Proceedings of Engineering and Technology Innovation, vol. 32, 2026, pp. 118-133

Enhancing Visual SLAM Robustness in Dynamic Scenes with YOLOv5-Assisted ORB-SLAM3

Rajaa Wejood Ali*, Heba Hakim, Dr. Mohammed Abd Ali Al-Ibadi

Department of Computer Engineering, University of Basrah, Basrah, Iraq

Received 04 June 2025; received in revised form 20 December 2025; accepted 29 December 2025

DOI: https://doi.org/10.46604/peti.2026.15235

Abstract

This study presents an enhanced visual SLAM (Simultaneous Localization and Mapping) framework that integrates ORB-SLAM3 with the YOLOv5 real-time object detection model to improve pose accuracy in dynamic environments. Although ORB-SLAM3 achieves robust performance in static scenes, its reliance on ORB feature tracking often degrades accuracy in the presence of moving objects. To overcome this limitation, YOLOv5 is employed to identify dynamic regions in each video frame, enabling the system to remove motion-related feature points before matching. This filtering mechanism reduces the influence of dynamic objects on trajectory estimation and enhances overall system robustness. The proposed method was evaluated using dynamic datasets, including BONN and TUM RGB-D, and further validated through real-world experiments with an Intel RealSense D435i camera. Experimental results demonstrate substantial improvements in pose accuracy compared with the baseline ORB-SLAM3 and the RTAB-Map system, confirming the effectiveness of the YOLOv5-assisted ORB-SLAM3 integration in dynamic scenes.

Keywords: ORB-SLAM3, YOLOv5, dynamic environments, pose estimation, visual SLAM

1. Introduction

The ability of a mobile robot to autonomously navigate an unknown environment relies heavily on its capacity to simultaneously localize itself and construct a consistent map of its surroundings. This problem—commonly referred to as Simultaneous Localization and Mapping (SLAM)—is central to mobile robotics and has been extensively studied over the past two decades. Classical SLAM techniques, such as Extended Kalman Filter SLAM (EKF-SLAM) [1], FastSLAM [2], and GraphSLAM [3], have proven effective in static environments, where the assumption of scene stability holds. However, these systems tend to degrade significantly in dynamic settings, where moving objects introduce incorrect feature associations, partial occlusions, and temporal inconsistencies in the map. As noted in a recent comprehensive survey by Chen et al. [4], conventional SLAM approaches struggle in unstructured or non-rigid scenes, prompting a shift toward more adaptive solutions.

Visual SLAM (vSLAM), which relies on camera inputs rather than LiDAR or other expensive sensors, has emerged as a promising approach for navigation in both indoor and outdoor environments. Algorithms such as ORB-SLAM2 and its successor ORB-SLAM3 [5] are widely adopted due to their robustness, support for multiple sensor modalities, and open-source implementations. Nevertheless, both systems assume that the environment remains largely static. In real-world scenarios—particularly in crowded or dynamic indoor environments—this assumption is often violated, leading to degraded pose estimation and corrupted mapping due to the inclusion of transient, non-static features.

To overcome these limitations, recent studies have incorporated semantic awareness into SLAM pipelines by leveraging deep learning models. These systems aim to detect, segment, and exclude dynamic elements from the SLAM process. For

* Corresponding author. E-mail address: pgs.rajaa.wejood@uobasrah.edu.iq

example, DynaVINS [6] and YDD-SLAM [7] integrate deep object detectors and segmenters, such as YOLOv5 and Mask R-CNN, to identify and filter moving objects in real-time. Similarly, hybrid approaches such as YS-SLAM [8] and MS-ViT SLAM [9] fuse feature-based SLAM with semantic segmentation, enhancing resilience against dynamic clutter. Other systems, including ReFusion and StaticFusion [10], employ dense tracking and motion analysis to isolate dynamic regions, but these methods are often sensitive to depth noise and computationally intensive, limiting real-time applicability.

Despite these advancements, a key limitation remains: most existing systems rely on supervised learning models trained on a fixed set of known object classes. Consequently, unknown or unlabeled dynamic objects—those not included in the training set—may still contaminate the visual input, causing localization drift and inconsistencies in the resulting map [11]. Some recent efforts attempt to mitigate this issue by using motion-based priors or depth inconsistencies; however, challenges related to computational cost and generalization persist.

To address this gap, the present study introduces an enhanced visual SLAM system based on ORB-SLAM3 that integrates YOLOv5 for object-level dynamic filtering. Unlike segmentation-heavy approaches, the use of object detection ensures high-speed inference while effectively identifying key dynamic actors in the scene. By removing dynamic features from the tracking pipeline, the system improves pose accuracy without significant computational overhead. RTAB-Map [12] is not incorporated into the framework; instead, it is used as a baseline in experimental comparisons to evaluate robustness and accuracy across a range of dynamic conditions. The proposed method is assessed on public RGB-D benchmarks, including TUM RGB-D [13] and BONN Dynamic RGB-D [14], as well as through real-world experiments using an Intel RealSense D435i sensor. Results demonstrate that the YOLOv5-ORB-SLAM3 system provides improved localization accuracy, enhanced resilience to both known and unknown moving objects, and greater overall robustness compared to standard ORB-SLAM3 and RTAB-Map.

The remainder of this paper is organized as follows. Section 2 reviews related research on visual SLAM in dynamic environments. Section 3 describes the proposed system architecture and details the integration of object detection for dynamic feature filtering. Section 4 outlines the experimental setup, datasets, and evaluation metrics. Section 5 presents and analyzes the results. Finally, Section 6 summarizes the contributions of this work and discusses directions for future research.

2. Related Work

Localization accuracy and map consistency in static situations have significantly improved due to recent developments in visual and visual-inertial SLAM. However, because moving objects are present in dynamic environments, performance degradation remains a significant issue. Numerous strategies have been put forth to address this problem, such as deep learning-assisted dynamic object filtering, geometry-based motion segmentation, and semantic SLAM. Relevant to the suggested approach, this section examines representative studies in dynamic scene management, learning-based SLAM upgrades, and classical SLAM systems.

2.1. Visual SLAM in Dynamic Environments

Traditional visual SLAM systems such as ORB-SLAM2 [15] assume static scenes, leading to inaccuracies in the presence of moving objects. To overcome this limitation, hybrid methods have been developed. DynaSLAM [16] integrates deep learning–based object detection into ORB-SLAM [17], improving robustness but struggling with low-texture scenes and limited real-time performance. Dyna VINS [6] extends this concept by combining visual–inertial odometry with motion detection, achieving greater stability but facing similar latency issues.

2.2. Integration of Deep Learning with SLAM

Deep learning techniques have recently been introduced into visual SLAM to improve performance in dynamic environments. YDD-SLAM [7] couples YOLOv5 with depth filtering to eliminate dynamic features before matching, while

SEG-SLAM [15] employs semantic segmentation with geometric constraints to maintain map consistency. YLS-SLAM [16] adopts a lightweight YOLOv5s backbone for faster inference, demonstrating improved localization in complex scenes. Despite these advancements, most existing systems rely on segmentation-based filtering or are built upon ORB-SLAM2, which lacks visual–inertial fusion and causes delays in real-time applications. To address these issues, this study proposes an enhanced YOLOv5-ORB-SLAM3 framework that integrates dynamic object detection directly into the feature extraction stage, filtering unstable keypoints before feature matching to achieve improved accuracy and efficiency.

2.3. Loop Closure and Map Optimization

Loop closure methods such as RTAB-Map are widely used for global optimization but can produce false matches in dynamic environments. Recent studies have attempted to integrate object detection to alleviate this problem. For instance, a dynamic-scene ORB-SLAM3 variant using YOLOv5 improves localization but lacks a comprehensive loop-closure mechanism. The proposed YOLOv5-ORB-SLAM3 system extends this concept by combining dynamic filtering with full map optimization, enhancing both global consistency and robustness.

2.4. Limitations and Research Gaps

Although substantial progress has been made, several open challenges remain:

(1) Handling Unknown Dynamic Objects: Existing methods depend on predefined object categories and fail to manage unseen dynamic entities.
(2) Real-Time Performance: Deep models such as YOLOv5 and SegNet introduce computational overhead that limits frame-rate performance.
(3) Comprehensive Map Optimization: Most systems do not correct mapping errors introduced by dynamic features, leading to inconsistency and reduced map quality.

Unlike previous YOLO-based SLAM systems, YDD-SLAM and YLS-SLAM [16], the proposed YOLOv5-ORB-SLAM3 integrates dynamic object detection directly into the ORB-SLAM3 pipeline with visual–inertial fusion, achieving better robustness and maintaining real-time performance.

3. System Overview

This work employs an integrated system combining YOLOv5 for object detection with ORB-SLAM3 for visual SLAM, aiming to construct a semantically enriched 3D map in dynamic environments. Fig. 1 illustrates the architecture of the proposed YOLOv5-ORB-SLAM3 framework. The system is built upon the ORB-SLAM3 backbone and extends it with an additional thread for real-time object detection using YOLOv5. The complete system comprises five parallel threads: YOLO-based object detection, tracking, local mapping, loop closing, and, optionally, visual-inertial odometry (VIO) for multi-sensor setups.

The YOLOv5 model detects dynamic objects in real-time by generating bounding box outputs from the RGB-D camera. These bounding boxes are then used to filter out ORB features associated with dynamic objects before the tracking stage proceeds with the pose estimation. The tracking thread relies solely on static background keypoints to estimate the camera pose, while loop closures are detected and corrected using a place recognition module integrated into the ORB-SLAM3 core.

To balance real-time performance with detection accuracy, the system classifies each detected object into one of three categories: dynamic, possibly dynamic, or static, based on YOLOv5's object class labels. Temporal consistency checks are applied across successive frames; if no dynamic objects are detected within a window of five or more frames, the environment is considered static.

The dynamic feature filtering module leverages bounding box coordinates to identify and remove ORB feature points corresponding to recognized dynamic objects, such as humans. Geometric and depth-based heuristics are employed to address

edge cases, including overlapping bounding boxes or detection failures.

Camera pose is estimated using the remaining static ORB feature points. The system establishes 2D–3D correspondences between map points and image keypoints to solve the Perspective-n-Point (PnP) problem, ensuring consistently accurate trajectory estimation in dynamic indoor environments. Features from multiple viewpoints are aggregated and forwarded to local mapping and loop closure modules in multi-camera configurations, supporting robust and semantically aware SLAM.

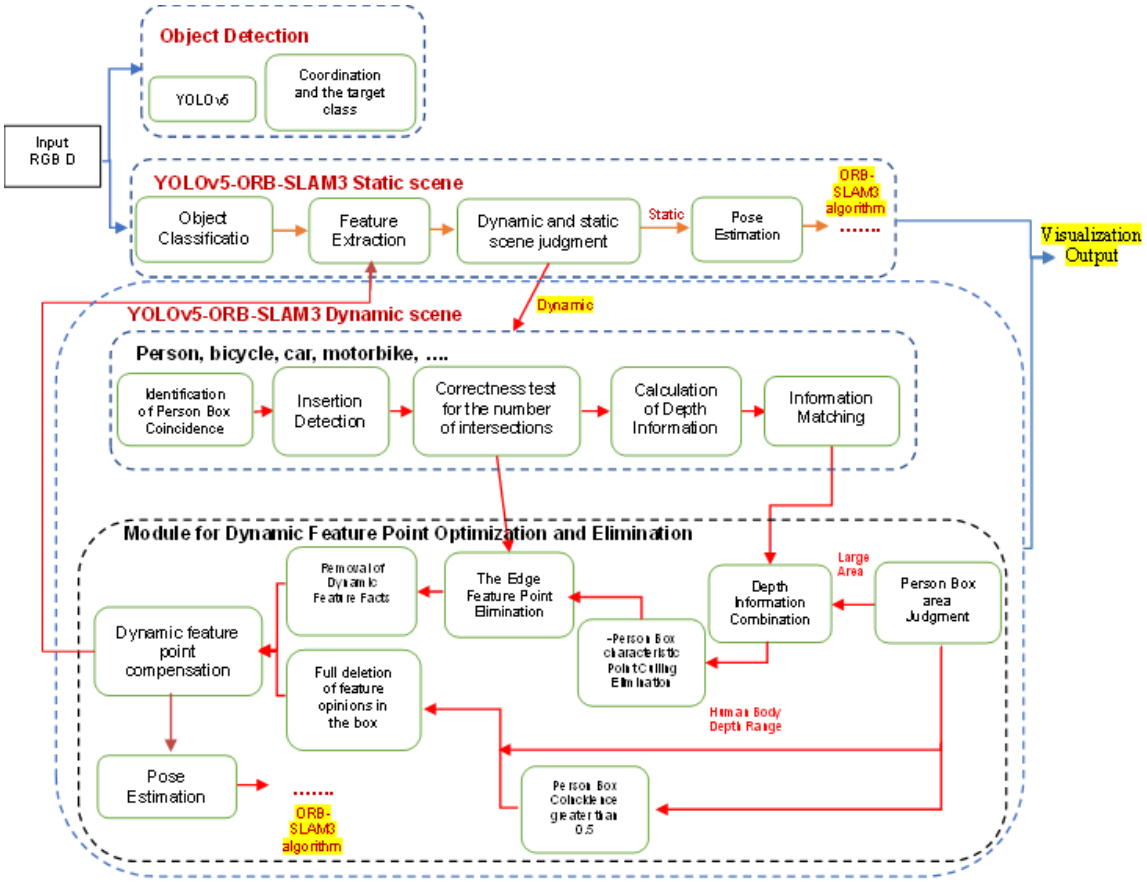


Fig. 1 YOLOv5-ORB-SLAM3 Structure

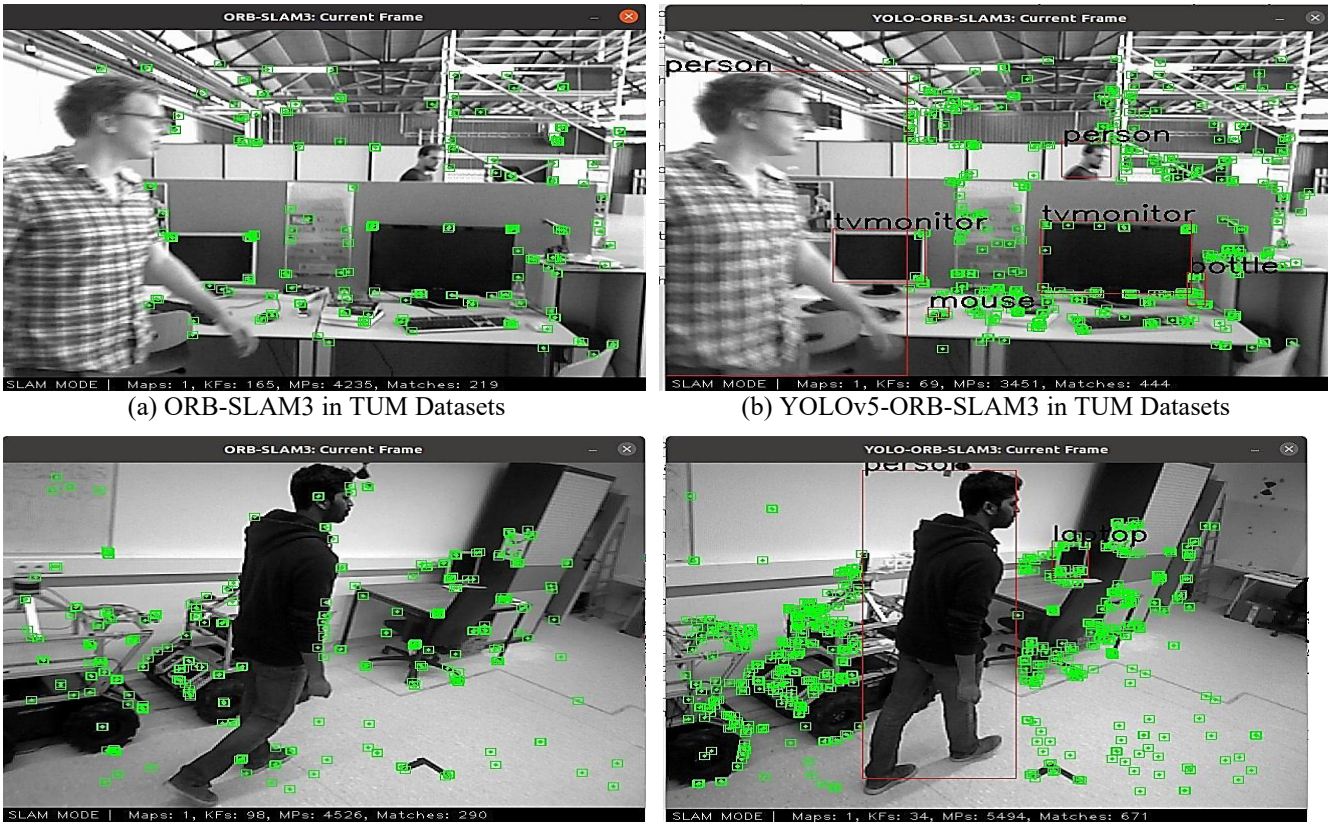
Feature management and dynamic compensation are continuously adjusted to account for changes in scene content. The feature extraction threshold is increased to compensate for points filtered out as dynamic, while it is reduced to conserve computational resources when the scene is identified as predominantly static. Detection and SLAM results are visualized in real-time, with bounding boxes displayed around objects according to their dynamic, static, or possibly dynamic classification, providing immediate feedback to validate the dynamic feature filtering process.

The system was evaluated on the TUM RGB-D [17] and BONN Dynamic RGB-D [14] datasets, as well as using real-time data from an Intel RealSense D435i camera. ORB-SLAM3 was employed to estimate camera poses and generate a sparse 3D map in real-time [18], while YOLOv5 [19], pretrained on the COCO dataset [20], detected and classified objects within the scene. ROS Noetic served as the middleware, enabling communication among system modules and managing sensor input from the D435i. Camera calibration parameters were obtained to ensure accurate projection from 2D detections to 3D coordinates. YOLOv5 was executed at predetermined intervals (e.g., every fifth frame) to maintain real-time performance, producing object bounding boxes, class labels, and confidence scores. Concurrently, ORB-SLAM3 monitored visual features to determine the camera trajectory and reconstruct the environment.

A semantic fusion strategy was implemented to associate 2D object detections with ORB-SLAM3 map points. Bounding boxes were projected into three-dimensional space using the current camera pose and intrinsic parameters, and semantic labels were assigned to nearby map points. The system was developed in Python and C++, with YOLOv5 running in PyTorch and

ORB-SLAM3 integrated via ROS. Experiments were conducted on a workstation equipped with an Intel i7 CPU and NVIDIA RTX 3060 GPU. System performance was evaluated in both static and dynamic environments, using Absolute Trajectory Error (ATE) for quantitative trajectory accuracy and qualitative assessment of semantic labeling on the 3D map.

Fig. 2 Comparison between the baseline ORB-SLAM3 and the planned YOLOv5-ORB-SLAM3 pipeline. ORB-SLAM3 employs inertial sensing, real-time loop closure, and multi-map processing, but still pulls feature points from dynamic objects, which reduces tracking accuracy. Before calculating the basic matrix using RANSAC, the YOLOv5-ORB-SLAM3 variation uses real-time object detection to filter out keypoints connected to moving objects. By masking dynamic classes such as "person" and "car," only static ORB keypoints are retained and matched across frames, resulting in more reliable feature correspondences and enhanced resilience.



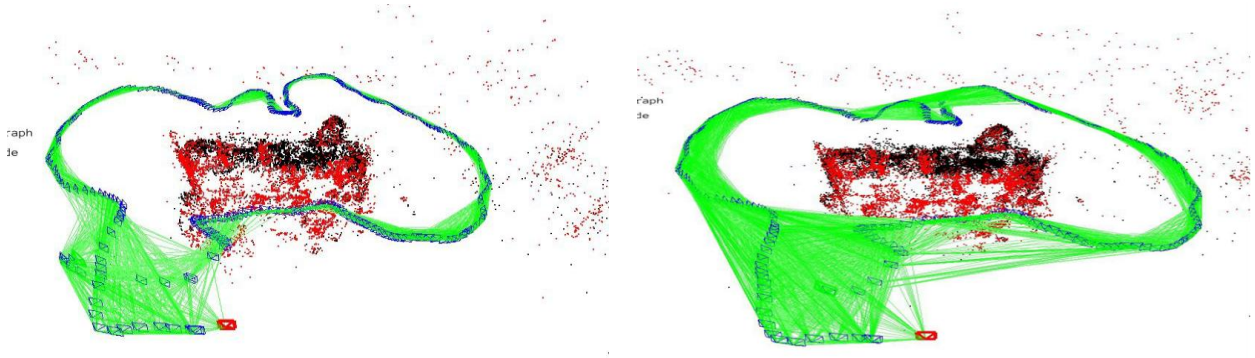
(a) ORB-SLAM3 in TUM Datasets (b) YOLOv5-ORB-SLAM3 in TUM Datasets

(c) ORB-SLAM3 in BONN Dynamic Datasets (d) YOLOv5-ORB-SLAM3 in BONN Dynamic Datasets

Fig. 2 Human keypoint filtering process

Because of its tightly integrated visual–inertial optimization and effective feature extraction, ORB-SLAM3 exhibits improved runtime performance; yet, its resilience is restricted in highly dynamic settings. By using semantic object identification to detect and filter dynamic areas, YOLOv5-ORB-SLAM3, on the other hand, lessens the impact of moving objects on pose estimation. Under dynamic situations, this integration results in a discernible improvement in tracking accuracy and trajectory consistency. However, there is a trade-off between accuracy and real-time speed due to the moderate computational burden introduced by the extra deep learning module.

In addition to the improved camera trajectory, YOLOv5-ORB-SLAM3 produces a sparse semantic map that is enhanced with semantic annotations and dynamic object filtering, while the ORB-SLAM3 algorithm generates a feature-based map with camera poses, registered map points, and graph edges. Fig. 3 shows an example of the final output for ORB-SLAM3 and the suggested YOLOv5-ORB-SLAM3 on the TUM RGB-D dataset's fr3_long_office_household sequence. The blue squares show estimated camera poses, the red dots provide reconstructed map points, and the green lines show pose correlations between successive frames. Compared with the baseline ORB-SLAM3, the YOLOv5-ORB-SLAM3 system provides a cleaner and more stable trajectory by deleting dynamic-object keypoints before mapping and pose estimation.



(a) ORB-SLAM3 (b) YOLOv5-ORB-SLAM3

Fig. 3 Final output of ORB-SLAM3 vs. YOLOv5-ORB-SLAM3 in a long office household

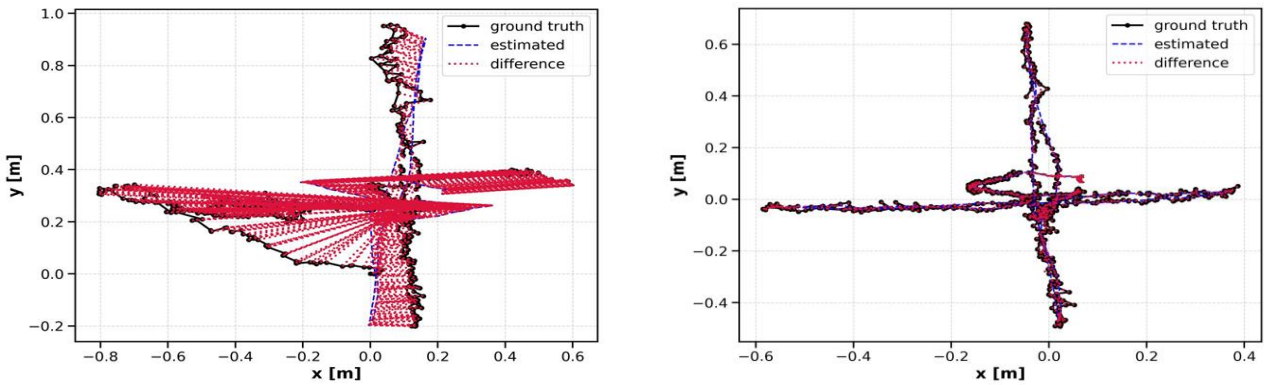
4. Results and Evaluations

This section covers the assessment of the proposed YOLOv5-ORB-SLAM3 system in dynamic situations, utilizing the TUM RGB-D and BONN Dynamic RGB-D datasets. These datasets comprise sequences with varied amounts of motion, including moving persons and objects, giving an appropriate baseline for measuring robustness. The incorporation of YOLOv5 into ORB-SLAM3 enables the elimination of dynamic zones before feature extraction and pose estimation, boosting the reliability of feature correspondences.

Performance was tested using the Absolute Trajectory Error (ATE), calculated as Root Mean Square Error (RMSE) compared to ground-truth trajectories. The approach continuously increases localization accuracy in highly dynamic environments while maintaining real-time speed despite the increased processing introduced by the semantic detection module. A noted difficulty is that excessive removal of dynamic areas can occasionally lower the amount of usable ORB features, which may lead to temporary pose estimation instability. Despite this, the overall findings indicate better robustness over the baseline ORB-SLAM3.

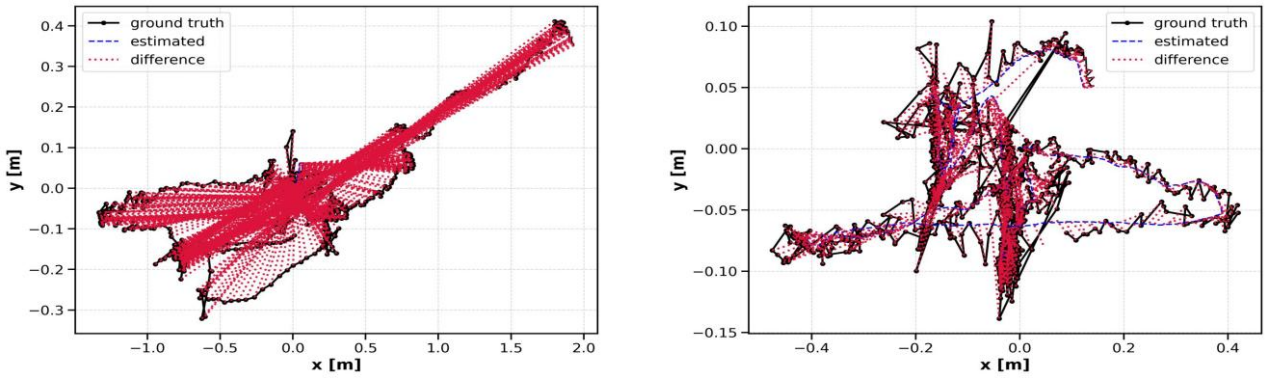
4.1 TUM RGB-D Datasets

For the fr3_walk_xyz, fr3_walk_rpy, and fr3_walk_halfsphere sequences of the TUM RGB-D dataset, Figs. 4-6 show qualitative comparisons between the ground-truth trajectories and those predicted by ORB-SLAM3 and the suggested YOLOv5-ORB-SLAM3 system. These scenes are particularly demanding owing to quick camera mobility and the presence of several moving objects. ORB-SLAM3 demonstrates substantial drift and pose instability, showing its susceptibility to dynamic aspects in the scene, whereas YOLOv5-ORB-SLAM3 yields trajectories that closely mirror the ground truth, displaying much enhanced stability and accuracy throughout all sequences.



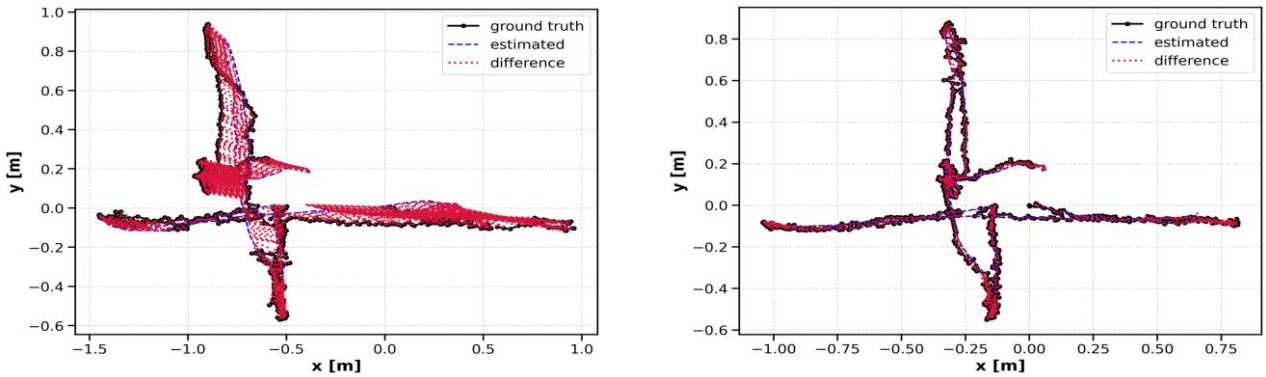
(a) ORB_SLAM3 (b) YOLOv5-ORB-SLAM3

Fig. 4 Trajectory comparison on the fr3_walk_xyz sequence



(a) ORB_SLAM3 (b) YOLOv5-ORB-SLAM3

Fig. 5 Trajectory comparison on the fr3_walk_rpy sequence



(a) ORB_SLAM3 (b) YOLOv5-ORB-SLAM3

Fig. 6 Trajectory comparison on the fr3_walk_halfsphere sequence

Table 1 summarizes the quantitative Absolute Trajectory Error (ATE, meters) and average tracking time per frame (milliseconds) for YOLOv5-ORB-SLAM3, ORB-SLAM3, and RTAB-Map. The proposed method achieves the lowest ATE RMSE in *fr3_walk_xyz* (0.013914 m), outperforming ORB-SLAM3 (0.461090 m) and RTAB-Map (0.901539 m). Similar improvements are observed in *fr3_walk_halfsphere* and *fr3_walk_rpy*, where YOLOv5-ORB-SLAM3 consistently achieves reduced trajectory errors and lower variance. Although ORB-SLAM3 offers faster runtime performance, YOLOv5-ORB-SLAM3 delivers substantially higher tracking accuracy in dynamic scenarios due to its effective dynamic-object keypoint filtering.

Table 1 ATE [m], standard deviation [m], and tracking-time [s] comparison for ORB-SLAM3, RTAB-Map, and YOLOv5-ORB-SLAM3

Sequence	RMSE			Standard Deviation			Mean Time		
	ORB SLAM3	RTAB-Map	YOLOv5-ORB-SLAM3	ORB SLAM3	RTAB-Map	YOLOv5-ORB-SLAM3	ORB SLAM3	RTAB-Map	YOLOv5-ORB-SLAM3
Fr1/desk	0.06235	0.034636	0.019062	0.01597	0.01481	0.009834	0.014752	0.1	0.065943
Fr1/room	0.08788	0.034636	0.062805	0.04624	0.01481	0.028654	0.014171	0.1	0.065222
Fr1/plant	0.02116	0.467912	0.017865	0.00823	0.271309	0.008944	0.01423	0.38123	0.069586
Fr2/desk	0.04539	1.680612	0.077554	0.02179	0.84621	0.011884	0.015314	1.45203	0.068339
Fr2/xyz	0.00988	1.682684	0.014842	0.00547	0.881003	0.009464	0.013568	1.43362	0.069452
Fr3/sit_xyz	0.01077	0.012185	0.015564	0.00526	0.005206	0.007134	0.013386	0.36176	0.067417
Fr3/sit_halfsphere	0.02277	0.045543	0.040893	0.01287	0.030745	0.031006	0.014249	0.82	0.068017
Fr3/sit_rpy	0.0219	0.047622	0.036471	0.01385	0.029882	0.025659	0.013022	0.38639	0.064161
Fr3/walk_xyz	0.46109	0.901539	0.01391	0.20493	0.757535	0.007849	0.013319	0.48879	0.076512
Fr3/walk_halfsphere	0.26942	0.876504	0.02461	0.14007	0.740418	0.013213	0.013764	0.46871	0.076322
Fr3/walk_rpy	0.60834	1.06315	0.03033	0.31437	0.815978	0.017794	0.013169	0.68152	0.074993

Table 1 also provides an overview of the three SLAM systems' average processing times per frame. Because of its lightweight pipeline, ORB-SLAM3 achieves the quickest runtime, averaging about 0.014 s per frame (~71 FPS). RTAB-Map has much greater processing times, ranging from 0.1 s to over 1.45 s per frame (~1.9 FPS on average), indicating the computational expense of global map maintenance and loop-closure procedures. YOLOv5-ORB-SLAM3 retains real-time performance despite added cost from real-time YOLOv5 inference, with per-frame durations ranging from 0.064 to 0.077 s (~14 to 15 FPS). Despite this lower frame rate, YOLOv5-ORB-SLAM3 delivers considerably enhanced trajectory accuracy in dynamic circumstances, indicating an appropriate compromise between computational cost and resilience.

Table 2 gives a comparative evaluation of Absolute Trajectory Error (ATE, meters) for three dynamic TUM RGB-D sequences—fr3_walk_xyz, fr3_walk_rpy, and fr3_walk_halfsphere. The proposed YOLOv5-ORB-SLAM3 system is examined with SEG-SLAM [18], YDD-SLAM [7], and Panoptic-SLAM [21] to benchmark localization accuracy under dynamic situations. Across the investigated sequences, the suggested technique yields the lowest ATE values—0.0139 m, 0.0303 m, and 0.0246 m, respectively—indicating improved trajectory accuracy. These results reveal that adding YOLOv5 [22] for dynamic-object filtering greatly boosts pose stability and improves tracking robustness compared with previous YOLO- and segmentation-based SLAM frameworks.

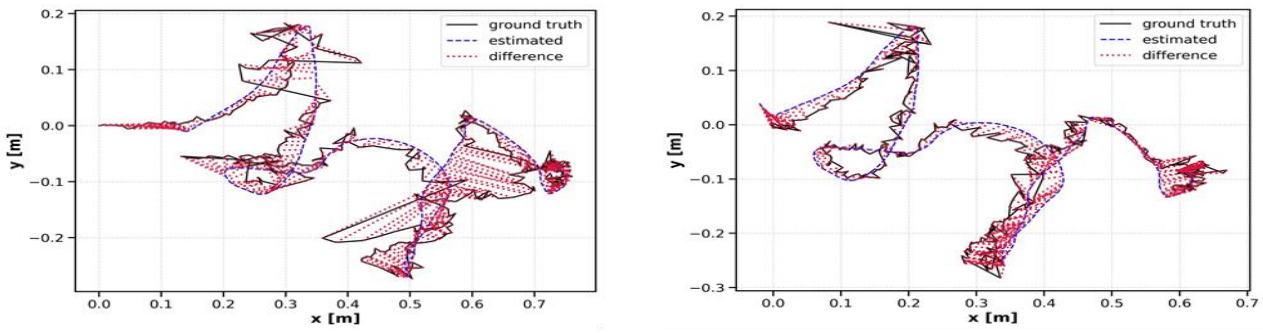
Table 2 Quantitative comparison of ATE RMSE [m] among YOLOv5-ORB-SLAM3, SEG-SLAM, YDD-SLAM, and Panoptic SLAM

Sequence	YOLOv5-ORB-SLAM3	SEG-SLAM [18]	YDD-SLAM [7]	Panoptic SLAM [21]
fr3_walk_xyz	0.01391	0.0141	0.0151	0.014
fr3_walk_rpy	0.03033	0.0306	0.0355	0.032
fr3_walk_halfsphere	0.02461	0.0243	0.0275	0.025

4.2 BONN Dynamic RGB-D Dataset

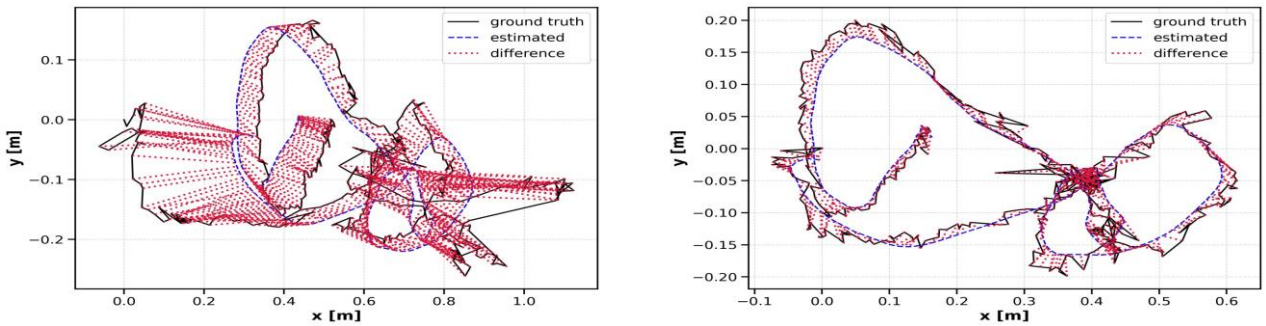
The BONN Dynamic RGB-D Dataset was used to further assess the robustness of YOLOv5-ORB-SLAM3, particularly in scenarios involving both known and previously unseen dynamic objects. The dataset consists of 15 indoor sequences featuring interactions with a balloon, a cardboard box, and a moving person. Notably, the cardboard box is not included in the COCO [23] label set, meaning YOLOv5 does not detect it; this provides an opportunity to evaluate the system's behavior when dynamic elements are not recognized by the segmentation model.

Figs. 7 and 8 show results from the balloon sequences, where both the person and the balloon appear and disappear intermittently. YOLOv5-ORB-SLAM3 consistently produces accurate trajectories, while ORB-SLAM3 fails to maintain stable tracking. Figs. 9 and 10 present the non-obstructing box sequences, in which a person moves a stationary cardboard box within the camera's field of view. ORB-SLAM3 experiences substantial deviations from the ground-truth trajectory due to the influence of undetected dynamic objects. In contrast, YOLOv5-ORB-SLAM3 maintains trajectories closely aligned with ground truth, demonstrating resilience even when dynamic objects are not explicitly recognized by the detection model.



(a) ORB_SLAM3 (b) YOLOv5-ORB-SLAM3

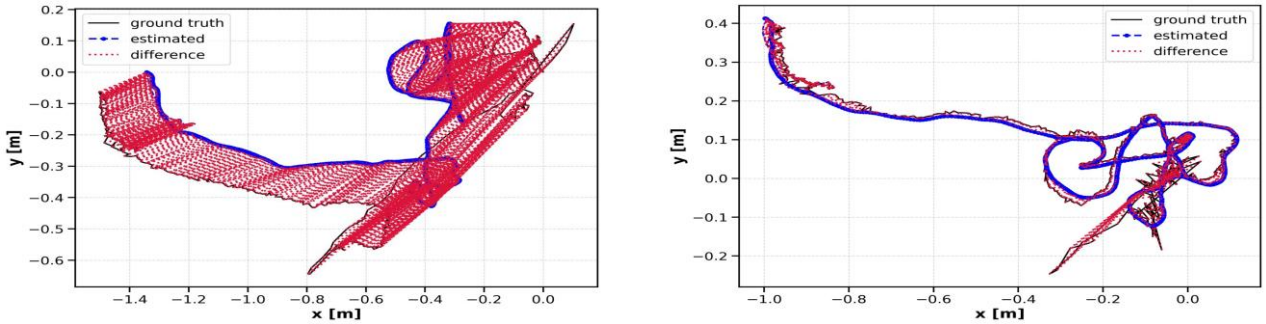
Fig. 7 Trajectory comparison on the Balloon sequence of the BONN dynamic dataset



(a) ORB_SLAM3 (b) YOLOv5-ORB-SLAM3

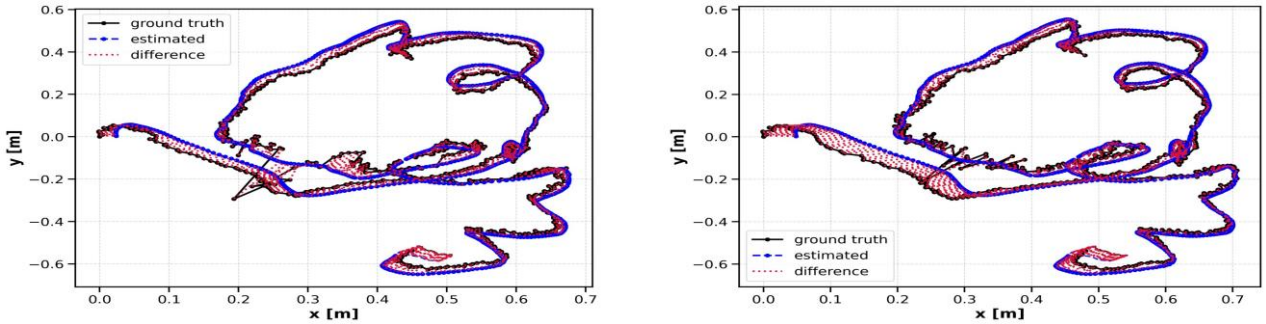
Fig. 8 Trajectory comparison on the Balloon2 sequence of the BONN dynamic dataset

Figs. 11 and 12 present outcomes from person-tracking sequences, where an individual enters the scene and places a package on the ground. YOLOv5-ORB-SLAM3 maintains accurate pose estimation, whereas ORB-SLAM3 produces significant errors. These results demonstrate the effectiveness of the proposed dynamic feature filtering approach in handling moving objects and previously unseen dynamic elements.



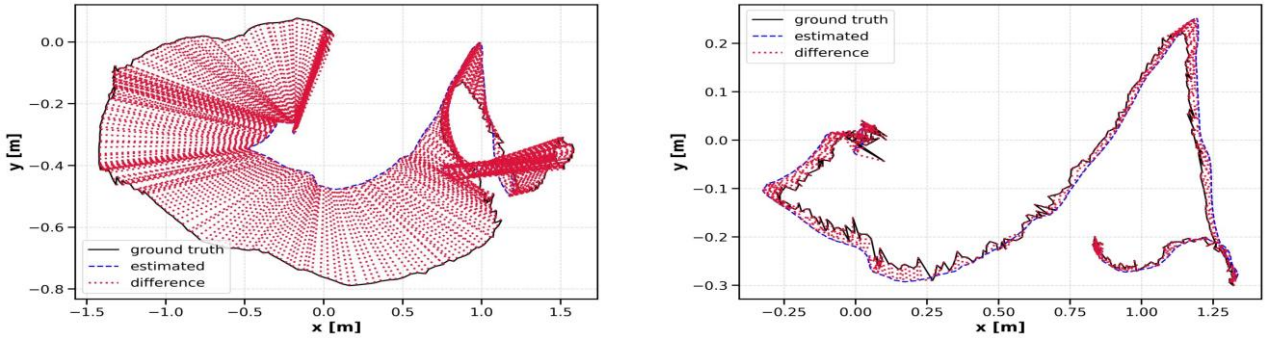
(a) ORB_SLAM3 (b) YOLOv5-ORB-SLAM3

Fig. 9 Trajectory comparison on the moving_nobstr_box



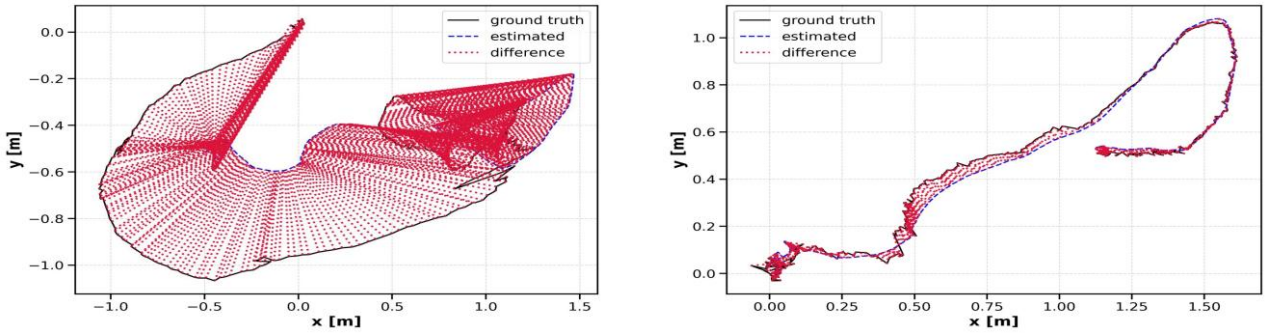
(a) ORB_SLAM3 (b) YOLOv5-ORB-SLAM3

Fig. 10 Trajectory comparison on the moving_nobstr_box2 sequence



(a) ORB_SLAM3 (b) YOLOv5-ORB-SLAM3

Fig. 11 Trajectory comparison on the person tracking sequence



(a) ORB_SLAM3 (b) YOLOv5-ORB-SLAM3

Fig. 12 Trajectory comparison on the person_tracking2 sequence

Table 3 presents the Root Mean Square Error (RMSE, meters) of the Absolute Trajectory Error (ATE) for YOLOv5-ORB-SLAM3, ORB-SLAM3, and RTAB-Map throughout dynamic sequences from the BONN Dynamic RGB-D dataset, which include intense human motion. The suggested YOLOv5-ORB-SLAM3 consistently produces lower ATE values than ORB-SLAM3 across all investigated sequences, indicating improved resilience under dynamic situations. Except for Balloon_track, where RTAB-Map gets a slightly lower RMSE, YOLOv5-ORB-SLAM3 performs better than RTAB-Map in almost all sequences. These results illustrate the usefulness of dynamic-object filtering in increasing trajectory stability and localization accuracy in highly dynamic situations.

Table 3 A comparison of the BONN Dynamic dataset's ATE RMSE [m] and standard deviations [m] for YOLOv5-ORB-SLAM3, ORB-SLAM3, and RTAB-Map

Sequence	RMSE			Standard Deviation		
	ORB SLAM3	RTAB-Map	YOLOv5-ORB-SLAM3	ORB SLAM3	RTAB-Map	YOLOv5-ORB-SLAM3
Balloon	0.05538	0.85889	0.03509	0.03546	0.54039	0.01274
Balloon2	0.09459	0.31692	0.03025	0.03622	0.19126	0.0132
Balloon_track	0.03108	0.02625	0.04194	0.01634	0.01576	0.01528
Balloon_track2	0.02671	0.086	0.03272	0.01137	0.04071	0.01367
Crowd	1.13103	1.95004	0.10162	0.72599	0.72679	0.09224
Crowd2	0.82757	1.7772	0.12648	0.37093	0.75797	0.08886
Crowd3	0.50367	0.65395	0.0363	0.31728	0.27164	0.02163
kidnap_box	0.02537	0.03795	0.02875	0.01289	0.01414	0.01448
kidnap_box2	0.02638	0.02867	0.02591	0.01072	0.01149	0.01103
nonobstr_box	0.28107	0.46332	0.04203	0.08267	0.14203	0.03479
nonobstr_box2	0.03117	0.07552	0.03581	0.01048	0.03246	0.01252
obstr_box	0.57849	0.69738	0.22564	0.15798	0.2167	0.07648
obstr_box2	0.63856	0.77914	0.4133	0.25653	0.40944	0.15745
person_track	0.72089	0.43657	0.07553	0.34019	0.19783	0.07169
person_track2	0.78916	0.38135	0.06064	0.41559	0.1298	0.03457
Average	0.38407	0.57128	0.08747	0.18671	0.24656	0.04471
MIN	0.02537	0.02625	0.02591	0.01048	0.01149	0.01103
MAX	1.13103	1.95004	0.4133	0.72599	0.75797	0.15745

The findings indicate that integrating YOLOv5 with ORB-SLAM3 enhances scene understanding through semantic annotation of 3D maps. ORB-SLAM3 maintained reasonable accuracy in controlled dynamic environments, but performance decreased slightly in scenes with frequent occlusions or significant motion, as observed in the BONN Dynamic RGB-D dataset. While YOLOv5 provided precise object detections, the additional computational overhead reduced real-time performance compared to standard ORB-SLAM3.

The semantic fusion process enabled effective differentiation between dynamic and static objects, highlighting potential avenues for future work in dynamic object filtering. A noted challenge was the misclassification of overlapping or partially visible objects, which affected the consistency of the semantic map. Employing depth-aware segmentation or temporal consistency checks could mitigate this issue.

The integration of object detection with SLAM demonstrates clear advantages, particularly for applications that require interaction with the environment, such as robotic navigation or augmented reality. In addition to evaluations on the BONN and TUM RGB-D datasets, YOLOv5-ORB-SLAM3 was tested on two custom datasets recorded in a residential environment using an Intel RealSense D435i RGB-D camera, as illustrated in Fig. 13.

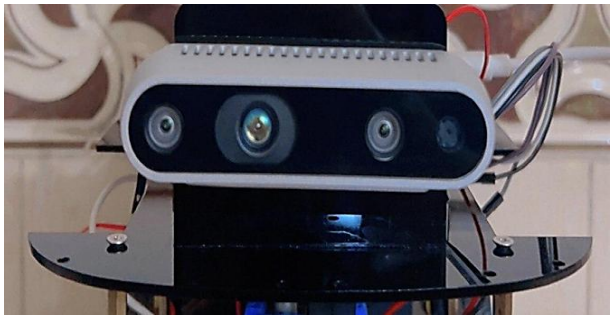


Fig. 13 The Intel RealSense D435i depth camera

The objective of this experiment was to create a dynamic sequence featuring people moving throughout the environment while interacting with various objects. As a motion capture system was unavailable, the camera remained stationary during the recordings. Fig. 14 presents the results obtained using RTAB-Map from two frames of the first sequence. In Fig. 14(a), one individual is walking while carrying a potted plant, while another is walking with a suitcase. In Fig. 14(b), the first person continues moving, whereas the second has placed the suitcase on the ground and is moving alongside it. The images demonstrate that RTAB-Map incorporates features from both static and dynamic elements in the scene when estimating the camera pose, which may introduce errors in dynamic environments.

(a) Result 1 of RTAB-Map on the custom dataset

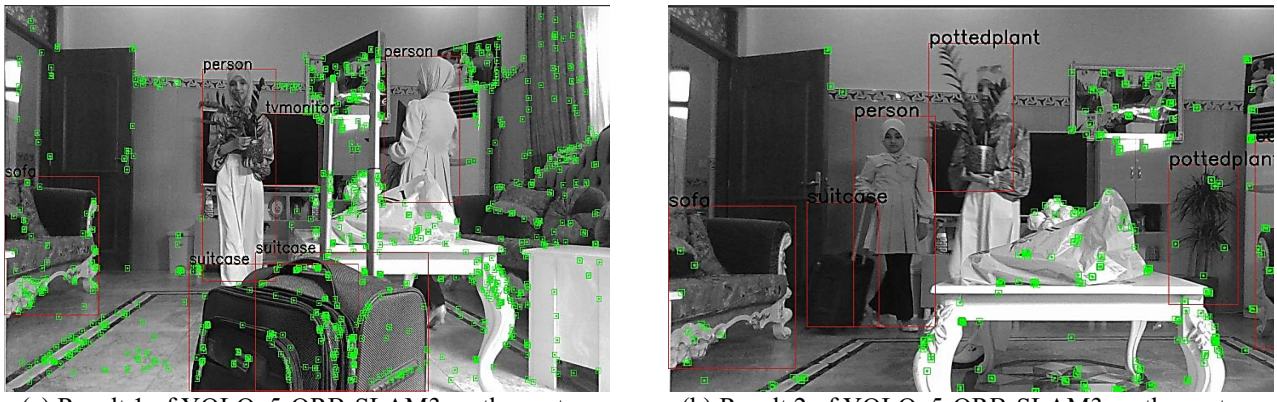
(b) Result 2 of RTAB-Map on the custom dataset

Fig. 14 RTAB-Map results on the first experiment dataset sequence

Fig. 15 presents the results of YOLOv5-ORB-SLAM3 for the same dynamic sequence. Figs. 15(a) and 15(b) depict the scenarios previously described. Although the potted plant was incorrectly classified by the YOLO model—mistakenly labeled as a pair of scissors—the system nonetheless successfully treated it as a dynamic object, filtering out its keypoints during pose estimation.

A further observation concerns the suitcase carried by one of the individuals. While in motion, the suitcase's keypoints were correctly excluded due to its dynamic classification. Once the suitcase became stationary, its keypoints were retained and utilized in the pose estimation process. This demonstrates the system's capability to dynamically adapt to changes in object motion, effectively improving trajectory estimation even when semantic labels are partially incorrect or missing.

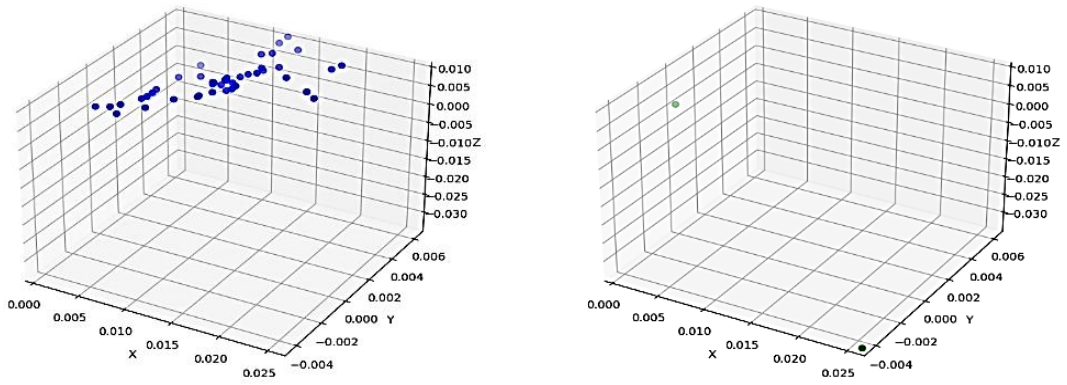
Fig. 16 presents a comparison of camera pose estimation between YOLOv5-ORB-SLAM3 and RTAB-Map. YOLOv5-ORB-SLAM3 accurately estimated the camera trajectory, with only a few minor outliers. In contrast, RTAB-Map produced more dispersed trajectory estimates, exhibiting larger error margins and a greater number of outliers compared to YOLOv5-ORB-SLAM3. Although both systems were generally able to estimate the camera pose near the origin for this dataset, the results demonstrate that YOLOv5-ORB-SLAM3 offers superior accuracy and robustness in dynamic environments.



(a) Result 1 of YOLOv5-ORB-SLAM3 on the custom dataset

(b) Result 2 of YOLOv5-ORB-SLAM3 on the custom dataset

Fig. 15 The output of the YOLOv5-ORB-SLAM3 system on the custom dataset



(a) YOLOv5-ORB-SLAM3

(b) RTAB-Map

Fig. 16 Camera pose estimation using YOLOv5-ORB-SLAM3 and RTAB-Map in the first experiment

In the second experiment, multiple individuals moved unpredictably while carrying a suitcase and a potted plant. This dataset, illustrated in Fig. 17, represents a highly dynamic environment characterized by frequent occlusions, interactions with objects, and people entering and exiting the scene. Such conditions create significant challenges for camera pose estimation, as dynamic elements continuously introduce potential errors in feature correspondence and trajectory computation.

(a) A person carrying a suitcase

(b) A person carrying a potted plant

(c) Another person carrying a potted plant and a suitcase together

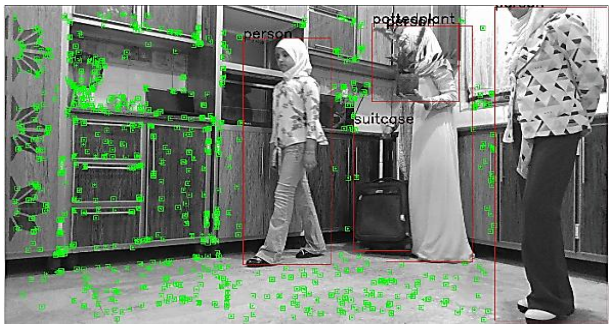
Fig. 17 Second experiment dataset sequence

The objective of this experiment was to evaluate the capability of YOLOv5-ORB-SLAM3 to handle previously unseen dynamic objects. At the start of the sequence, a potted plant remained stationary on a table. As shown in Fig. 18(a), YOLOv5 correctly detected the plant and its associated keypoints, which were then used by YOLOv5-ORB-SLAM3 for camera pose estimation. For comparison, Fig. 18(b) shows the results obtained using RTAB-Map under the same conditions; lacking a semantic detection module, RTAB-Map does not exclude dynamic keypoints.

In a later frame, the plant is picked up by another individual. Fig. 18(c) illustrates how YOLOv5-ORB-SLAM3 excludes the dynamic keypoints associated with the moving object, maintaining accurate pose estimation. Conversely, Fig. 18(d) shows that RTAB-Map continues to use keypoints from both the moving object and the person, resulting in tracking instability. Finally, Fig. 18(e) demonstrates the sustained robustness of YOLOv5-ORB-SLAM3 in dynamic scenes, whereas Fig. 18(f) reveals that RTAB-Map suffers from increased pose estimation error and noticeable map drift due to the inclusion of dynamic features.

(a) Frame with a static plant and a laptop YOLOv5-ORB-SLAM3

(b) Frame with static plant and laptop RTAB-Map



(c) Frame with moving plant using YOLOv5-ORB-SLAM3

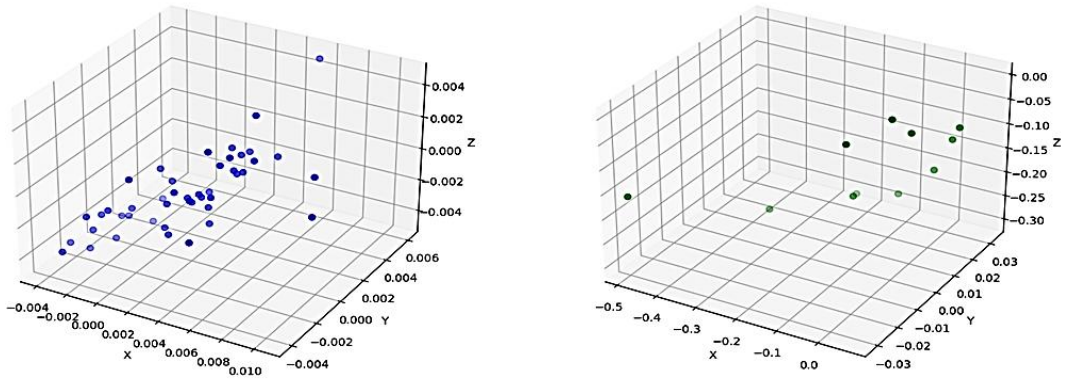
(d) Frame with moving plant RTAB-Map

(e) YOLOv5-ORB-SLAM3 with a static plant again

(f) RTAB-Map with a static plant again

Fig. 18 Unknown moving object filter in the second experiment

Fig. 19 presents the estimated camera trajectories from both YOLOv5-ORB-SLAM3 and RTAB-Map in this highly dynamic environment. Table 4 summarizes trajectory statistics, including average translation error and standard deviation in millimeters, with the best-performing values highlighted in bold. The results confirm that integrating YOLOv5 with ORB-SLAM3 significantly enhances robustness to dynamic objects by filtering them before feature association, enabling accurate camera tracking and minimal drift throughout the sequence.



(a) YOLOv5-ORB-SLAM3 (b) RTAB-Map

Fig. 19 Camera pose estimation using YOLOv5-ORB-SLAM3 and RTAB-Map in the second experiment

Table 4 Statistical evaluation of camera pose accuracy [mm]

Statistic	YOLOv5-ORB-SLAM3	RTAB-Map
X-mean	1.993	-79.196
Y-mean	-0.406	-621.715
Z-mean	-0.804	-7.354
X-std	3.320	76.162
Y-std	1.715	699.409
Z-std	2.142	69.744

In contrast, the YOLOv5-ORB-SLAM3 system demonstrates greater robustness in dynamic environments compared to RTAB-Map; however, it can exhibit occasional inconsistencies in filtering moving objects. To mitigate this, a temporal consistency buffer can be employed to track object classification over multiple frames, reducing false positives and misclassifications. Despite these challenges, YOLOv5-ORB-SLAM3 maintains a stable trajectory, with mean errors close to zero and lower standard deviations, whereas RTAB-Map exhibits larger drift and increased instability in dynamic scenes.

5. Conclusions

This study suggests a semantic-aware visual SLAM system that combines ORB-SLAM3 with YOLOv5-based real-time object identification to enhance mapping and localization capabilities in dynamic situations. Detecting and eliminating moving objects from the SLAM pipeline improves the system's resilience to dynamic disturbances that are frequently seen in real-world situations. In-depth studies on publicly available benchmark datasets, such as TUM RGB-D and BONN RGB-D Dynamic, as well as practical trials utilizing an Intel RealSense D435i camera, are used to verify the method. Comparative analyses are conducted against ORB-SLAM3 and RTAB-Map. The following is a summary of the primary conclusions and important findings:

(1) The proposed YOLOv5-ORB-SLAM3 system effectively integrates real-time object detection into the ORB-SLAM3 pipeline, enabling reliable operation in dynamic environments by identifying and excluding moving objects from the SLAM process.

(2) By filtering dynamic features before feature matching, the system significantly reduces the influence of moving objects, resulting in improved pose estimation accuracy and enhanced map consistency compared to traditional SLAM approaches.

(3) Extensive evaluations on TUM RGB-D and BONN Dynamic datasets, as well as real-world experiments using an Intel RealSense D435i camera, demonstrate that YOLOv5-ORB-SLAM3 consistently outperforms ORB-SLAM3 and RTAB-Map in terms of accuracy and robustness.

(4) The proposed method maintains stable and continuous trajectories in complex and crowded scenes, where ORB-SLAM3 often incorporates dynamic key points, and RTAB-Map lacks semantic awareness.

(5) The integration of semantic information allows YOLOv5-ORB-SLAM3 to handle unknown or unlabeled moving objects better, improving localization reliability under challenging real-world conditions.

(6) Overall, the experimental results confirm that incorporating semantic object detection into visual SLAM significantly enhances performance in environments with unavoidable dynamic interactions.

Despite its advantages, the system is limited by the predefined object classes supported by YOLOv5. Future work may focus on developing adaptive or class-agnostic models capable of detecting arbitrary dynamic regions, as well as incorporating temporal consistency or motion-based cues to further improve robustness. These enhancements would extend the applicability of the proposed system to a broader range of real-world autonomous robotics scenarios.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] A. Barrau and S. Bonnabel, "The Invariant Extended Kalman Filter as a Stable Observer," IEEE Transactions on Automatic Control, vol. 62, no. 4, pp. 1797-1812, 2017.

[2] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "FastSLAM: An Efficient Solution to the Simultaneous Localization and Mapping Problem with Unknown Data," Journal of Machine Learning Research, vol. 4, no. 3, pp. 1-44, 2004.

[3] S. Thrun and M. Montemerlo, "The GraphSLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures," International Journal of Robotics Research, vol. 25, no. 5-6, pp. 403-429, 2006.

[4] L. Chen, G. Li, W. Xie, J. Tan, Y. Li, J. Pu, et al., "A Survey of Computer Vision Detection, Visual SLAM Algorithms, and their Applications in Energy-Efficient Autonomous Systems," Energies, vol. 17, no. 20, article no. 5177, 2024.

[5] X. Zhang, H. Dong, H. Zhang, X. Zhu, S. Li, and B. Deng, "A Real-time, Robust, and Versatile Visual-SLAM Framework Based on Deep Learning Networks," IEEE Transactions on Instrumentation and Measurement, vol. 74, pp. 1-13, 2025.

[6] S. Song, H. Lim, A. J. Lee, and H. Myung, "DynaVINS: A Visual-Inertial SLAM for Dynamic Environments," IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 11523-11530, 2022.

[7] P. Cong, J. Liu, J. Li, Y. Xiao, X. Chen, X. Feng, et al., "YDD-SLAM: Indoor Dynamic Visual SLAM Fusing YOLOv5 with Depth Information," Sensors, vol. 23, no. 23, article no. 9592, 2023.

[8] J. Li and J. Luo, "YS-SLAM: YOLACT++ Based Semantic Visual SLAM for Autonomous Adaptation to Dynamic Environments of Mobile Robots," Complex & Intelligent Systems, vol. 10, no. 4, pp. 5771-5792, 2024.

[9] M. Chen, H. Guo, R. Qian, G. Gong, and H. Cheng, "Visual Simultaneous Localization and Mapping (vSLAM) Algorithm Based on Improved Vision Transformer Semantic Segmentation in Dynamic Scenes," Mechanical Sciences, vol. 15, no. 1, pp. 1-16, 2024.

[10] C. Xu, E. Bonetto, and A. Ahmad, "DynaPix SLAM: A Pixel-Based Dynamic Visual SLAM Approach," Proceedings of the 46th DAGM German Conference on Pattern Recognition (DAGM GCPR 2024), Part II, Springer-Verlag, pp. 168-184, 2023.

[11] A. Eslamian and M. R. Ahmadzadeh, "Det-SLAM: A Semantic Visual SLAM for Highly Dynamic Scenes using Detectron2," Proceedings of the 8th International Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), IEEE Press, pp. 1-5, 2022.

[12] M. Labbé and F. Michaud, "RTAB-Map as an Open-Source Lidar and Visual Simultaneous Localization and Mapping Library for Large-Scale and Long-Term Online Operation," Journal of Field Robotics, vol. 36, no. 2, pp. 416-446, 2019.

[13] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An Evaluation of the RGB-D SLAM System," Proceedings of the IEEE International Conference on Robotics and Automation, Saint Paul, Minnesota, USA, pp. 1691-1696, 2012.

[14] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss, "ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals," Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), Macau, China, pp. 7855-7862, 2019.

[15] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras,” IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255-1262, 2017.

[16] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, “DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes,” IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 4076-4083, 2018.

[17] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A Versatile and Accurate Monocular SLAM System,” IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, 2015.

[18] P. Cong, J. Li, J. Liu, Y. Xiao, and X. Zhang, “SEG-SLAM: Dynamic Indoor RGB-D Visual SLAM Integrating Geometric and YOLOv5-Based Semantic Information,” Sensors, vol. 24, no. 7, article no. 2102, 2024.

[19] D. Feng, Z. Yin, X. Wang, F. Zhang, and Z. Wang, “YLS-SLAM: A Real-time Dynamic Visual SLAM based on Semantic Segmentation,” Industrial Robot: The International Journal of Robotics Research and Application, vol. 52, no. 1, pp. 106-115, 2024.

[20] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the Evaluation of RGB-D SLAM Systems,” Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, pp. 573-580, 2012.

[21] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM,” IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874-1890, 2021.

[22] Ultralytics, "YOLOv5: in PyTorch > ONNX > CoreML > TFLite," https://github.com/ultralytics/yolov5, accessed in 2025.

[23] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, et al., “Microsoft COCO: Common Objects in Context,” Lecture Notes in Computer Science, vol. 8693, pp. 740-755, 2014.