world indoor navigation tasks.

2.1. Data Acquisition

The data acquisition process was implemented asynchronously, allowing LiDAR, camera, and odometry subsystems to operate in parallel without sequential blocking (see Figure 2). This architecture ensured that each sensor stream contributed independently to the mapping pipeline, thereby reducing latency and improving throughput to get the lowest computational cost [10].

The 2D LiDAR sensor, configured to produce 360 samples per scan, served as the primary source of geometric information. Each scan provided a full circular sweep of distance measurements, which were published to a dedicated topic for downstream SLAM processing. The LiDAR data established the occupancy grid structure, defining free, occupied, and unknown regions of the environment.

Complementing the LiDAR, six RGB cameras were mounted in a circular arrangement around the robot chassis, each with a 60° field of view. Together, they provided panoramic coverage of the indoor environment. Raw image streams were first corrected for lens distortion and then processed through an object segmentation model. Unlike bounding-box detection, segmentation generated pixel-level masks that outlined

the precise contours of detected objects. These masks were subsequently transformed into angular coordinates, aligning visual perception with the LiDAR's 360° sampling resolution. To maintain geometric consistency, only detections within valid angular regions overlapping the LiDAR's coverage were retained.

Odometry data were supplied by a custom robotic platform based on Turtlebot3, which integrated wheel encoder readings and inertial measurements. These signals were continuously published to localization topics, providing both motion trajectories and footprint boundaries. A rebroadcasting mechanism unified all sensor frames under a consistent namespace, ensuring that LiDAR scans, camera detections, and odometry estimates shared a common reference frame for fusion.

The synchronized fusion of these subsystems produced a semantic occupancy grid, where each cell carried both geometric and contextual information. LiDAR defined spatial boundaries, cameras enriched the grid with semantic labels, and odometry stabilized the alignment of sensor data. This integrated pipeline enabled real-time simulation of hazard-aware navigation, supporting adaptive decision-making in cluttered indoor environments.

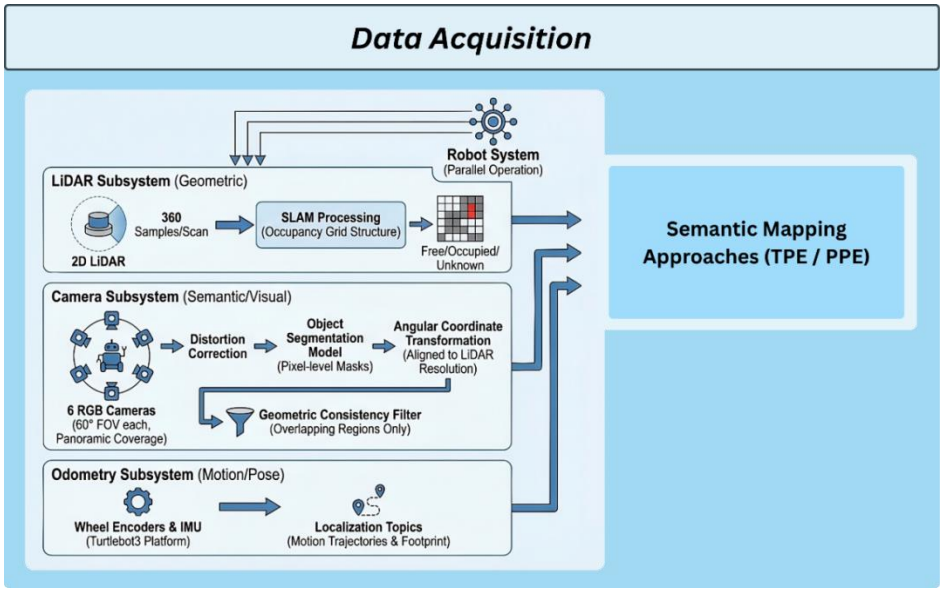

Figure 2. Data Acquisition

2.2. Semantic Embedding Approaches

Semantic embedding represents the process of enriching geometric maps with contextual information derived from visual perception. In the domain of SLAM, embedding strategies determine how semantic labels are integrated, thereby influencing the granularity, persistence, and consistency of the resulting map. This study explores two distinct approaches that reflect different philosophies of integration, as illustrated in Figure 3 and elaborated in the following section.

2.2.1 Trajectory-Based Payload Embedding (TPE)

TPE builds upon the particle filter mechanism of GMapping, where semantic labels are associated with the robot's trajectory and projected into the occupancy grid. In practice, each laser scan produces a 360-element array of labels, which is then attached to the trajectory node and registered into the occupancy grid during traversal. This means that every grid cell intersected by the robot's motion path is annotated with semantic information, effectively overlaying contextual data along the navigation trajectory.

The embedding process in TPE operates at the scan level: the entire set of labels is treated as a single batch per scan cycle. During map construction, these labels are indexed against the trajectory tree and integrated into the grid cells through scan registration. Because the system overwrites older data with the latest available segmentation result, semantic persistence is transient labels are primarily used at runtime to enrich the map but are not permanently embedded into the pose graph. This design emphasizes immediacy and simplicity, ensuring that semantic overlays follow the navigation timeline without additional synchronization overhead.

2.2.2 Pose-Based Payload Embedding (PPE)

PPE leverages the pose-graph architecture of SLAM Toolbox, attaching semantic information directly to graph nodes. Each node in the pose graph stores hazard labels derived from segmentation results, which are then propagated into the occupancy grid through per-point ray tracing. Unlike TPE's scan-level batch,

PPE enables point-level granularity, where every individual laser measurement can carry its own semantic label.

The embedding workflow begins with segmentation results being queued and matched to the closest LiDAR scan timestamp. These matching labels are subsequently copied into the occupancy grid, where ray tracing associates each laser point with its corresponding hazard information. Because the payloads are stored within the pose graph, semantic data in PPE are graph-persistent: they survive optimization, map merging, and even serialization/deserialization cycles. This persistence ensures that semantic annotations remain consistent throughout lifelong mapping scenarios, while timestamp synchronization reduces the risk of mismatch in multi-sensor setups.

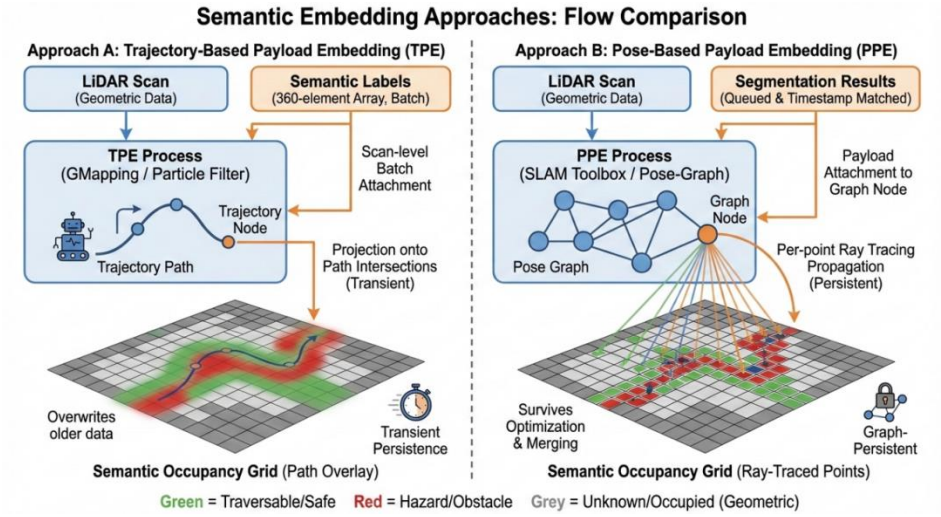

Figure 3. Flow Comparison

2.3. Evaluation

To assess semantic mapping performance, two complementary evaluation metrics are used: pixel-level accuracy and Cycle time. Each metric captures a different aspect of system effectiveness, ranging from annotation precision to computational efficiency and long-term reliability. Together, these measures provide a basis for comparing the accuracy and performance of TPE and PPE within indoor navigation scenarios [11].

2.3.1. Pixel-level accuracy

Pixel-level accuracy evaluation represents a fundamental metric for assessing semantic annotation precision in SLAM systems. Prior research demonstrates that quantitative experiments utilizing ground truth comparisons are essential for evaluating SLAM system accuracy in complex environments [12]. The approach of comparing system-generated maps against manually annotated ground truth through pixel-level analysis aligns with established practices in semantic SLAM evaluation [13], [14]In practice, this evaluation is implemented using Python-based image processing to calculate the deviation between the Occupied-label map and a manually annotated ground-truth map. The comparison relies on pixel-level color differences, thereby providing a direct, quantitative measure of semantic annotation accuracy.

2.3.2. Time Cost

Computational efficiency, measured through time cost analysis, is widely recognized in SLAM literature as a critical performance indicator [11], [15]. Runtime evaluation provides insight into how additional operations—such as semantic payload embedding—affect the responsiveness of the mapping pipeline [16], [17]. In this study, time cost is defined as the average computational duration required to integrate semantic labels into the occupancy grid or pose graph during each mapping update. This metric captures the efficiency of the embedding process by quantifying how quickly the system incorporates new sensor data and refreshes the semantic map. Lower time costs indicate higher responsiveness, while increased computational overhead from complex embedding operations may reduce the practicality of real-time deployment [17], [18].

2.4. Experimental Setup

The experiments were conducted using the self-designed custom robot platform within the Gazebo simulation environment. The robot was equipped with a 2D LiDAR sensor for geometric mapping and a six-

camera array for visual perception. Object segmentation was performed using a pre-trained YOLOv11 model, which provided real-time classification of indoor objects. The simulation environment was designed to represent moderately cluttered indoor spaces with static elements, with objects drawn from the pre-trained datasets. A preview of the experimental setup, along with the specifications of the hardware used to run the simulations, can be seen in Figure 4.

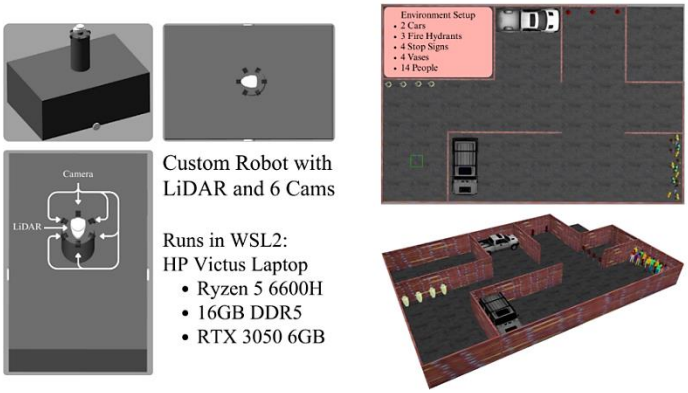

Figure 4. Experimental Setup

To assess the robustness of both approaches, experiments were conducted under varied speed conditions. Variations in the robot's linear velocity (m/s) were introduced to evaluate how semantic labels remain stable across repeated traversals. This scenario highlights the impact of different velocity settings on mapping accuracy and label consistency. For each semantic embedding approach, three trials were performed at different speed levels: low (0.08 m/s), medium (0.16 m/s), and high (0.24 m/s).

3. RESULTS AND DISCUSSION

The experimental evaluation was conducted to examine the stability of semantic embedding under varied velocity conditions. Figure 5 illustrated that trials were performed using two embedding approaches—Trajectory-Based Payload Embedding (TPE) and Pose-Based Payload Embedding (PPE)—across three speed levels: low (0.08 m/s), medium (0.16 m/s), and high (0.24 m/s). Each trial produced a semantic occupancy grid that combined geometric mapping with contextual annotations derived from visual perception.

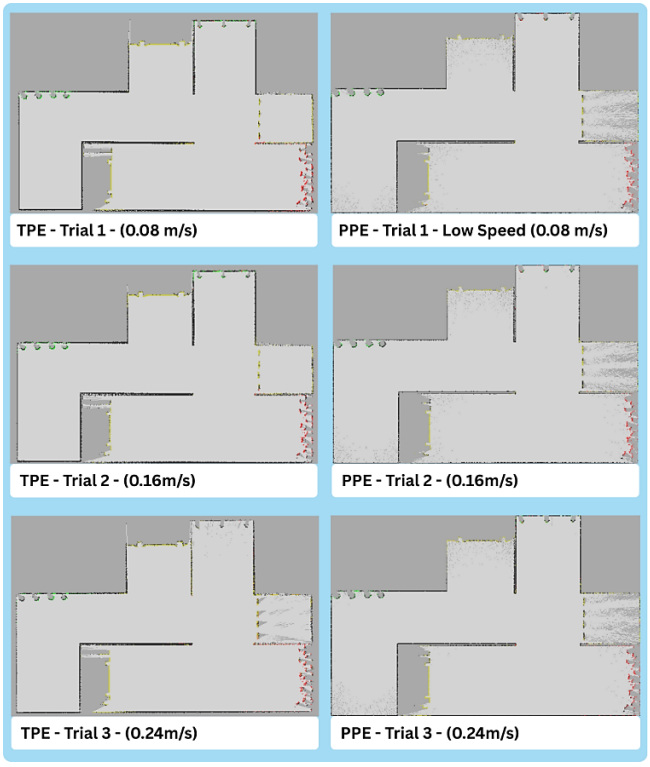

Figure 5. Semantic Map Results

At lower speeds, both approaches demonstrated stable label distribution and consistent alignment between semantic overlays and geometric structures. Semantic annotations were clearly visualized, with minimal dropout and strong correspondence to environmental features. As velocity increased, differences between the two methods became more apparent. TPE maintained lightweight integration but exhibited higher sensitivity to motion, resulting in occasional misalignment of labels along the trajectory. In contrast, PPE showed stronger resilience, with semantic data remaining consistent across repeated traversals due to its graph-based persistence and timestamp synchronization.

The comparative visualization of trials highlights how speed variations influence semantic density and label stability. At medium velocity, both methods retained acceptable consistency, though PPE produced smoother overlays across cluttered regions. At higher velocity, semantic dropout became more pronounced, particularly in TPE, indicating that motion dynamics directly affect the reliability of scan-level embedding. These observations establish the foundation for quantitative analysis of cycle time and pixel-level accuracy, which are discussed in the following sections.

While the observed decline in accuracy at higher velocities can be partially attributed to odometry drift, which introduces positional misalignment in trajectory-based embedding, visual factors also play a role. At increased speeds, the six-camera array experiences motion blur, reducing the sharpness of object boundaries and thereby increasing the likelihood of false positives in segmentation masks generated by YOLOv11 (see Figure 6). This degradation in image quality causes the system to misclassify background textures or blurred contours as valid objects, leading to spurious semantic overlays. Consequently, the accuracy drop is not solely a geometric issue but also reflects the susceptibility of visual perception to motion dynamics. By acknowledging both odometry drift and motion-induced false positives, the analysis highlights that semantic SLAM performance depends on the interplay between geometric stability and visual fidelity.

Figure 6. Influence of Motion Dynamics on Visual Perception Quality

3.1. Pixel-Level Accuracy

The accuracy of semantic label placement was evaluated by comparing annotated maps with ground-truth object positions. Both approaches' accuracy, shown in Table 1, was measured at the grid-cell level, reflecting how well semantic labels aligned with the map.

Table 1. Results of Pixel-Level Accuracy Calculation

Metric	Trajectory-Based Payload Embedding (TPE — GMapping)	Pose-Based Payload Embedding (PPE — SLAM Toolbox)
Accuracy of Trial 1 (%)	87.6	88.7
Accuracy of Trial 2 (%)	83.9	86.7
Accuracy of Trial 3 (%)	81.3	85.1
Cumulative		
Mean Accuracy (%)	84.27	86.83
Standard Deviation	2.58	1.47

The evaluation of pixel-level accuracy shows how semantic labels were aligned with ground-truth object positions across different trials. As shown in Table 1, both embedding approaches achieved relatively high accuracy in Trial 1, with values exceeding 87%. This indicates that under low-speed conditions, semantic overlays were consistently mapped to the correct grid cells, reflecting stable synchronization between sensor data and map annotation.

In subsequent trials, accuracy values decreased slightly as traversal speed increased. For TPE, accuracy dropped from 87.6% in Trial 1 to 81.3% in Trial 3, suggesting that scan-level embedding is more sensitive to motion dynamics and may experience label misalignment when the robot moves faster. PPE also exhibited a reduction in accuracy, though the decline was less pronounced, with values ranging from 88.7% to 85.1%. This relative stability reflects the benefits of timestamp matching and graph-based persistence, which help maintain semantic consistency even under higher-velocity conditions.

The cumulative metrics further illustrate these trends. TPE achieved a mean accuracy of 84.27% with a standard deviation of 2.58%, indicating greater variability across trials. In contrast, PPE recorded a higher

mean accuracy of 86.83% and a lower standard deviation of 1.47%, suggesting more consistent performance. These results imply that while both approaches are capable of producing semantically enriched maps, their robustness differs in terms of sensitivity to speed variations and consistency of label placement.

3.2. Time Cost

This metric was measured as the processing time required for each mapping update, explicitly ignoring geometric reconstruction and measuring only the semantic payload embedding process. Thus, the data shown in Table 2 does not reflect the overall computational demand of the mapping pipeline, while it only highlights the additional overhead introduced by semantic integration.

Table 2. Results of Payload Embedding Time Cost

Metric	Trajectory-Based Payload Embedding (TPE — GMapping)			Pose-Based Payload Embedding (PPE — SLAM Toolbox)		
	Fastest (ms)	Slowest (ms)	Average (ms)	Fastest (ms)	Slowest (ms)	Average (ms)
Time Cost of Trial 1	0.28	1085.86	316.93	0.15	1.07	0.32
Time Cost of Trial 2	0.32	1348.11	347.25	0.15	3.83	0.5
Time Cost of Trial 3	0.35	1103.02	372.67	0.18	1.93	0.39
Cumulative						
Avg. of All Trials (ms)		350.47			0.40	
Standard Deviation		362.87			0.43	

For Trajectory-Based Payload Embedding (TPE), average time costs across trials ranged between 316.93 ms and 372.67 ms, with cumulative mean of 350.47 ms. While the fastest recorded embedding was only 0.28 ms, the slowest instances extended beyond 1,300 ms, indicating significant variability. This fluctuation reflects the batch nature of scan-level embedding, where the entire 360-element label array is processed per scan. Under certain conditions, especially when sensor streams are desynchronized or when trajectory indexing requires additional computation, the embedding process can incur substantial delays.

In contrast, Pose-Based Payload Embedding (PPE) demonstrated consistently lower time costs. Average values remained below 0.5 ms across all trials, with a cumulative mean of 0.4 ms. The fastest embedding was 0.15 ms, while the slowest was 3.83 ms. This stability is attributable to the point-level embedding strategy, in which labels are attached directly to pose-graph nodes and propagated via ray tracing. Although this design involves per-point operations, the graph-persistent architecture ensures efficient serialization and minimizes runtime variability.

3.3. Comparative Analysis

The comparative evaluation of semantic embedding approaches reveals distinct trade-offs between accuracy and complexity. As presented in Section 3.1, PPE consistently achieved higher pixel-level accuracy and lower trial-to-trial variability. Its graph-persistent design and timestamp synchronization contributed to more stable semantic overlays, particularly under increased traversal speeds. In contrast, TPE demonstrated greater sensitivity to motion dynamics, with accuracy declining more sharply as velocity increased.

However, the results of Section 3.2 highlight a contrasting perspective when computational efficiency is considered. TPE incurred significantly higher time costs, with average embedding durations exceeding 300 ms and occasional peaks above 1,000 ms. This variability reflects the batch nature of scan-level embedding, where the entire label array must be processed per scan cycle. PPE, on the other hand, maintained consistently low time costs, averaging less than 0.5 ms across all trials. Its point-level embedding strategy, despite involving per-point ray tracing, benefited from graph-based serialization that minimized runtime overhead.

Taken together, these findings illustrate a clear trade-off between accuracy and efficiency. PPE offers superior accuracy and stability with negligible computational overhead, making it well-suited for real-time deployment scenarios where responsiveness is critical. TPE, while simpler in design and capable of immediate semantic enrichment at runtime, introduces substantial variability in processing time that may hinder scalability in dynamic environments.

This comparative analysis underscores the importance of selecting an embedding strategy based on application requirements. Systems prioritizing robust semantic consistency and real-time responsiveness may favor PPE, whereas contexts emphasizing lightweight integration and runtime simplicity could still benefit from TPE despite its computational variability.

4. CONCLUSION

This study compared two semantic embedding strategies TPE in GMapping and PPE in SLAM Toolbox for indoor semantic mapping using LiDAR and visual perception. Both approaches successfully enriched occupancy grids with contextual information, demonstrating the feasibility of integrating semantic

data into SLAM pipelines. The evaluation of pixel-level accuracy showed that PPE consistently achieved higher accuracy and lower variability across trials. Its graph-persistent design and timestamp synchronization ensured semantic consistency even under higher traversal speeds. In contrast, TPE exhibited greater sensitivity to motion dynamics, with accuracy declining more sharply as velocity increased. These findings highlight the importance of embedding granularity and persistence in maintaining reliable semantic overlays.

In terms of computational efficiency, the time cost analysis revealed a significant difference between the two approaches. TPE incurred longer, more variable embedding durations, averaging 350.47 ms, reflecting the batch nature of scan-level integration. PPE, on the other hand, maintained negligible overhead, averaging less than 0.5 ms across trials. This stability underscores PPE's suitability for real-time deployment scenarios where responsiveness is critical.

Taken together, the comparative analysis demonstrates a clear trade-off between accuracy and efficiency. PPE emerges as the more robust and efficient solution for real-time semantic SLAM, particularly in dynamic environments requiring stable label placement and low computational latency. TPE, while simpler in design and emphasizing lightweight integration, remains relevant for scenarios prioritizing immediate runtime enrichment and reduced implementation complexity. Future research may extend this evaluation to physical robot deployments and explore hybrid strategies that combine the immediacy of TPE with the persistence of PPE.

REFERENCES

[1] A. A. Fikri and L. Anifah, "Mapping And Positioning System On Omnidirectional Robot Using Simultaneous Localization And Mapping (Slam) Method Based On Lidar," *J. Teknol.*, vol. 83, no. 6, pp. 41–52, Sep. 2021, doi: 10.11113/jurnalteknologi.v83.16918.

[2] P. S. Laksono and T. M. Kusuma, "Performance Analysis Of Hector Slam And Gmapping For Navigation For Mobile Robot Navigation," *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 27, no. 2, pp. 144–153, Aug. 2022, doi: 10.35760/tr.2022.v27i2.6063.

[3] Y. Ran, X. Xu, Z. Tan, and M. Luo, "A Review of 2D Lidar SLAM Research," *Remote Sens. (Basel).*, vol. 17, no. 7, p. 1214, Mar. 2025, doi: 10.3390/rs17071214.

[4] Q. An *et al.*, "Dynamic SLAM Dense Point Cloud Map by Fusion of Semantic Information and Bayesian Moving Probability," *Sensors*, vol. 25, no. 17, p. 5304, Aug. 2025, doi: 10.3390/s25175304.

[5] J. Li, R. Zhang, Y. Liu, Z. Zhang, R. Fan, and W. Liu, "The Method of Static Semantic Map Construction Based on Instance Segmentation and Dynamic Point Elimination," *Electronics (Basel).*, vol. 10, no. 16, p. 1883, Aug. 2021, doi: 10.3390/electronics10161883.

[6] A. Tourani, H. Bavle, J. L. Sanchez-Lopez, and H. Voos, "Visual SLAM: What Are the Current Trends and What to Expect?," *Sensors*, vol. 22, no. 23, p. 9297, Nov. 2022, doi: 10.3390/s22239297.

[7] Z. Zheng, K. Su, S. Lin, Z. Fu, and C. Yang, "Development of vision–based SLAM: from traditional methods to multimodal fusion," *Robotic Intelligence and Automation*, vol. 44, no. 4, pp. 529–548, Jul. 2024, doi: 10.1108/RIA-10-2023-0142.

[8] M. R. Hanafi, M. A. Faadhilah, M. T. Dwi Putra, and D. Pradeka, "Comparison Analysis of Bubble Sort Algorithm with Tim Sort Algorithm Sorting Against the Amount of Data," *Journal of Computer Engineering, Electronics and Information Technology*, vol. 1, no. 1, pp. 29–38, Apr. 2022, doi: 10.17509/coelite.v1i1.43794.

[9] A. Adiwilaga, I. Taufikurrahman, E. Hidayat, and B. R. Trilaksono, "Design of a modular, compact, long endurance autonomous underwater vehicle with gliding capabilities for research purpose operations," in *2017 2nd International Conference on Control and Robotics Engineering (ICCRE)*, IEEE, Apr. 2017, pp. 150–154. doi: 10.1109/ICCRE.2017.7935060.

[10] Q. U. Islam, H. Ibrahim, P. K. Chin, K. Lim, and M. Z. Abdullah, "MVS-SLAM: Enhanced multiview geometry for improved semantic RGBD SLAM in dynamic environment," *J. Field Robot.*, vol. 41, no. 1, pp. 109–130, Jan. 2024, doi: 10.1002/rob.22248.

[11] Y. Lu, H. Wang, J. Sun, and J. A. Zhang, "Enhanced Simultaneous Localization and Mapping Algorithm Based on Deep Learning for Highly Dynamic Environment," *Sensors*, vol. 25, no. 8, p. 2539, Apr. 2025, doi: 10.3390/s25082539.

[12] Y. Sun *et al.*, "Multi-Objective Location and Mapping Based on Deep Learning and Visual Slam," *Sensors*, vol. 22, no. 19, p. 7576, Oct. 2022, doi: 10.3390/s22197576.

[13] Z. Li, J. Zhao, X. Zhou, S. Wei, P. Li, and F. Shuang, "RTSDM: A Real-Time Semantic Dense Mapping System for UAVs," *Machines*, vol. 10, no. 4, p. 285, Apr. 2022, doi: 10.3390/machines10040285.

[14] F. Wang, L. Zhao, Z. Xu, H. Liang, and Q. Zhang, "LDVI-SLAM: a lightweight monocular visual-inertial SLAM system for dynamic environments based on motion constraints," *Meas. Sci. Technol.*, vol. 35, no. 12, p. 126301, Dec. 2024, doi: 10.1088/1361-6501/ad71e7.

[15] Y. Wu *et al.*, "An FPGA Based Energy Efficient DS-SLAM Accelerator for Mobile Robots in Dynamic Environment," *Applied Sciences*, vol. 11, no. 4, p. 1828, Feb. 2021, doi: 10.3390/app11041828.

[16] Q. Ji, Z. Zhang, Y. Chen, and E. Zheng, "DRV-SLAM: An Adaptive Real-Time Semantic Visual SLAM Based on Instance Segmentation Toward Dynamic Environments," *IEEE Access*, vol. 12, pp. 43827–43837, 2024, doi: 10.1109/ACCESS.2024.3379269.

[17] Y. Shen and X. Zhang, "A dynamic SLAM system with YOLOv7 segmentation and geometric constraints for indoor environments," *Robotica*, vol. 43, no. 7, pp. 2527–2545, Jul. 2025, doi: 10.1017/S0263574725101823.