International Journal of Network Dynamics and Intelligence Scilight

Article

Enhancing Visual SLAM Localization Accuracy through Dynamic Object Detection and Adaptive Feature Filtering

Zhang Qiang * and Wang Tao

School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China
* Correspondence: lucky_zhangqiang@163.com

Received: 8 May 2025
Accepted: 2 July 2025
Published: 18 September 2025

Abstract: To address the critical challenge of localization accuracy degradation in visual Simultaneous Localization and Mapping (SLAM) systems, primarily caused by dynamic feature point interference in complex environments, we propose an advanced visual SLAM framework that integrates deep learningbased object detection with an optimized feature-point filtering strategy. The proposed methodology follows a two-stage processing pipeline. First, a YOLOv5-based object detection module accurately identifies and segments dynamic objects in the operational environment. Second, a probabilistic dynamic feature-point elimination mechanism refines localization precision by selectively retaining reliable static features. To validate the framework, we conduct comprehensive experiments using datasets collected from an Automated Guided Vehicle (AGV) system operating in port environments. Comparative results demonstrate that our approach achieves a 50% improvement in localization accuracy over the conventional ORB-SLAM3 framework in dynamic scenarios. These findings not only confirm the effectiveness of our method but also provide valuable insights for developing robust SLAM systems in real-world engineering applications.

Keywords: object detection; dynamic feature points; YOLOv5; localization; visual simultaneous localization and mapping

1. Introduction

Visual Simultaneous Localization and Mapping (SLAM) is a computational technique that enables camera-equipped mobile platforms to extract environmental information, construct maps, detect and cluster objects, percept emotion as well as estimating their pose in unknown environments [1–3]. SLAM systems are broadly categorized into monocular and stereo SLAM based on the number of camera lenses [4]. While monocular SLAM offers advantages such as a simple architecture and wide field of view, it suffers from inherent limitations. Most notably, it cannot determine the true scale of environmental features from image data alone. Instead, it relies on camera motion-induced parallax and epipolar constraints to estimate depth and enable localization. However, this approach becomes susceptible to significant feature-point matching errors in dynamic environments, where moving objects degrade localization accuracy. To mitigate dynamic object interference, current research employs epipolar constraints to validate feature-point matches [5]. Existing methods for eliminating invalid feature points fall into four categories: (1) Graph- based methods [6]: Effective but computationally intensive, making them unsuitable for real-time applications [7, 8]. (2) Block theory-based methods [9]: Work well locally but lack global scalability. (3) Geometric resampling methods [10]: Use random sample consensus (RANSAC) to select static feature points from a coarse-matched subset. While effective, their performance degrades as dynamic features increase due to rising computational complexity. (4) Deep learning-based methods [11]: Emerging as a promising alternative.

The rise of deep learning has revolutionized how we approach environmental perception in robotics [3, 12]. Among its many applications, one particularly promising direction has been the integration of object detection into visual SLAM systems. By leveraging neural networks to identify and classify objects complete with precise bounding boxes, researchers have unlocked new ways to enhance mapping and localization in dynamic environments [13, 14].

This fusion of deep learning and SLAM has led to several breakthroughs [15]. For example, the work of [16], tackled one of SLAM's trickiest challenges: handling motion-blurred, dynamic indoor scenes. Their solution introduced an inertial sensors with deep learning, embedded directly into ORB-SLAM2's front end. The result was a system that could intelligently separate true environmental features from misleading dynamic points with unprecedented reliability. Meanwhile, It should be Ref. [17] took a different approach. They armed their monocular SLAM system with DeeplabV3+'s semantic segmentation prowess, teaching it to recognize moving objects like pedestrians or vehicles at a glance. But they didn't stop there each detected object then faced rigorous motion consistency checks, ensuring only the most trustworthy features survived for localization. Furthermore, Ref. [18] presented a lightweight object detection network for real-time performance. Their NarkNet19-YOLOv3 trimmed the information from traditional object detection, offering SLAM systems both speed and semantic understanding.

These advances herald a new era of SLAM systems that transcend basic environmental mapping to achieve true scene understanding. With the fast development of deep learning technologies, the distinction between robotic perception and comprehension continues to blur, particularly in dynamic navigation scenarios.

For visual SLAM systems, two fundamental challenges remain paramount: (1) precise camera trajectory estimation from image sequences, and (2) accurate 3D environmental reconstruction [19]. Successful implementation depends on efficient extraction of discriminative image features that capture local environmental characteristics. These features enable both global localization and map construction through 3D coordinate calculation. However, current visual SLAM methods exhibit notable limitations in dynamic environments, where moving objects frequently degrade localization and mapping accuracy. While recent deep learning enhanced approaches show promise, their practical adoption is often hindered by excessive computational demands and specialized hardware requirements. Addressing these limitations, our work presents two key innovations:

(1) Hybrid SLAM Architecture: We seamlessly integrate YOLOv5's efficient object detection with classical SLAM algorithms, enabling a) precise dynamic object identification and b) probabilistic feature point filtering. This combination accelerates inter frame feature matching while significantly improving localization accuracy.

(2) Real World Validation: Through extensive testing using operational data from our custom port AGV platform, we demonstrate the method's effectiveness in authentic industrial settings. The results not only validate our approach but also provide actionable insights for developing intelligent port infrastructure.

The remainder of this article is organized as follows: Section 2 outlines the system's overall framework and details the task execution process. Section 3 elaborates on the enhanced algorithms for object detection and feature point filtering. Section 4 is dedicated to experimental validation and the analysis of results. Finally, Section 5 concludes the entire study.

2. System Framework and Workflow of the Improved Algorithm

Within the realm of classical visual SLAM architectures, ORB-SLAM3 represents a significant advancement over its predecessors. It supports various camera configurations, including monocular, stereo, and RGB D cameras, and is the world's first SLAM framework to integrate vision, visualinertial odometry, and multi map capabilities. Furthermore, this framework utilizes temporally correlated data to suppress zero drift and provides inter map data correlations to achieve precise localization and mapping. The overall structure of the framework is illustrated in Figure 1. Building upon the framework depicted in Figure 1, this study integrates a YOLOv5 module into the front-end image frame tracking thread to identify dynamic feature points. The workflow of the improved visual SLAM algorithm is outlined as follows: First, feature points are extracted from the image data captured by the camera. Concurrently, the YOLOv5 object detection network identifies objects within the scene and transmits real time target classifications and bounding box information to the SLAM system. Upon receiving this information, the SLAM system evaluates the target classifications. If the targets are identified as dynamic, such as pedestrians or moving vehicles, the feature points within the corresponding bounding boxes are removed. To ensure a sufficient number of feature points for subsequent processing, the algorithm verifies the count offeature points in the current frame. If the number of feature points exceeds a predefined threshold, the initialization process proceeds. Subsequently, by applying appropriate thresholds, suspicious feature points are filtered out, enabling the matching of static feature points and the estimation of the initial pose for target points. It is noteworthy that the backend of the system requires nonlinear optimization to accomplish loop closure detection and mapping tasks.

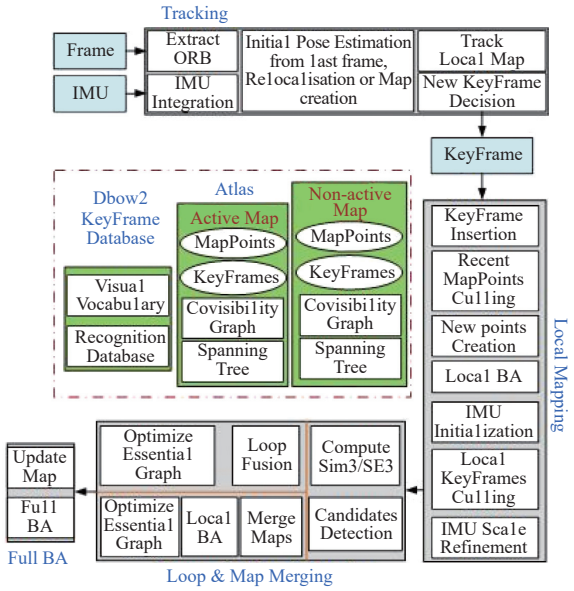

Figure 1. Block Diagram of the ORB-SLAM3 System.

3. Object Detection and Dynamic Feature Point Filtering

3.1. Improvements in Object Detection Algorithm

Currently, deep learning-based object detection methods can be categorized into those represented by YOLO [20] and those by Mask R-CNN [21, 22]. Mask R-CNN has a frame processing time of over 2 s on a CPU, making it challenging to ensure real-time performance. In contrast, the improved YOLOv5 object detection network meets real-time system requirements due to its minimal model parameters and the lowest floating-point operations within the object detection framework, resulting in the fastest execution speed. YOLOv5 is a classic algorithm in the YOLO series, noted for its lightweight design and higher accuracy compared to YOLOv4. This significantly enhances both the efficiency and accuracy of object detection.

In the object detection process, inference time and nonmaximum suppression (NMS) [23, 24] processing are timeconsuming stages, indirectly affecting the localization accuracy of the SLAM system. Given that NMS processing time is related to bounding box regression, this paper focuses on improving the inference time to enhance the speed of object detection. Excluding the input section, this paper divides YOLOv5 into three parts: the backbone network, the PANet network, and the output, as illustrated in Figure 2.

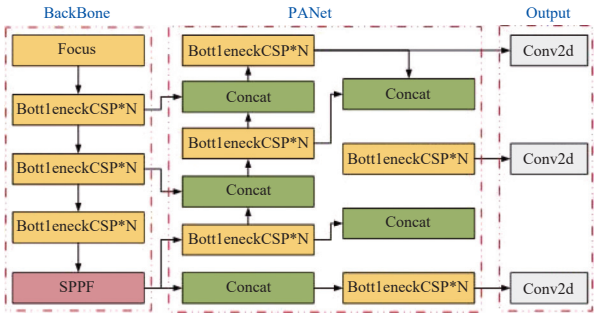

Figure 2. Improved YOLOv5 System Framework.

In the system framework shown in Figure 2, the backbone network is responsible for extracting useful features from the input images. It consists of multiple Bottleneck Cross-Stage Partial Network (CSPN) modules, which help reduce the computational burden and improve the efficiency of feature extraction. Additionally, it incorporates a focus module, which is designed to adjust the dimensions and the number of channels of the input data. The Path Aggregation Network (PANet) serves to strengthen feature integration by aggregating features from different levels through both bottom-up and top-down pathways. It consists of several Bottleneck CSPN modules and Concatenation operations to merge features from various hierarchical levels. This structure facilitates the model's improved understanding of contextual information within images, thereby enhancing detection accuracy. The Output represents

the final stage of the model, tasked with generating detection results. It includes multiple Conv2d layers that transform the integrated features into the final output format. Furthermore, the Conv2d layers in the output layer are typically employed to produce information such as bounding boxes and class probabilities. The entire model extracts features through the Backbone, integrates features via PANet, and ultimately gen- erates the final detection results through the output layer.

Based on the system framework shown in Figure 2, reducing the inference time is mainly achieved from the following aspects: First, the Spatial Pyramid Pooling-Fast (SPPF) module captures features at different scales through fast spatial pyramid pooling. This helps reduce the dependence on specific scales, thereby enhancing the network's detection ability for targets of different sizes and simultaneously reducing the computational burden. Second, The PANet aggregates features at different levels through bottom-up and top-down paths. This architecture enables the model to better understand the contextual information in the images, thus improving the effectiveness of feature fusion and reducing unnecessary computations. Additionally, in the PANet, feature maps at different levels are merged through concatenation operations. This allows for the effective utilization of features at different scales while reducing the size of the feature maps, thereby decreasing the computational load of subsequent layers. Third, the Conv2d layers in the output layer are optimized to minimize the computational amount while maintaining sufficient resolution to generate accurate detection results. Through the above three design points, the network can significantly reduce the inference time while maintaining high detection accuracy, thereby improving the speed of object detection. This is particularly crucial for application scenarios that require real time processing.

3.2. Dynamic Feature Point Filtering Algorithm

This paper initially employs YOLO to detect potential dynamic objects within the scene. Upon detecting these dynamic objects, the algorithm uses dynamic probability to differentiate between dynamic and static feature points on the objects. For targets that cannot be recognized by the neural network, a motion consistency algorithm based on geometric constraints is applied for filtering. The pseudocode for the dynamic feature point filtering algorithm is presented in Algorithm 1. Specifically, after YOLOv5 processes the current frame Fnow, the sets of target bounding box coordinates Bcoo, target categories Bcla, and confidence scores Bcom are transmitted in real-time to the SLAM system via a Socket. Within the SLAM system, the confidence scores Bcom are first used to determine whether to utilize the target information, retaining only those with confidence scores greater than threshold T1. Next, the target categories Bcla are evaluated to identify dynamic targets, such as pedestrians or moving vehicles. If a dynamic target category is identified, the feature points within the corresponding bounding box coordinates Bcoo are removed. Finally, the number of feature points Nnow in the current frame is checked; if Nnow exceeds the threshold T2, the system retains the current frame information for pose estimation.

Algorithm 1 Dynamic Feature Filtering Algorithm

Input: Current frame Fnow, historical frame Flast, number of feature points in current frame Nnow, thresholds T1 and T2
Output: Image frame with static features retained

```
begin
for Fnow in Flast do
Extract feature points from Fnow to SLAM thread
Perform object detection on Fnow to YOLOv5 thread
Transfer detection results to SLAM thread
if Bcon ≥ T1 then
continue
end if
if Bcla = "person"or"car"or"animal" etc then
delete
end if
if Nnow > T2 then
continue
end if
return Fsta
end for
end
```

4. Experimental Validation and Analysis

4.1. Description of the Experimental Environment

The algorithm is executed on a Lenovo laptop equipped with an Intel i7 - 11700 processor, running Ubuntu 20.04 and utilizing ROS noetic. Data is collected from a port experimental platform featuring dynamic targets, categorized

into four scenarios: vehicles with people, vehicles without people, no vehicles with people, and no vehicles without people. Images from the platform are annotated using object detection tools and classified into categories such as "person" and "car", forming a dataset in the TUM format. This dataset is divided into training, validation, and test sets in an 8:1:1 ratio, resulting in 3200 images for training and 400 images each for validation and testing. The training set is then used for training, with the resulting models applied to experiments on the experimental platform.

4.2. Object Detection and Feature Point Elimination

Traditional visual SLAM systems face challenges in defining dynamic objects from planar pixels and in detecting and tracking these dynamic objects. To address these issues, this study evaluates the performance of ORBSLAM3 and its optimized versions. As shown in Figure 3, the feature extraction using ORB-SLAM3 reveals that ORB feature points are often disorganized, significantly impacting SLAM mapping. These points not only increase redundant computations but also affect mapping speed and can cause drift. Figure 4 illustrates the dynamic object detection using the YOLOv5 method, which effectively eliminates dynamic feature points.

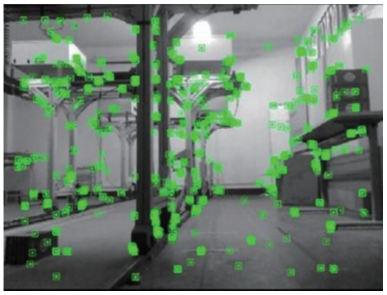

Figure 3. Feature Extraction in ORB-SLAM3.

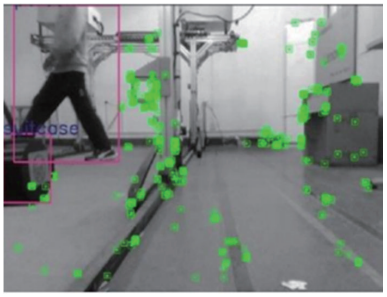

Figure 4. Object Detection and Feature Extraction in the Improved Algorithm.

The comparison between Figure 3 and Figure 4 indicates that the improved visual SLAM algorithm proposed in this paper not only effectively identifies dynamic objects in the environment but also accurately eliminates invalid dynamic feature points. This algorithm demonstrates excellent adaptability to varying environments.

4.3. Localization Experiment

The experimental platform records the dataset using an RGB-D camera mounted on a mobile AGV operating at a speed of 0.3 m/s. The dataset captures a dynamic scenario featuring pedestrians passing through a simulated port operation area. In this study, the trajectory map generated by running the dataset with the artography algorithm serves as reference data.

Figure 5 and Figure 6 illustrates the trajectory and localization accuracy of the ORB-SLAM3 algorithm in a scenario with dynamic individuals. The data reveals that ORB-SLAM3 operates stably for the first 120 seconds, maintaining a local- ization error within 0.1 m. However, after 120 seconds, the localization error increases significantly, reaching a peak of over 0.2 m at 140 seconds. This increase is attributed to the introduction of dynamic human feature points by the visual system around the 140-second mark, causing substantial data drift. This observation underscores the significant impact of dynamic feature points on localization accuracy.

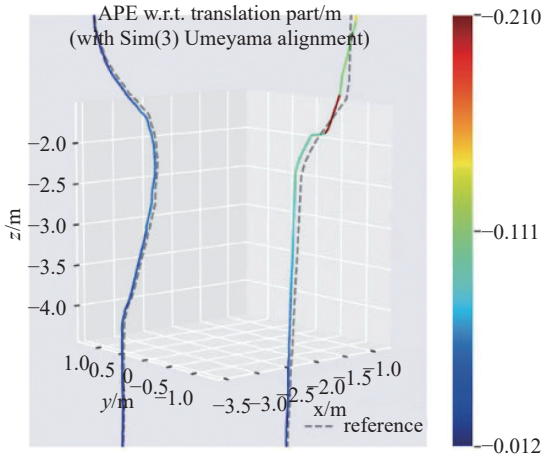

Figure 5. The Trajectory of ORB-SLAM3.

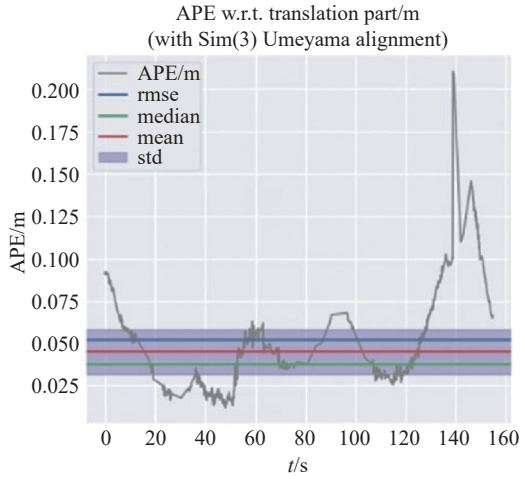

Figure 6. The Pose Error of ORB-SLAM3.

Conversely, Figure 7 and Figure 8 shows the localization performance using the proposed algorithm under the same experimental conditions. The data indicates that the system operates stably throughout the experiment, with a maximum localization error consistently within 0.1 m. Moreover, there is no noticeable data jitter or drift, demonstrating that the proposed algorithm framework maintains high localization accuracy in dynamic environments.

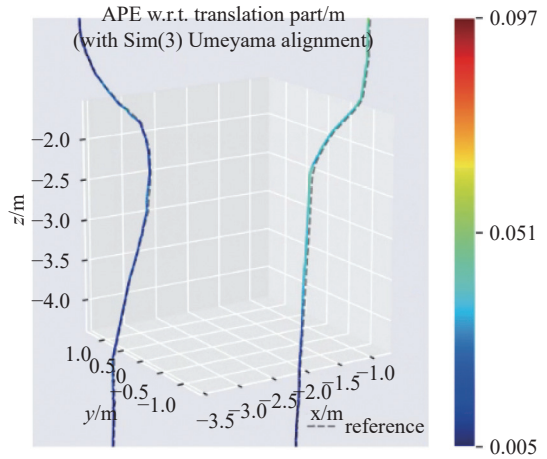

Figure 7. The Trajectory of the Improved Algorithm.

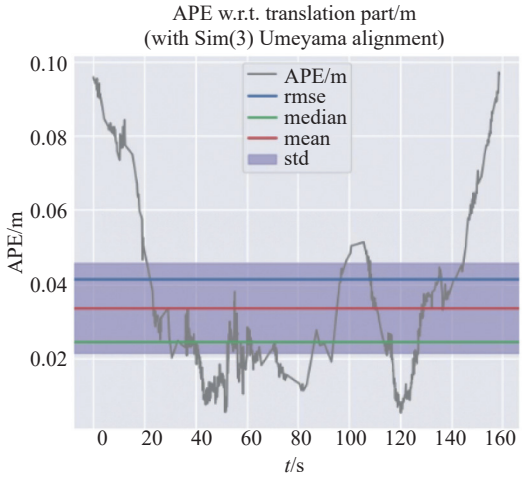

Figure 8. The Pose Error of Improved Algorithm.

4.4. Environmental Adaptability Experiment

To comprehensively assess the environmental adaptability of the proposed algorithm framework, comparative tests are conducted under varying AGV traveling speeds and environmental light intensities. Firstly, the operating speeds of the AGV are set to 0.2 m/s, 0.3 m/s, and 0.4 m/s to evaluate its positioning error. The corresponding results are presented in Figure 9 to Figure 14.

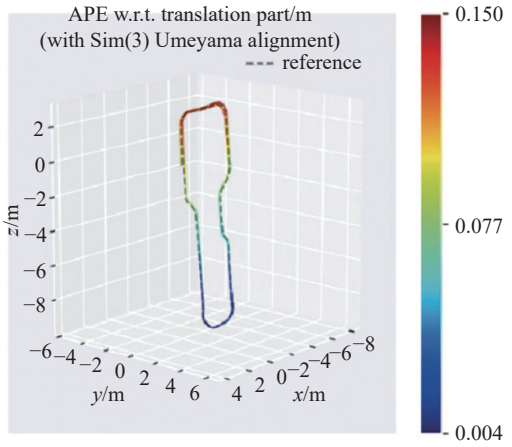

Figure 9. The Trajectory of the Improved Algorithm (0.2 m/s).

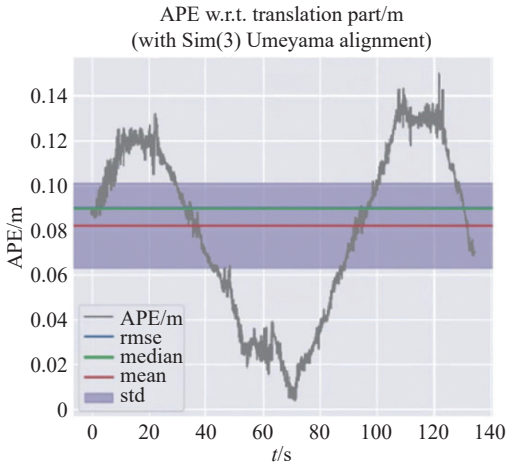

Figure 10. The Pose Error of Improved Algorithm (0.2 m/s).

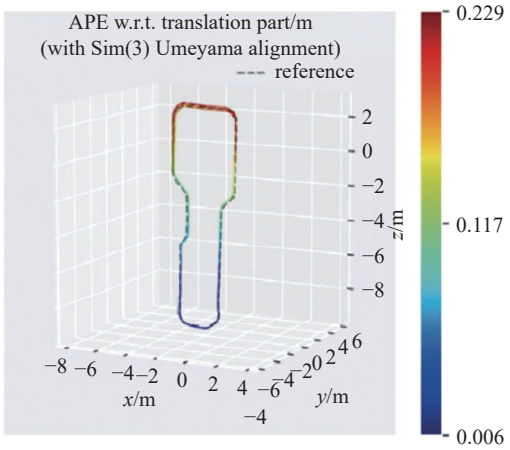

Figure 11. The Trajectory of the Improved Algorithm (0.3 m/s).

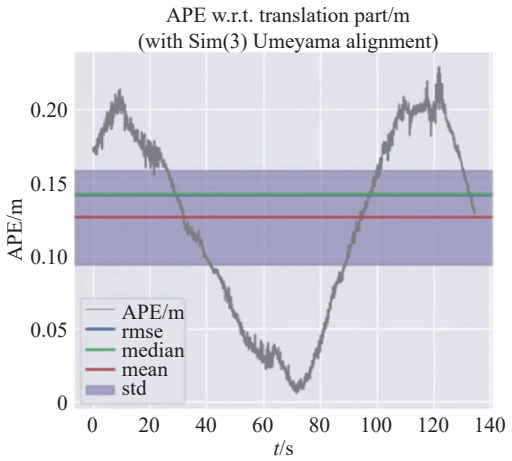

Figure 12. The Pose Error of Improved Algorithm (0.3 m/s).

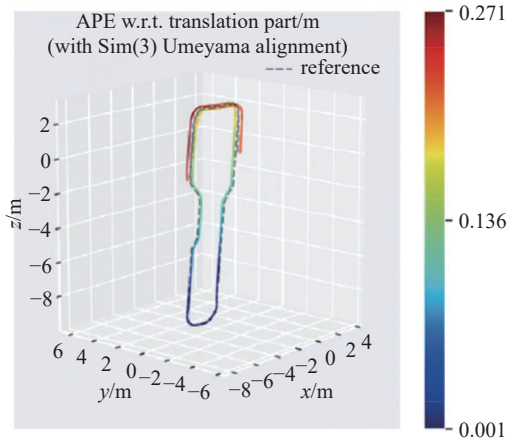

Figure 13. The Trajectory of the Improved Algorithm (0.4 m/s).

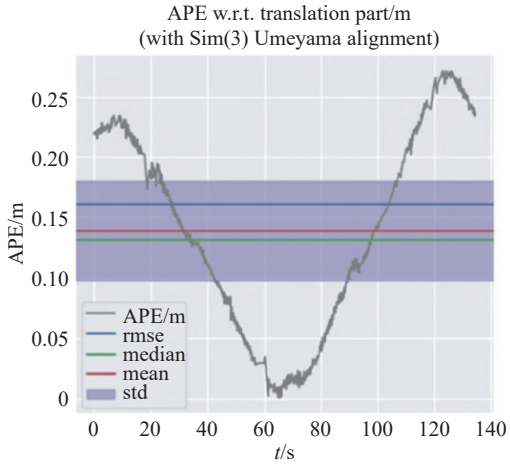

Figure 14. The Pose Error of Improved Algorithm (0.4 m/s).

The visual data demonstrate a correlation between increased AGV speed and a decrease in system positioning accuracy, as shown in Figures 9 to 14. Nonetheless, the system's positioning accuracy remains stably convergent across different speeds, fulfilling the predetermined performance standards. For instance, at a speed of 0.4 m/s, the maximum absolute error of the positioning system increases by 0.12 m compared to the scenario at 0.2 m/s. This error is primarily observed in turning sections and in the presence of dynamic objects. The observed variations in error can be attributed to two main factors: (1) The AGV used in this study is subject to non-holonomic constraints, which prevent lateral movement, thus making certain desired poses unattainable during turns and resulting in significant pose errors. (2) The presence of dynamic objects in the environment can interfere with the positioning system, affecting accuracy. Therefore, it is necessary to reasonably filter out unnecessary dynamic feature points.

The key statistical results from the error analysis at different speeds are presented in Table 1. The data in the table further confirm the negative correlation between AGV speed and system positioning accuracy, while also validating the system's overall stability. Consequently, it is crucial to set the AGV's speed reasonably according to specific application requirements to ensure optimal performance.

Table 1 Positioning error at different agv speeds

Speed (m/s)	RMSE (m)	Mean (m)	Max (m)
0.2	0.090297	0.082059	0.150030
0.3	0.141744	0.126117	0.228530
0.4	0.161515	0.139260	0.271395

The assessment of illumination effects is conducted under two distinct conditions: a brightly lit scenario with all eight experimental lights activated and a dimly lit scenario with only two lights on. These test environments are depicted in Figure 15 and Figure 16. The trajectory and positioning error resulting from the algorithm introduced in this study are illustrated in Figure 17 and Figure 18.

As depicted in Figure 17 and Figure 18, light intensity exerts a notable influence on the positioning accuracy of the algorithm presented in this study. For instance, under conditions of diminished lighting, the system's positioning accuracy is comparatively reduced. Nevertheless, the positioning error consistently converges within a confined range. Consider the standard deviation of the positioning system as an illustrative metric; even under suboptimal lighting conditions, it remains stably within an error margin of 0.04 m. This finding underscores the robust adaptability of the proposed algorithm to environments with fluctuating light levels.

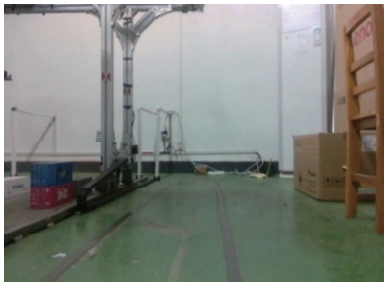

Figure 15. Strong Light Illumination Environment.

Figure 16. Weak Light Illumination Environment.

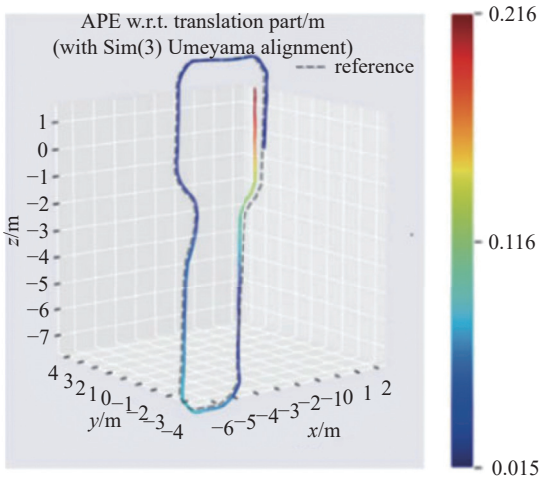

Figure 17. Trajectory Comparison Under Different Lighting Conditions.

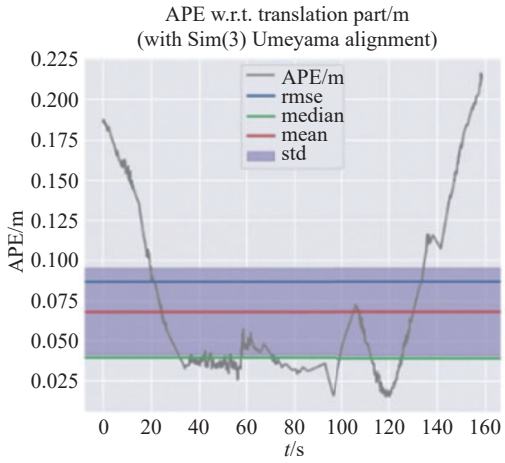

Figure 18. Comparison of Positioning Errors Under Different Lighting Conditions.

5. Conclusion

In visual SLAM systems, dynamic feature points in complex environments can interfere with localization accuracy. To address this issue, we have developed an advanced visual SLAM framework that integrates deep learning-based object detection and optimized feature-point filtering through a two-stage process. Our experiments demonstrate a 50% improvement in localization accuracy compared to the conventional ORB-SLAM3 in dynamic scenarios. For instance, in a busy port like environment, the average localization error of our method is reduced by 0.1m compared to ORB-SLAM3. This can enhance the practical performance of visual SLAM systems, such as improving AGV navigation in port logistics and mobile device positioning in intelligent navigation.

However, the method has some limitations. For example, the YOLOv5-based object detection module may be inaccurate for highly occluded or small dynamic objects. Specifically, a 50% occlusion can reduce segmentation

accuracy by approximately 20%. Additionally, the assumptions of the probabilistic feature point elimination mechanism may not hold in complex scenes with fast moving objects and lighting changes, potentially increasing the localization error by 0.05m.

For future research, we plan to explore integrating other detection algorithms, such as Mask R-CNN, and multi sensor fusion to enhance object detection. We will also optimize the assumptions of the feature point elimination mechanism using machine learning techniques. Additionally, we aim to apply this technology in autonomous driving and smart home service robots.

Author Contributions: Qiang Zhang: investigation, resources, writing–review & editing; **Tao wang:** supervision and writing–original draft. All authorshave read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abaspur Kazerouni, I.; Fitzgerald, L.; Dooly, G.; *et al*. A survey of state-of-the-art on visual SLAM. Expert Syst. Appl., **2022**, *205*: 117734. doi: 10.1016/j.eswa.2022.117734
2. Sahili, A.R.; Hassan, S.; Sakhrieh, S.M.; *et al*. A survey of visual SLAM methods. IEEE Access, **2023**, *11*: 139643−139677. doi: 10.1109/ACCESS.2023.3341489
3. Yan, X.Q.; Mao, Y.Q.; Ye, Y.D.; *et al*. Cross-modal clustering with deep correlated information bottleneck method. IEEE Trans. Neural Netw. Learn. Syst., **2024**, *35*: 13508−13522. doi: 10.1109/TNNLS.2023.3269789
4. Xie, T.; Sun, Q.H.; Sun, T.; *et al*. DVDS: A deep visual dynamic slam system. Expert Syst. Appl., **2025**, *260*: 125438. doi: 10.1016/j.eswa.2024.125438
5. He, J.M.; Li, M.R.; Wang, Y.Y.; *et al*. OVD-SLAM: An online visual SLAM for dynamic environments. IEEE Sens. J., **2023**, *23*: 13210−13219. doi: 10.1109/JSEN.2023.3270534
6. Hu, W.; Gao, X.; Cheung, G.; *et al*. Feature graph learning for 3D point cloud denoising. IEEE Trans. Signal Process., **2020**, *68*: 2841−2856. doi: 10.1109/TSP.2020.2978617
7. Wen, S.H.; Li, X.F.; Liu, X.; *et al*. Dynamic SLAM: A visual SLAM in outdoor dynamic scenes. IEEE Trans. Instrum. Meas., **2023**, *72*: 5028911. doi: 10.1109/TIM.2023.3317378
8. Sun, H.L.; Fan, Q.W.; Zhang, H.Q.; *et al*. A real-time visual SLAM based on semantic information and geometric information in dynamic environment. J. Real-Time Image Process., **2024**, *21*: 169. doi: 10.1007/s11554-024-01527-4
9. Yi, J.Z.; Chen, A.B.; Cai, Z.X.; *et al*. Facial expression recognition of intercepted video sequences based on feature point movement trend and feature block texture variation. Appl. Soft Comput., **2019**, *82*: 105540. doi: 10.1016/j.asoc.2019.105540
10. Lv, C.L.; Lin, W.S.; Zhao, B.Q. Intrinsic and isotropic resampling for 3D point clouds. IEEE Trans. Pattern Anal. Mach. Intell., **2022**, *45*: 3274−3291. doi: 10.1109/TPAMI.2022.3185644
11. Lee, O.; Joo, K.; Sim, J.Y. Learning-based reflection-aware virtual point removal for large-scale 3D point clouds. IEEE Robot. Autom. Lett., **2023**, *8*: 8510−8517. doi: 10.1109/LRA.2023.3329365
12. Wang, J.W.; Zhuang, Y.; Liu, Y.S. FSS-NET: A fast search structure for 3D point clouds in deep learning. Int. J. Netw. Dyn. Intell., **2023**, *2*: 100005. doi: 10.53941/ijndi.2023.100005
13. Li, R.H.; Wang, S.; Gu, D.B. Ongoing evolution of visual SLAM from geometry to deep learning: Challenges and opportunities. Cognit. Comput., **2018**, *10*: 875−889. doi: 10.1007/s12559-018-9591-8
14. Qin, Y.; Yu, H.D. A review of visual SLAM with dynamic objects. Ind. Rob., **2023**, *50*: 1000−1010. doi: 10.1108/IR-07-2023-0162
15. Li, W.Y.; Yang, F.W. Information fusion over network dynamics with unknown correlations: An overview. Int. J. Netw. Dyn. Intell., **2023**, *2*: 100003. doi: 10.53941/ijndi0201003
16. Liu, F.Y.; Cao, Y.; Cheng, X.H.; *et al*. A visual SLAM method assisted by IMU and deep learning in indoor dynamic blurred scenes. Meas. Sci. Technol., **2024**, *35*: 025105. doi: 10.1088/1361-6501/ad03b9
17. Zhang, X.Y.; Abd Rahman, A.H.; Qamar, F. Semantic visual simultaneous localization and mapping (SLAM) using deep learning for dynamic scenes. PeerJ Comput. Sci., **2023**, *9*: e1628. doi: 10.7717/peerj-cs.1628
18. Wu, W.X.; Guo, L.; Gao, H.L.; *et al*. YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint. Neural Comput. Appl., **2022**, *34*: 6011−6026. doi: 10.1007/s00521-021-06764-3
19. Pu, H.Y.; Luo, J.; Wang, G.; *et al*. Visual SLAM integration with semantic segmentation and deep learning: A review. IEEE Sen. J., **2023**, *23*: 22119−22138. doi: 10.1109/JSEN.2023.3306371
20. Iqra; Giri, K.J. SO-YOLOV8: A novel deep learning-based approach for small object detection with yolo beyond coco. Expert Syst. Appl., **2025**, *280*: 127447. doi: 10.1016/j.eswa.2025.127447
21. Yuan, Y.J.; Li, Y.; Fang, X.X.; *et al*. Automatic velocity picking based on improved mask R-CNN. IEEE Trans. Geosci. Remote Sens., **2023**, *61*: 5923312. doi: 10.1109/TGRS.2023.3335250
22. He, K.M.; Gkioxari, G.; Dollár, P.; *et al*. Mask R-CNN. In *Proceedings of 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017*; IEEE: New York, 2017, pp. 2980–2988. doi:10.1109/ICCV.2017.322
23. Guo, C.S.; Cai, M.; Ying, N.; *et al*. ANMS: Attention-based non-maximum suppression. Multimed. Tools Appl., **2022**, *81*: 11205−11219. doi: 10.1007/s11042-022-12142-5
24. Tang, X. S.; Xie, X.L.; Hao, K.R.; *et al*. A line-segment-based non-maximum suppression method for accurate object detection. Knowl.-Based Syst., **2022**, *251*: 108885. doi: 10.1016/j.knosys.2022.108885

Citation: Qiang, Z.; W. Tao. Enhancing Visual SLAM Localization Accuracy through Dynamic Object Detection and Adaptive Feature Filtering. *International Journal of Network Dynamics and Intelligence*. 2025, 4(3), 100018. doi: 10.53941/ijndi.2025.100018