Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.058944

ARTICLE

MG-SLAM: RGB-D SLAM Based on Semantic Segmentation for Dynamic Environment in the Internet of Vehicles

Fengju Zhang[1] **and Kai Zhu**[2,*]

[1]School of Mechanical Engineering, Jiangsu University of Technology, Changzhou, 213001, China

[2]School of Automobile and Traffic Engineering, Jiangsu University of Technology, Changzhou, 213001, China

*Corresponding Author: Kai Zhu. Email: fatkyo@jsut.edu.cn

Received: 24 September 2024 Accepted: 15 November 2024 Published: 17 February 2025

ABSTRACT

The Internet of Vehicles (IoV) has become an important direction in the field of intelligent transportation, in which vehicle positioning is a crucial part. SLAM (Simultaneous Localization and Mapping) technology plays a crucial role in vehicle localization and navigation. Traditional Simultaneous Localization and Mapping (SLAM) systems are designed for use in static environments, and they can result in poor performance in terms of accuracy and robustness when used in dynamic environments where objects are in constant movement. To address this issue, a new real-time visual SLAM system called MG-SLAM has been developed. Based on ORB-SLAM2, MG-SLAM incorporates a dynamic target detection process that enables the detection of both known and unknown moving objects. In this process, a separate semantic segmentation thread is required to segment dynamic target instances, and the Mask R-CNN algorithm is applied on the Graphics Processing Unit (GPU) to accelerate segmentation. To reduce computational cost, only key frames are segmented to identify known dynamic objects. Additionally, a multi-view geometry method is adopted to detect unknown moving objects. The results demonstrate that MG-SLAM achieves higher precision, with an improvement from 0.2730 m to 0.0135 m in precision. Moreover, the processing time required by MG-SLAM is significantly reduced compared to other dynamic scene SLAM algorithms, which illustrates its efficacy in locating objects in dynamic scenes.

KEYWORDS

Visual SLAM; dynamic scene; semantic segmentation; GPU acceleration; key segmentation frame

1 Introduction

The rapid advancement of deep learning and the Internet of Vehicles (IoV) has resulted in increasingly sophisticated autonomous driving technology. In intricate transportation settings, the capacity to precisely and efficiently determine the positions of nearby vehicles is essential for maintaining traffic flow and efficiency [1–3]. In recent years, visual SLAM has garnered significant attention due to the useful information stored in images, which can be extracted for various vision-based applications. In the study of Ali et al. [4], SLAM technology plays a pivotal role in the Internet of Vehicles (IoV), equipping intelligent vehicles with the ability to navigate uncharted territory autonomously. By leveraging data from various sensors and integrating with mapping information, SLAM provides

Copyright © 2025 The Authors. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

precise vehicle localization and environmental modeling-crucial for path planning and obstacle avoidance in self-driving cars. Furthermore, SLAM can be combined with other navigation systems like GPS to enhance the accuracy of vehicle positioning, particularly in urban environments where GPS signals may be weak or inaccessible. This synergy ensures that autonomous vehicles maintain reliable navigation even in challenging conditions. Nowadays, a well-developed framework of visual SLAM systems has emerged. Including ORB-SLAM2 [5] and LSD-SLAM [6], have achieved satisfactory results. However, most of these methods are intended for use in static environments. In reality, there is plenty of interference from moving objects, which significantly affects their performance. When in motion, people, vehicles, and other objects cause severe interference with pose estimation and map reconstruction for the camera. In this respect, various robust estimation techniques, such as RANSAC, can remove outliers. However, their effectiveness is limited when processing highly dynamic scenes, and failure may occur when moving objects are in the majority in the camera's view. This explains why a real-time visual SLAM system, namely MG-SLAM, has been proposed. Through a dynamic target detection process based on semantic segmentation and the Mask R-CNN algorithm, MG-SLAM can detect both known and unknown moving objects. This approach outperforms other dynamic scene SLAM algorithms in terms of precision and processing time, demonstrating its effectiveness in locating objects in dynamic environments.

With the constant progress in deep learning, semantic information has entered people's field of vision and SLAM systems about semantics have emerged, such as semantic SLAM [7,8] and object-level SLAM [9,10]. The semantic information in the environment is extracted by semantic segmentation of image frames, where the labels of detected objects are predicted and masks are generated, which enables the identification and removal of potential dynamic targets. This integration has shown a great potential in improving the performance of visual SLAM. However, there are still two major challenges to address for these approaches. Firstly, most deep neural networks such as Mask R-CNN [11], which are used for semantic segmentation, incur high computational cost and are slow in segmentation, resulting in their poor real-time performance, despite the high accuracy of tracking. Although lightweight networks can be used to reduce these issues, the accuracy of segmentation may be compromised, which affects the accuracy of tracking. Secondly, these approaches are applicable only to process the known objects included in the training dataset of the network. Consequently, unknown moving objects that are not included in it cannot be detected, which affects their overall effectiveness in dynamic environments. To address these challenges, innovative solutions have been developed, such as the MG-SLAM system. By leveraging dynamic target detection processes and multi-view geometry methods, MG-SLAM can detect and process both known and unknown moving objects in real time. By addressing these limitations, MG-SLAM represents a significant progress made in the development of robust visual SLAM solutions under dynamic scenarios.

To address the issue of neural networks taking excessive time to segment dynamic objects, which leads to the slow tracking by the camera and its poor real-time performance, graphics processing unit (GPU) have been developed. Some research focuses on comparatively evaluating the performance of computer vision algorithms executed in parallel on the CPU. In the study of Chaple et al. [12], the performances of image convolution on the GPU, FPGA, and CPU were compared. In the study of Li et al. [13], a similar comparison was conducted, the focus of which was on image convolution processing on GPU and FPGA platforms. These studies collectively highlight the superior real-time performance of GPU-based image processing. To achieve real-time performance in dynamic environments, this study proposes segmenting potential dynamic objects using GPU-accelerated neural networks, which mitigates the impact on the tracking processes. To further reduce computational costs, a semantic SLAM algorithm based on key segmentation frames is developed to solve the problem of the long time required for SLAM localization in dynamic environment. Moreover, although known dynamic objects can be effectively processed using neural networks, unknown dynamic objects, which are not predefined as dynamic, can still cause interference with tracking and positioning. To address this issue, this study introduces a geometric module based on multi-view geometric constraints. The primary contributions of this paper are a comprehensive solution proposed for dealing with dynamic environments in real-time visual SLAM systems. By applying GPU-accelerated neural networks, semantic SLAM algorithms, and geometric constraints, this methodology bolsters the precision and stability of SLAM systems within dynamic settings. The main contributions of this study are as follows:

1. A GPU-accelerated dynamic object segmentation module is introduced to effectively identify and segment dynamic targets within complex environments. This module significantly enhances the system's overall positioning precision and stability in dynamic settings. By leveraging the Mask R-CNN instance segmentation network on GPU hardware, the segmentation of image frames is significantly optimized, which ensures an excellent real-time performance of the system.
2. A key frame segmentation selection method is proposed, which does not require the segmentation of each frame when a deep learning network is applied to segment the images. Only the selected key frame is segmented. This method can not only reduce the frequency of semantic segmentation but also accelerate its speed.
3. The tracking process involves dynamic feature point removal. In this step, the integration of a semantic segmentation network coupled with multi-view geometric approaches facilitates the elimination of dynamic objects within actual environments, thereby enhancing the precision of positioning within such dynamic settings.

The remainder of this paper is structured as follows: Section 2 is the Related Work on SLAM. In Section 3, the framework of this system is elaborated. In Section 4, a comparative analysis of the system is conducted using a standardized dataset. In Section 5, the experimental performances are summarized and analyzed briefly.

2 Related Works

Considering numerous challenges encountered by visual SLAM in dynamic environment, several approaches have been proposed by scholars to address these challenges. In recent years, various methods of visual SLAM have emerged. In general, these approaches can be categorized into two groups: geometric-based dynamic Visual SLAM and semantic- and deep learning-based dynamic visual SLAM.

2.1 Dynamic VSLAM Based on Geometric Theory

The core idea of the geometry-based approach intended to process dynamic objects is that these objects are treated as outliers to remove them. Sun et al. [14] proposed a method of motion removal, where particle filtering was performed to track and filter the moving patches in images. The difference in intensity between successive RGB images was calculated to remove dynamic obstacles. Tan et al. [15] proposed a new method of online key frame representation and updating for the adaptive modeling of dynamic environments. Even in some challenging environments, the attitude of the camera can be accurately estimated using a novel prior-based adaptive RANSAC algorithm for effective removal of the outliers. Kim et al. [16] adopted a non-parametric background model to estimate self-motion based solely on the background model estimation. Zhang et al. [17] used dense optical flow residuals to perform dynamic region detection and applied a framework similar to StaticFusion for reconstruction of the static background. More recently, Dai et al. [18] proposed to differentiate between static and dynamic map points through point correlations. Qin et al. [19] used the relationship between optical flow and objects to identify objects that were moving but predefined as static, and determined whether objects were really moving through geometric constraints. Zhang et al. [20] used the particle filter method to segment non-prior moving objects and estimate and track inter-frame changes by combining optical flow and motion equations established by Gaussian distribution. Long et al. [21] proposed a method to address the indoor flat environment by extracting the plane from the flat region, using different motions to separate the dynamic flat rigid bodies, and independently tracking them.

While geometric-based VSLAM methods can mitigate the impact of dynamic targets to a certain degree, they encounter certain limitations: they fail to detect potential dynamic targets that are temporarily stationary, and they lack semantic information, which is essential for utilizing prior scene knowledge to identify moving objects.

2.2 Dynamic VSLAM Based on Semantic Information

With the rapid progress in deep learning technology and hardware, an increasing number of studies have been conducted to integrate deep learning networks into SLAM systems, such as those for object detection. This is aimed at enhancing the cognitive abilities of robots in their surrounding environments. The approach that leverages learning utilizes deep neural networks to obtain semantic information from the surroundings. This method enables the identification of potential dynamic objects without the need to process numerous image frames. In CNN-SLAM [22], a straightforward method is applied to estimate the pose of the camera, CNN is applied to depth estimation, and the semantic segmentation of images is performed. The depth prediction based on CNN is combined with the directly calculated depth measurement based on monocular SLAM to create a map carrying semantic information. Kaneko et al. [23] introduced a framework under which the masks generated by semantic segmentation networks were used to remove unstable feature points. Additionally, for the improvement of accuracy, some learning-based methods have been proposed, where traditional geometric methods serve to distinguish between static and dynamic objects. For instance, the DS-SLAM proposed by Yu et al. [24] integrated SegNet network and optical flow to identify the moving individuals. A dense semantic octree graph suitable for advanced tasks was generated, despite a lower positioning accuracy during tracking. DynaSLAM [25], which is built on ORBSLAM2, leverages Mask R-CNN [11] to segment each image frame, with a semantic mask obtained. By combining geometric and semantic information, it filters out dynamic feature points. Nevertheless, in monocular and stereoscopic scenarios, these methods are not applicable to extract feature points from the masked regions, when they are assumed to be completely dynamic. Moreover, semantic segmentation incurs considerable computational and time cost, thus leading to sub-optimal real-time performance. Xu et al. proposed OD-SLAM [26], where the SegNet [27] network was applied to extract dynamic object-related information within the scene and filter dynamic features based on re-projection error. However, recognition failure may occur due to inaccurate semantic segmentation. Zhong et al. proposed Detect-SLAM [28], a system that merges target detection techniques with SLAM technology. By utilizing semantic information, it eliminates dynamic target feature points, facilitating the tracking and identification of specific targets. Liu et al. proposed RDS-SLAM [29], a system capable of real-time tracking and mapping in dynamic settings, utilizing a semantic thread and a non-blocking model. While it tackles the difficulties posed by variable-speed semantic segmentation methods, its positioning accuracy still falls short of optimal levels. Peng et al. proposed RO-SLAM [30], which uses an improved PSPNet to detect moving objects in the scene. A filling method based on GANs is proposed, which can fill the defective parts caused by the removal of dynamic objects. Wu et al. [31] improved the YOLOv3 neural network to make the network lightweight and combine with RANSAC to detect dynamic objects. Cheng et al. [32] improved the ORB-SLAM2 system by adding object detection for obtaining semantic information and fusing geometric information to detect moving objects. Jin et al. [33] proposed a SLAM system for indoor environments that combines lightweight segmentation networks and polar constraints to detect dynamic feature points. Chen et al. [34] proposed a real-time SLAM method combining deep learning and dynamic probabilistic strategies, which improves the real-time performance. Liu et al. [35] combined target detection and lightweight feature detection to reduce the cost of calculation and improve the efficiency of the system.

3 System Description

3.1 Framework of MG-SLAM

In practice, there are two key factors to consider for evaluating autonomous robots: the precision of camera position and the reliability of operation in dynamic environments. ORB-SLAM2 performs well in most cases. For this reason, MG-SLAM adopts ORB-SLAM2 as its fundamental framework. To eliminate dynamic objects from the environment based on the input RGB-D image, MG-SLAM utilizes semantic information and geometric techniques. For robotic applications that require real-time performance, MG-SLAM employs a GPU-accelerated semantic segmentation network to segment the images. Also, it is proposed to extract semantic information exclusively from key frames. Unlike the traditional ORB-SLAM2 framework, MG-SLAM includes two additional modules: a semantic segmentation module and a dynamic feature point removal module.

1. A separate semantic segmentation thread and a Mask R-CNN segmentation network are used to segment the key segmentation frames. In this way, the semantic labels of images and the mask of dynamic targets can be obtained.
2. Dynamic feature removal module: This module employs the Mask R-CNN network for image frame segmentation to derive dynamic object masks. These masks, in conjunction with multi-view geometric techniques, are utilized to eliminate previously identified dynamic targets and the associated feature points. Additionally, it removes feature points from regions identified as non-prior dynamic targets by the geometric method.

This system is illustrated in Fig. 1. Firstly, the image frames are input into the system, and a key segmentation frame is determined for each image frame. Under the system, ORB feature points are extracted and multi-view geometric detection is performed. When the key segmentation frame becomes available, semantic segmentation using GPU acceleration is performed, and semantic segmentation is performed by Mask R-CNN. After the segmentation results are obtained, dynamic feature points are identified and eliminated based on the semantic information from the segmented image, as well as through multi-view geometric detection.

3.2 Semantic Segmentation

During the tracking process, numerous dynamic objects may emerge in the camera view, such as pedestrians. When a feature point exists in a dynamic object, data association errors could arise. To prevent such a situation, the optimal approach is to identify the moving objects and subsequently eliminate the feature points associated with them. ORB SLAM2 is not suitable for processing dynamic objects. The solution is a separate object segmentation thread based on Mask R-CNN. Mask R-CNN [7] network is capable of detecting and segment existing dynamic information in the scene, representing the latest technology of object instance segmentation. It can be used to determine the location and category of dynamic targets, making pixel-level predictions. For the detection of dynamic objects, it is proposed to implement mask R-CNN based on TensorFlow, obtain instance labels. Finally, static and dynamic targets are marked during the segmentation process. The Mask R-CNN training set, which utilizes the COCO dataset, encompasses a total of 80 object categories. Assuming that, in the context of most environments, including potentially all dynamic targets, and if additional classes are required, training is performed using the new dataset. When segmentation is performed, the image frame is first input, and then the dynamic objects are segmented using the Mask R-CNN to obtain the segmentation masks. In the frame, the feature points located in the segmented regions are discarded, assuming that the region is dynamic. In an indoor scenario, people are predefined as dynamic objects. As depicted in Fig. 2, Fig. 2a illustrates the graph retaining the dynamic feature points, whereas Fig. 2b presents the graph with the dynamic feature points eliminated.

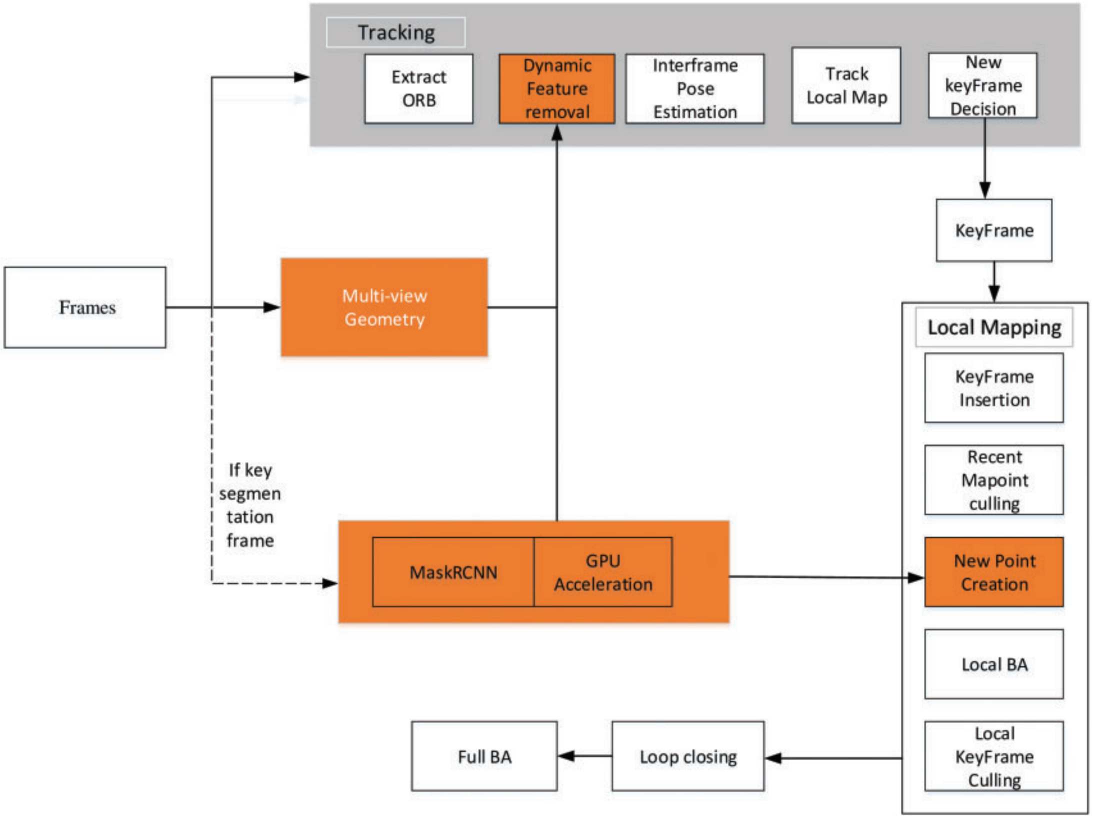


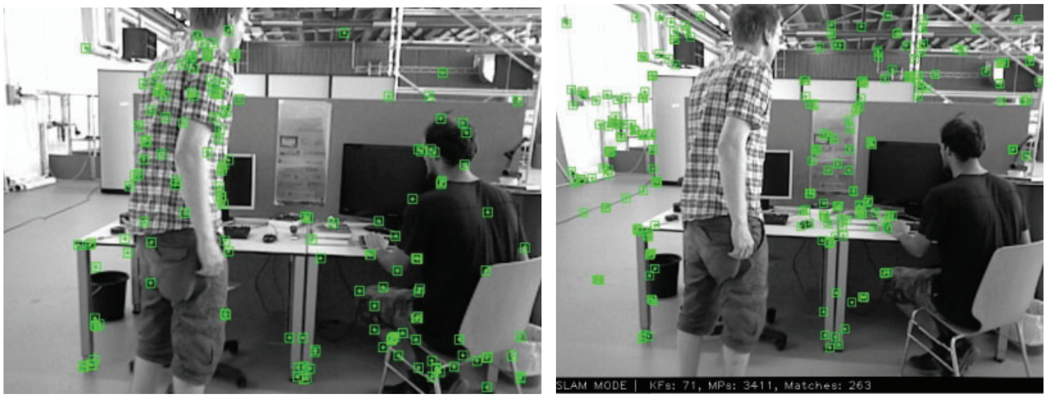
Figure 1: The architecture of our system

Different from other types of semantic and learn-based dynamic SLAM, it is proposed in this study to perform semantic segmentation only when key segmentation frames become available, rather than each new frame, which reduces the computational cost of semantic segmentation significantly. Also, it is proposed to accelerate the segmentation process of the semantic segmentation network through GPU, which facilitates real-time semantic information tracking.

3.3 Key Segmentation Frame Selection

Most of the existing dynamic SLAM methods based on learning or semantics can be used to semantically segment all incoming image frames, However, the running time are long and the speed are low, which results in only being able to run in offline mode. To solve this problem, it is proposed to extract semantic information only from the key segmentation frame. When utilizing the segmentation network for image frame segmentation, it is not essential to segment all frames, but rather to selectively segment key frames. This novel method for selecting key segmentation frames can reduce the computational cost of semantic segmentation, shorten processing time, and enhance real-time performance. The specific selection method is as follows:

1. If the frame is the first frame or the first frame after relocation;
2. If the number of dynamic features tracked in the last frame falls below a certain threshold;
3. The last frame is taken as the key frame;
4. If the past five consecutive frames meet the criteria;



(a) (b)

Figure 2: Comparison graphs before and after removal of dynamic feature points: (a) Before dynamic feature removal. (b) After the dynamic feature is removed

When using the network to segment the image, certain conditions must be met. When motion points are detected in key segmentation frames,the image frames need to be evaluated. Firstly, after the image frame has been input, the first frame and the first frame obtained after relocation are segmented, which ensures that dynamic points are tracked and semantic images are generated. Secondly, if the count of dynamic features tracked in the previous frame falls below a specified threshold, the subsequent frame image undergoes segmentation. Thirdly, the last frame is taken as the key frame because it carries more semantic information, which enables the camera to better estimate its attitude. Finally, it is ensured that at least one frame in every five key frames undergoes semantic segmentation, achieving a fine balance between positioning accuracy and time cost. When one of the above mentioned principles is satisfied, the current frame is considered the key segmentation frame, and semantic segmentation is then performed in a separate thread.

3.4 Dynamic Feature Removal

For the removal of moving objects, a specific approach is necessary. Employing the Mask R-CNN network facilitates the elimination of dynamic objects. However, there remain instances of moving objects that escape detection through this method; these objects are not inherently dynamic but are in motion, such as a book that someone carries in their hand while moving. These objects tend to affect tracking and positioning. To remove these unknown non-prior dynamic objects, a new multi-view geometric constraint based on the polar line constraint is proposed in this study.

The objective is to integrate semantic data with the outcomes of multi-view geometric detection in order to distinguish between the semantic details of moving and stationary objects. Since human activities significantly disrupt robot positioning in reality, people are considered a suitable representative of dynamic objects. After obtaining the results of semantic segmentation, all ORB features are matched directly with the last frame if no people are detected. When a person is detected, the results of multi-view geometry detection are referenced to determine whether the person is moving. If the person is found to be stationary, the mentioned feature points can be utilized for estimating their pose. However, if they are moving, the ORB feature points should be removed to avoid any matching errors.

The geometric constraints with polar geometric properties can utilize to determine whether a feature is dynamic or static. In multi-view geometry, static features are supposed to satisfy the polar constraint, while dynamic features violate the standard polar constraint. As a common method, polar constraints are shown in Fig. 3.

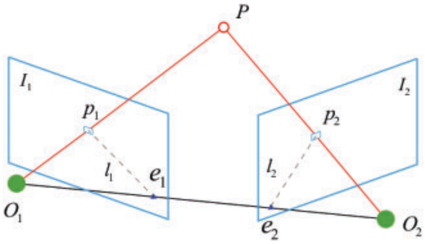


Figure 3: Antipolar geometry

Based on the pinhole camera model, the camera captures the same spatial point P in space from various angles. The optical centers of the camera are denoted by O_1, O_2. P_1, P_2 represent the corresponding feature points of point P in the previous and current frames, where p_1 and p_2 are their homogeneous coordinate form:

$$P_1 = [u_1, v_1, 1], P_2 = [u_2, v_2, 1] \tag{1}$$

where u and v are the values in the image frame. The epipolar line, referred to as L_2, can be determined using the subsequent equation:

$$L_2 = [X, Y, Z]^T = FP_1 = F[u_1, v_1, 1]^T \tag{2}$$

where X, Y, Z represent line vector, F is the foundation matrix. In the optimal scenario, the coordinates of the corresponding point pairs between the two images adhere to the following constraint:

$$P_2{}^T FP_1 = [u_2, v_2, 1]^T L_2 \tag{3}$$

Then the distance D from P_2 point to the pole L_2 is:

$$D = \frac{|P_2{}^T FP_1|}{\sqrt{||X||^2 + ||Y||^2}} \tag{4}$$

In general, the distance from D may be basically the same. However, the distance from D may vary significantly in dynamic scenarios. If the distance D exceeds a certain value, it does not conform to the polar coordinate constraint for which it is treated as a dynamic point. To set the threshold, dynamic objects were manually labeled for 30 images on four high-dynamic sequences, with accuracy and recall rate evaluated using different thresholds. By maximizing the expression $\varepsilon = 0.7 * precision + 0.3 * recall$, the reasonable threshold ε is determined as 0.40.

Firstly, feature points are identified from two consecutive frames, followed by matching the descriptors through calculation. After the matching points are obtained, the distance of each matching pair as previously defined is calculated. Dynamic points can be identified by comparing them with thresholds. If the threshold is exceeded, the point is classified as dynamic. The detection procedure is outlined in Algorithm 1.

Algorithm 1: Dynamic points detection algorithm

Input: Previous frame, F_1; Current frame, F_2; Previous frame's feature points, P_1; Current frame's feature points, P_2

Output: Dynamic feature point set S;

```
$P_2$ = CalcOpticalFlowPyrLK ($F_1, F_2, P_1$)
Remove outliers in $P_2$
FundmentalMatrix = FindFundamentalMat ($P_2$, $P_1$)
for each matched pairs $p_1, p_2$ in $P_1$, $P_2$ do
    $L_2 = FindEpipolarLine(p_1, F)$
    $D = CalcDistanceFromEpipolarLine(p_2, L_2)$
    $if\ D > \varepsilon\ then$
        Append $p_2\ to\ S$
    end if
end for
```

4 Experiment and Analysis

In this section, experiments were conducted using the commonly used TUM RGB-D dataset to evaluate the time efficiency of our approach. The performance of the enhanced algorithm was assessed using the publicly accessible TUM RGB-D dataset. To evaluate the algorithm's performance in dynamic environments, two categories of scenes were selected: low-dynamic (sitting) and high-dynamic (walking). These scenes incorporated various camera motion modes such as "half-sphere" (camera moving along a hemispherical path), "r-p-y" (camera rotating around the x-y-z axes), and "static" (camera remaining virtually fixed). Each scene was labeled using the format "dataset name algorithm type", where "w" denoted high-dynamic scenes and "s" denoted low-dynamic scenes. Our approach was compared with ORB-SLAM2 [5], DS-SLAM [21], DynaSLAM [22], and RDS-SLAM [26]. Since our approach is built upon ORB-SLAM2, its performance was evaluated against the original ORB-SLAM2 to demonstrate its improved efficiency. Additionally, our method was compared to DynaSLAM and DS-SLAM. Through these comparisons, the advantages of our approach are highlighted. To evaluate SLAM approach, the TUM automated evaluation tool was used.

The software used for this experiment is Ubuntu 18.04, while the hardware configuration consisted of an 8-core AMD i5-12490F CPU, a GeForce RTX 4060Ti graphics card, and 32 GB of RAM. The GPU was used exclusively for semantic.

4.1 Ablation Study

To demonstrate the effects of geometric methods and semantic segmentation on how the system performs, ablation experiments were conducted using the TUM dataset. The outcomes are shown in Table 1. To isolate the effects of the two modules, (S) represents the use of semantic segmentation alone for dynamic feature point detection and removal, and (G) represents the use of the geometric method alone. (S+G) indicates the combination of two modules. As shown in Table 1, ORB-SLAM2 has the highest absolute trajectory error and the lowest accuracy. Although the semantic segmentation approach can significantly improve accuracy in dynamic environments, it cannot solve some unknown dynamic interferences, preventing the accuracy from reaching an optimal level. While accuracy can also be improved by using only the geometric segmentation module, the effect is not as pronounced as when using the semantic segmentation module alone. This is because this method cannot quickly identify potential dynamic points. The combined system (S+G) has the minimal error and achieves the best results.

Table 1: ATE result from ablation experiments utilizing various methods with our system (unit: m)

Sequence	ORB-SLAM2	MG-SLAM(S)	MG-SLAM(G)	MG-SLAM(S+G)
Walking-half	0.2446	0.0190	0.0280	0.0162
Walking-rpy	0.1532	0.0445	0.1070	0.0341
Walking-static	0.0251	0.0082	0.0094	0.0072
Walking-xyz	0.2731	0.0174	0.0230	0.0135
Sisting-half	0.0609	0.0512	0.0609	0.0450

4.2 Comparative Experimental Data and Analysis with ORB-SLAM2

For the purpose of comparison, RMSE, MEAN, and Std of both ATE (Absolute Trajectory Error) and RPE (Relative Pose Error) were chosen as metrics to assess the precision of the SLAM system. According to the ATE results, there is a significant deviation between attitude estimation and ground truth value. RPE are used mainly to calculate the relative motion error between trajectories. As demonstrated by the data in Tables 2 and 3, MG-SLAM exhibits superior positioning accuracy compared to ORB-SLAM2 within dynamic settings. Across all dynamic sequences, MG-SLAM delivers reliable results, with trajectory errors consistently lower than those of ORB-SLAM2. This superiority is particularly evident in high-dynamic scenarios characterized by moving objects and noticeable motion in the imagery, where MG-SLAM exhibits enhanced performance. Specifically, MG-SLAM achieves over a 95% improvement in localization accuracy. In the walk-half sequences, this improvement is slightly more than 93%, and even in relatively static motion scenes, MG-SLAM still manages to demonstrate some degree of accuracy enhancement.

Fig. 4 illustrates the distribution of ATE for both ORB-SLAM2 and MG-SLAM. Specifically, Fig. 4a,b depicts the ATE for ORB-SLAM2 and MG-SLAM, respectively, in the walking-xyz sequence. A comparison reveals that MG-SLAM exhibits a lower absolute trajectory error compared to ORB-SLAM2, suggesting greater positioning accuracy.

Table 2: Results of ATE in TUM (unit: m)

Sequence	ORB-SLAM2/m			MG-SLAM/m			Improvement/%		
	RMSE	Mean	Std	RMSE	Mean	Std	RMSE	Mean	Std
Walking-half	0.2446	0.2192	0.1082	**0.0162**	0.118	0.0110	**93.37**	94.62	89.93
Walking-rpy	0.1532	0.1373	0.0679	**0.0341**	0.0178	0.0160	77.74	87.03	76.44
Walking-static	0.0251	0.0203	0.0148	**0.0079**	0.0073	0.0036	68.53	64.04	75.68
Walking-xyz	0.2731	0.0246	0.1188	**0.0135**	0.1080	0.0102	**95.05**	94.78	91.36
Sisting-half	0.0609	0.0519	0.0325	0.0450	0.0340	0.0132	26.11	34.48	59.38

Table 3: Results of RPE in TUM (unit: m)

Sequence	ORB-SLAM2/m			MG-SLAM/m			Improvement/%		
	RMSE	Mean	Std	RMSE	Mean	Std	RMSE	Mean	Std
Walking-half	0.0161	0.0118	0.0110	0.0115	0.0093	0.0072	28.57	21.18	34.54
Walking-rpy	0.0056	0.0045	0.0033	0.0020	0.0016	0.0012	64.29	64.44	63.64
Walking-static	0.0110	0.0108	0.0104	**0.0008**	0.0006	0.0005	92.73	94.40	94.19
Walking-xyz	0.0089	0.0078	0.0042	**0.0026**	0.0015	0.0018	70.78	80.76	57.14
Sisting-half	0.0139	0.0142	0.0087	0.0078	0.0065	0.0042	43.88	54.23	51.72

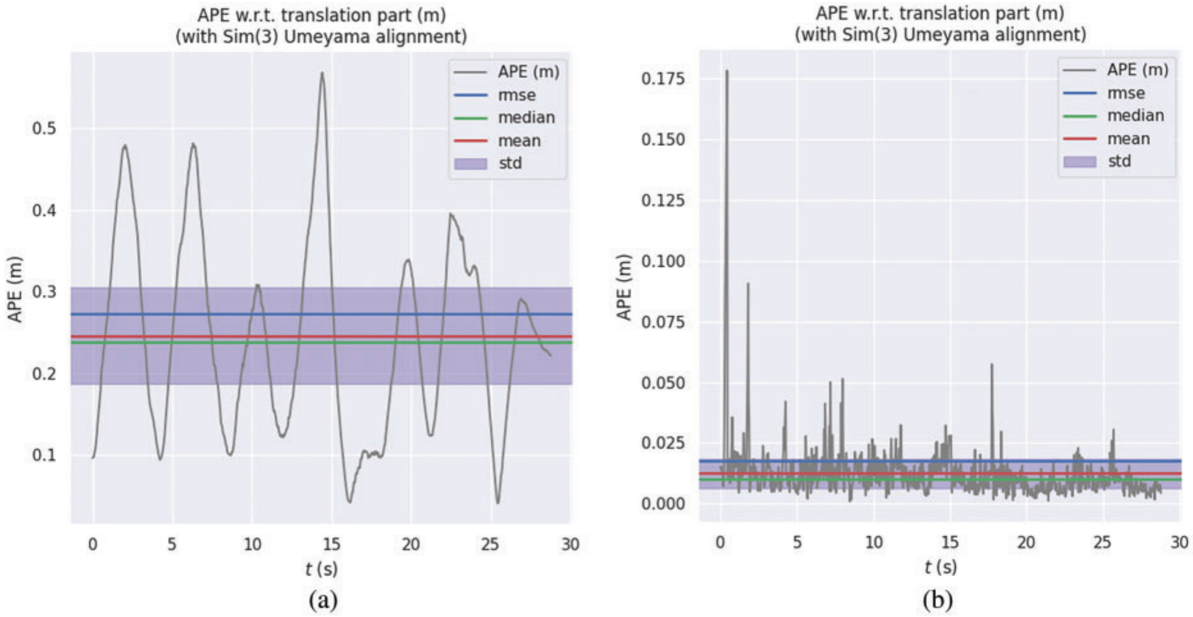
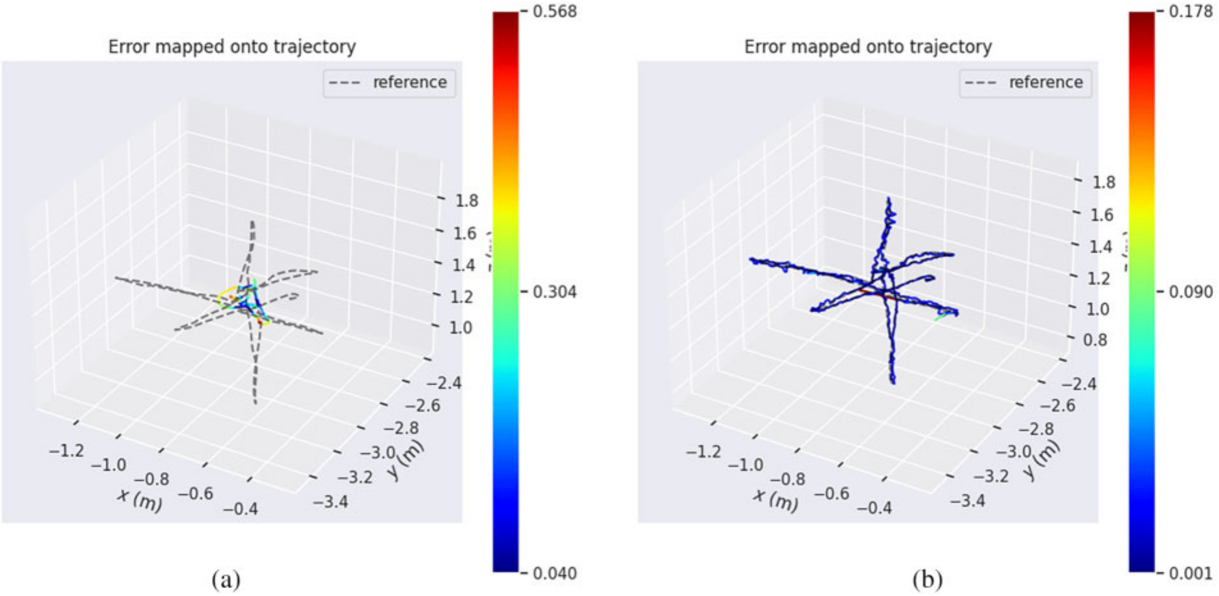


Figure 4: Walking-xyz ATE graphs: (a) ORB-SLAM2; (b) MG-SLAM

Fig. 5 illustrates the ATE of both ORB-SLAM2 and MG-SLAM in the 3D spatial space for the walking-xyz sequence. In the figures, the colored lines represent the system's trajectory errors mapped onto the true path. As shown in Fig. 5a, there is a notable difference between ORB-SLAM2 and the actual ground trajectory, which is mainly attributed to the number of dynamic objects within the camera's view. As the quantity of dynamic objects increases in this field, so too does the ATE error associated with ORB-SLAM2. Consequently, the divergence between trajectory mapping and ground truth values becomes increasingly pronounced due to tracking failures caused by dynamic objects. In contrast, MG-SLAM demonstrates superior performance, with its color curves closely aligning with real-world trajectories. Fig. 5a presents the ATE diagram for ORB-SLAM2, and Fig. 5b presents the ATE diagram for MG-SLAM.

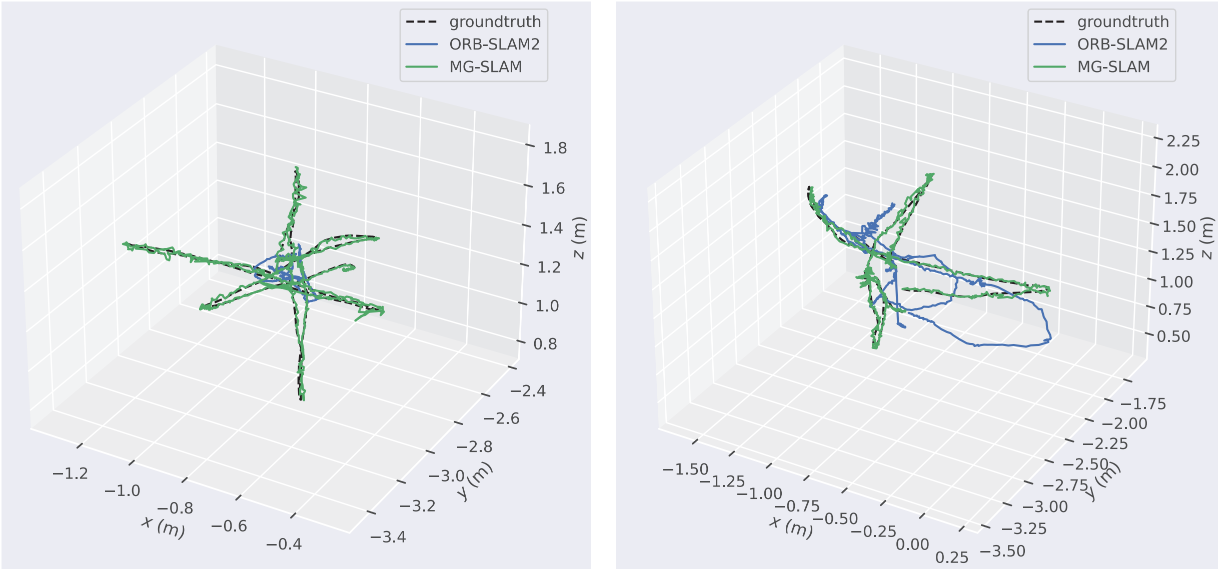


(a)

(b)

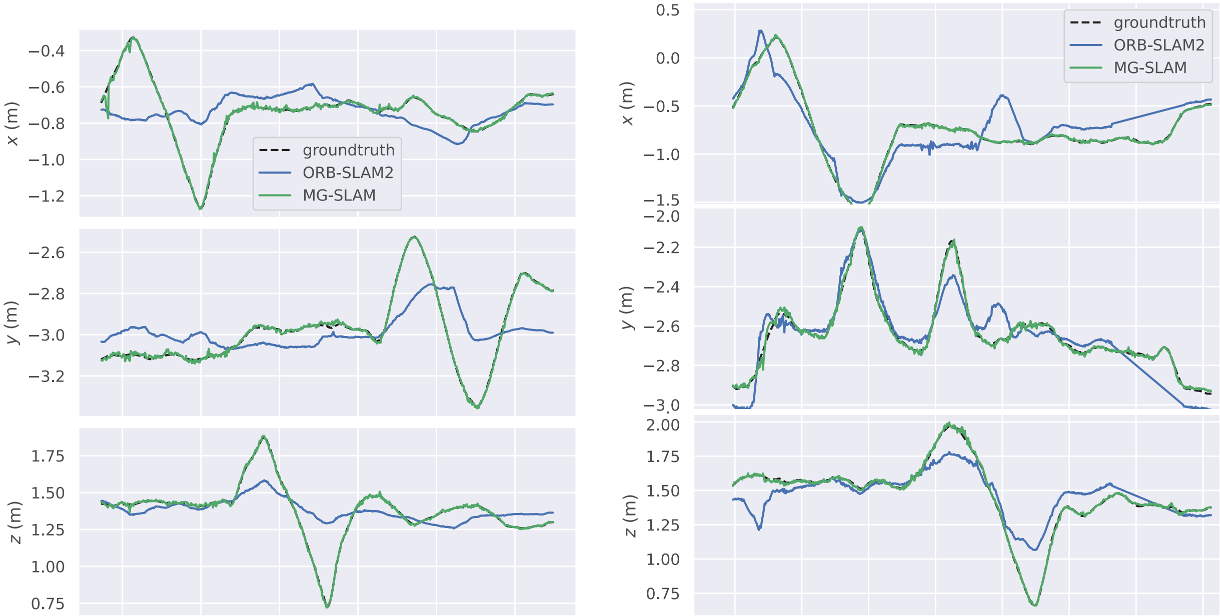
Figure 5: ATE graphs for the walking-xyz sequence in 3D space: (a) ORB-SLAM2; (b) MG-SLAM

To improve the clarity of the comparison, the actual trajectories of ORB-SLAM2 and MG-SLAM were juxtaposed against the computed trajectories. Fig. 6 displays the distinct 3D trajectories, and Fig. 7 provides a comparative overview of these trajectories in both the xyz and rpy coordinate systems. Figs. 6a and 7b specifically present the results from the walking-xyz sequence within the dataset, while Figs. 6b and 7b show results from the first half of the walking sequence. In these figures, the true ground trajectory is indicated by a gray dashed line, and the trajectories for ORB-SLAM2 and MG-SLAM are represented by blue and green lines, respectively. As motion intensity increases, the discrepancy between the trajectories of ORB-SLAM2 and MG-SLAM, when compared to the actual trajectory, is becoming increasingly noticeable. This is due to the movement of dynamic objects potentially causing feature point tracking to fail, thereby impacting the precision of pose estimation and increasing the ATE error values. Consequently, the mapped trajectories diverge further from the actual path on the ground. Notably, MG-SLAM's trajectory, represented by the green line, is significantly closer to the actual ground track than ORB-SLAM2's blue line, indicating superior accuracy. This suggests that MG-SLAM demonstrates superior performance compared to ORB-SLAM2 in pose estimation and localization precision, especially under high motion intensity.



(a) (b)

Figure 6: Graphs comparing distinct 3D trajectories: (a) walking-xyz (b) walking-half



(a) (b)

Figure 7: Comparative graphs displaying trajectories in the xyz directions: (a) walking-xyz (b) walking-half

4.3 Comparative Experimental with Other Dynamic SLAM

In order to further demonstrate the advantages of our algorithm, a comparative experiment was conducted with other SLAM algorithms commonly used in dynamic cases, including DS-SLAM

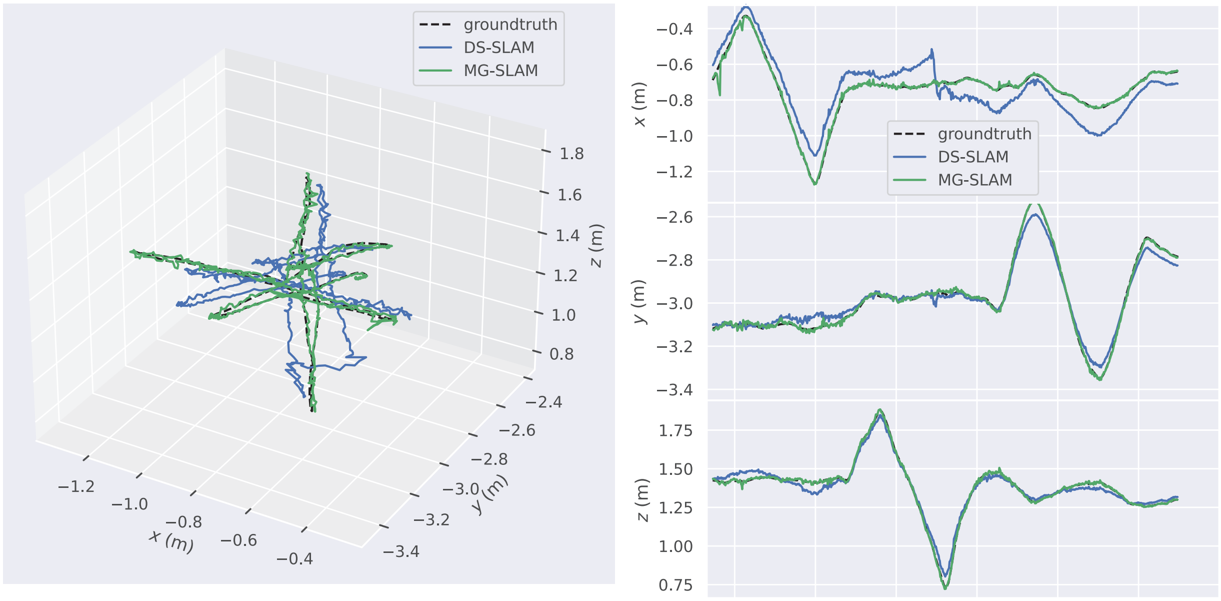
[21], DynaSLAM [22], RDS-SLAM [26] and Dynamic-VINS [35]. The ATE serves as a metric for evaluation, and the corresponding results are presented in Table 4. The approach presented in this paper demonstrates impressive performance on the TUM dataset. with the results in the table indicating that MG-SLAM outperforms current advanced dynamic SLAM. In highly dynamic sequences, MG-SLAM has lower ATE absolute values even when compared to DynaSLAM, which performs well in localization accuracy. Compared with the RDS-SLAM, MG-SLAM achieves higher localization accuracy and shows stronger performance. The ATE value is also lower than that of Dynamic-VINS. However, in low-dynamic sequences, the localization accuracy of MG-SLAM is lower than that of DS-SLAM. This could be attributed to the removal of an excessive number of feature points belonging to dynamic but stationary objects during the process of eliminating dynamic feature points. This, in turn, may affect subsequent tracking and result in decreased positioning accuracy.

Table 4: Results of ATE in TUM (unit: m)

Sequence	DS-SLAM/m		DynaSLAM/m		RDS-SLAM/m		Dynamic-VINS/m		MG-SLAM	
	RMSE	Std	RMSE	Std	RMSE	Std	RMSE	Std	RMSE	Std
Walking-half	0.0271	0.0113	0.0473	0.0096	0.0259	0.0341	0.0608	0.0423	**0.0162**	0.0110
Walking-rpy	0.4169	0.2564	0.2903	0.2277	0.1468	0.1051	0.0629	0.0516	**0.0341**	0.0160
Walking-static	0.0092	0.0046	0.0162	0.0036	0.0815	0.0224	0.0087	0.0076	**0.0079**	0.0036
Walking-xyz	0.0218	0.0127	0.0198	0.0077	0.0213	0.0127	0.0486	0.0324	**0.0135**	0.0102
Sisting-half	**0.0252**	0.0029	0.1140	0.1076	0.0473	0.0096	0.0562	0.0432	0.0405	0.0132

Fig. 8a shows the 3D spatial comparison of DS-SLAM and MG-SLAM with the real ground on the dataset walking-xyz. Fig. 8b shows a comparison plot in the xyz direction. The figure demonstrates that MG-SLAM's estimated motion path closely follows the actual ground trajectory, outperforming DS-SLAM in this regard. It indicates that MG-SLAM has higher positioning accuracy. Fig. 9a shows the distribution of ATE between DS-SLAM and MG-SLAM on the sitting-half sequence of the dataset, and Fig. 9b compares their traces in the xyz directions. The results in the figures show that the fluctuation amplitude of the green line trajectory is relatively small. This suggests that the DS-SLAM system achieves greater precision in locating within the sitting-half sequence.

In summary, MG-SLAM exhibits smaller absolute estimation errors and superior localization accuracy compared to DS-SLAM [21], DynaSLAM [22], RDS-SLAM [26] and Dyna-VINS [35] in sequences featuring larger dynamic target motion amplitudes. However, the effectiveness of DS-SLAM becomes more pronounced when fewer movements of the dynamic targets are detected. This is due to the fact that when MG-SLAM performs dynamic feature removal, it may incorrectly identify stationary but potentially movable object areas as dynamic objects, resulting in the removal of all their feature points. Consequently, only a limited number of feature points are available for subsequent tracking and positioning, which in turn reduces tracking and positioning accuracy.



(a) (b)

Figure 8: Walking-xyz shows a comparison diagram of three tracks: (a) in 3D space (b) in the xyz direction

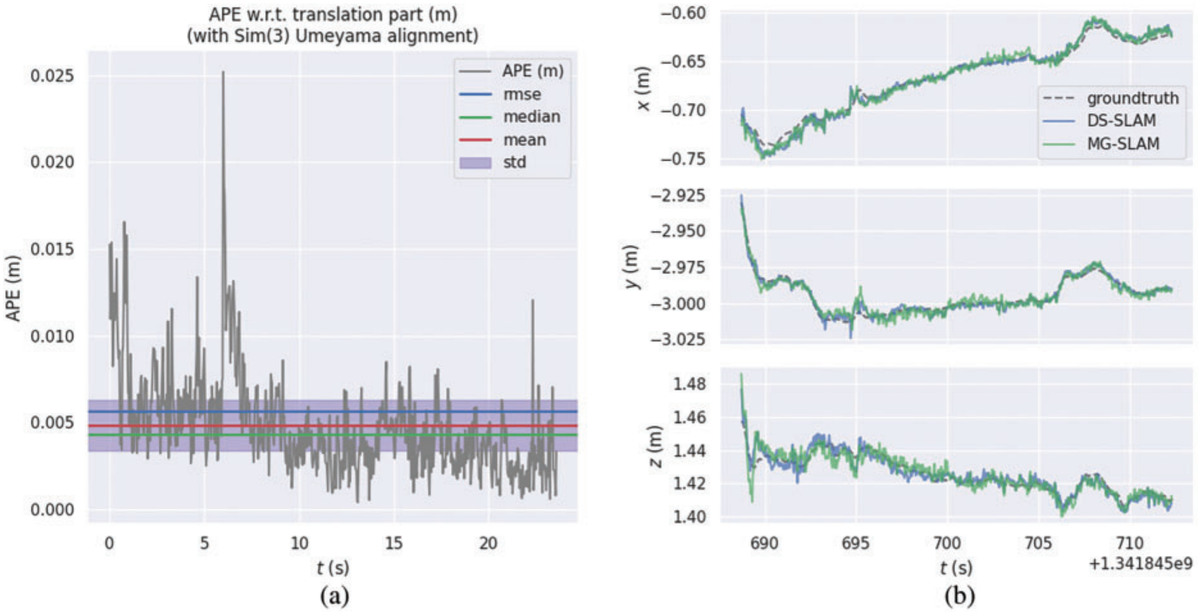


Figure 9: Experimental data graphs for the sitting-half sequence: (a) shows the distribution of ATE. (b) is a comparison diagram of the three trajectories on the *xyz* line

To enhance the system's resilience and demonstrate its performance in an outdoor environment, the system was validated on the KITTI dataset. The KITTI dataset enjoys widespread use and international recognition, serving as a pivotal benchmark for the autonomous driving industry. It contains realistic images of many outdoor environments. This real-world scenario data provides rich challenges for algorithms, helping to develop more robust and accurate algorithms. In this paper, Sequences 00, 01, 03, 07 were selected, MG-SLAM and ORB-SLAM2 were compared and analyzed, and the RMSE of ATE and RPE was selected as the measurement index. As can be seen from Table 5, MG-SLAM has lower trajectory error than ORB-SLAM2, especially in Sequence 03. Fig. 10 presents a comparative analysis of the ATE distribution between MG-SLAM and ORB-SLAM2 in that sequence. MG-SLAM demonstrates better positioning accuracy than ORB-SLAM2, with the absolute estimated error (ATE) decreasing from 0.3193 to 0.2405 m. These results show that MG-SLAM has good performance in handling outdoor environments.

Table 5: Results of ATE in KITTI (unit: m)

Sequence	00		01		03		07	
	APE (m)	RPE (m)	APE (m)	RPE (m)	APE (m)	RPE (m)	APE (m)	RPE (m)
ORB-SLAM2	0.9280	0.0275	5.7857	0.0538	0.3193	0.0215	0.5171	0.0183
MG-SLAM	0.8935	0.0235	5.1658	0.0479	**0.2405**	0.0171	**0.4253**	0.0149

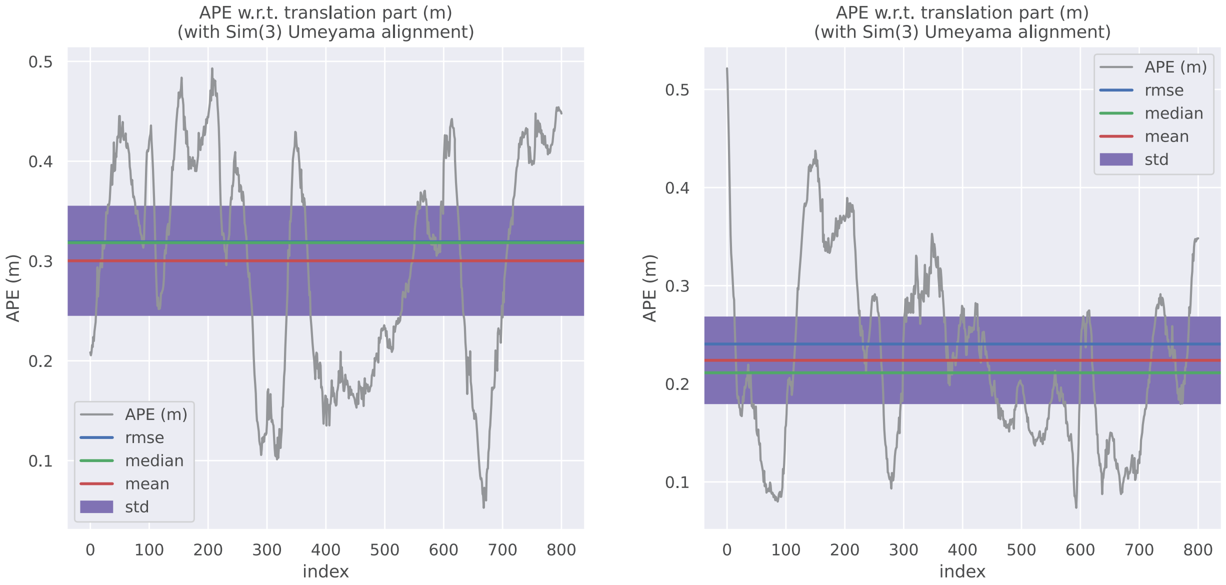


Figure 10: KITTI-03 ATE graphs: (a) ATE of ORB-SLAM2; (b) ATE of MG-SLAM

4.4 Real-Time Comparison

For SLAM, real-time performance is another crucial metric. When systems often sacrifice real-time efficiency in pursuit of high accuracy, they may not be suitable for practical applications. The time cost per frame was compared in various running environments. Also, the real-time properties of DS-SLAM [21], DynaSLAM [22], RDS-SLAM [26], ORB-SLAM2 and MG-SLAM were verified. The experimental results are displayed in Table 6.

According to Table 6, MG-SLAM takes less time than other dynamic SLAM algorithms and has better real-time performance, and GPU-based MG-SLAM greatly improves the performance of dynamic object segmentation and detection. Therefore, despite the addition of the dynamic target segmentation detection process, which took more time than the ORB-SLAM2, the performance of MG-SLAM was still superior.

Table 6: Real-time performance comparison of MG-SLAM and other dynamic scene SLAM algorithms

Sequence	DS-SLAM	DynaSLAM	ORB-SLAM2	RDS-SLAM	MG-SLAM
Walking-half	0.2530	0.4274	0.0143	0.0752	0.0349
Walking-rpy	0.2528	0.4131	0.0137	0.0785	0.0340
Walking-static	0.2637	0.4189	0.0162	0.0774	0.0354
Walking-xyz	0.2530	0.4274	0.0143	0.0750	0.0349

5 Conclusions

In this paper, a novel SLAM (Simultaneous Localization and Mapping) algorithm is proposed to address the challenges in attitude estimation for dynamic targets. This algorithm employs an optimized deep learning approach to segment and detect the targets within dynamic scenes, which ensures real-time and high-precision performance. It is built upon the ORB-SLAM2 [5] algorithm and incorporates the semantic segmentation algorithm and a multi-view geometry algorithm. To mitigate the impact of dynamic objects, modules for dynamic object segmentation and dynamic feature point removal are introduced. These modules effectively filter out dynamic elements from the scene, such as moving pedestrians. To assess and benchmark the performance of our system, an extensive series of experiments has been conducted. We tested our system on both the indoor TUM dataset and the outdoor KITTI dataset, comparing it against the established baseline ORB-SLAM2 method and other dynamic SLAM techniques, including DS-SLAM [21], DynaSLAM [22], RDS-SLAM [26], and Dynamic-VINS [35]. The experimental outcomes demonstrate that MG-SLAM system significantly reduces the positioning error in highly dynamic indoor settings, with a positioning accuracy that surpasses the original ORB-SLAM2 baseline by up to 95%. Furthermore, MG-SLAM exhibits commendable localization capabilities in outdoor scenarios as well. Leveraging GPU acceleration for the Mask R-CNN segmentation network, MG-SLAM performs real-time segmentation only on key frames. This strategic approach substantially enhances the efficiency of semantic segmentation, minimizes computational costs, and boosts the system's time efficiency. In comparison to other methods, MG-SLAM processes data at a faster rate, even outperforming RDS-SLAM, which underscores its exceptional real-time performance. Despite these advancements, there is scope for future optimization. This paper utilizes a semantic segmentation network that incorporates semantic information, which can be harnessed to construct a three-dimensional semantic map. This would enable path planning in dynamic environments. Additionally, integrating additional sensors, such as Inertial Measurement Units (IMU), could improve the system's resilience in a variety of challenging environments.

Acknowledgement: I am deeply grateful to Dr. Kai Zhu for his invaluable advice and mentorship throughout the research journey. My heartfelt thanks also go out to all my colleagues in the lab, whose professional insights and suggestions have significantly influenced our research endeavors.

Funding Statement: This research was funded by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (grant number 22KJD440001) and Changzhou Science & Technology Program (grant number CJ20220232).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Fengju Zhang; analysis and interpretation of results: Fengju Zhang; draft manuscript preparation: Fengju Zhang, Kai Zhu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data supporting the findings of this study are openly available in a public repository at https://cvg.cit.tum.de/data/datasets/rgbd-dataset/download, accessed on 14 November 2024 and https://www.cvlibs.net/datasets/kitti, accessed on 14 November 2024.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] K. Qi, Q. Wu, N. Cheng, J. Wang, and K. B. Letaief, “Deep-reinforcement-learning-based AoI-aware resource allocation for RIS-aided IoV networks,” *IEEE Trans. Veh. Technol.*, vol. 73, no. 8, pp. 1–14, 2024. doi: 10.1109/TVT.2024.3452790.

[2] Q. Wu, W. Wang, P. Fan, Q. Fan, J. Wang and K. B. Letaief, “URLLC-awared resource allocation for heterogeneous vehicular edge computing,” *IEEE Trans. Veh. Technol.*, vol. 73, no. 8, pp. 11789–11805, Aug. 2024. doi: 10.1109/TVT.2024.3370196.

[3] Q. Wu, W. Wang, P. Fan, Q. Fan, H. Zhu and K. B. Letaief, “Cooperative edge caching based on elastic federated and multi-agent deep reinforcement learning in next-generation networks,” *IEEE Trans. Netw. Serv. Manag., Aug*, vol. 21, no. 4, pp. 4179–4196, 2024. doi: 10.1109/TNSM.2024.3403842.

[4] E. S. Ali *et al.*, “Machine learning technologies for secure vehicular communication in internet of vehicles: Recent advances and applications,” *Secur. Commun. Netw.*, vol. 2021, no. 3, pp. 1–23, Mar. 2021. doi: 10.1155/2021/8868355.

[5] R. Mur-Artal and J. D. Tards, “ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras,” *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, 2017. doi: 10.1109/TRO.2017.2705103.

[6] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” in *Eur. Conf. Comput. Vis.*, Springer, 2014, pp. 834–849.

[7] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, “Semanticfusion: Dense 3D semantic mapping with convolutional neural networks,” in *2017 IEEE Int. Conf. Robot. Autom. (ICRA)*, IEEE, 2017, pp. 4628–4635.

[8] A. Hermans, G. Floros, and B. Leibe, “Dense 3D semantic mapping of indoor scenes from RGB-D images,” in *2014 IEEE Int. Conf. Robot. Autom. (ICRA)*, IEEE, 2014, pp. 2631–2638.

[9] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, “Fusion++: Volumetric object-level slam,” in *2018 Int. Conf. 3D Vis. (3DV)*, IEEE, 2018, pp. 32–41.

[10] S. Yang and S. Scherer, “CubeSLAM: Monocular 3D object slam,” *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, 2019. doi: 10.1109/TRO.2019.2909168.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[12] G. Chaple and R. D. Daruwala, “Design of Sobel operator based image edge detection algorithm on FPGA,” in *2014 Int. Conf. Commun. Signal Process.*, IEEE, 2014, pp. 788–792.

[13] Y. Li, Z. Liu, K. Xu, H. Yu, and F. Ren, “A GPU-outperforming FPGA accelerator architecture for binary convolutional neural networks,” *ACM J. Emerg. Technol. Comput. Syst. (JETC)*, vol. 14, no. 2, pp. 1–16, 2018. doi: 10.1145/3154839.

[14] Y. Sun, M. Liu, and M. Q. -H. Meng, “Improving RGB-D SLAM in dynamic environments: A motion removal approach,” *Robot. Auton. Syst.*, vol. 89, no. 5, pp. 110–122, 2017. doi: 10.1016/j.robot.2016.11.012.

[15] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, “Robust monocular SLAM in dynamic environments,” in *2013 IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, IEEE, 2013, pp. 209–218.

[16] D. -H. Kim and J. -H. Kim, “Effective background model-based RGB-D dense visual odometry in a dynamic environment,” *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1565–1573, 2016. doi: 10.1109/TRO.2016.2609395.

[17] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, “Flowfusion: Dynamic dense RGB-D SLAM based on optical flow,” in *2020 IEEE Int. Conf. Robot. Autom. (ICRA)*, IEEE, 2020, pp. 7322–7328.

[18] W. Dai, Y. Zhang, P. Li, Z. Fang, and S. Scherer, “RGB-D SLAM in dynamic environments using point correlations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 373–389, 2020. doi: 10.1109/TPAMI.2020.3010942.

[19] Y. Qin, T. Mei, Z. Gao, Z. Lin, W. Song and X. Zhao, “RGB-D SLAM in dynamic environments with multilevel semantic mapping,” *J. Intell. Robot Syst.*, vol. 105, no. 4, Aug. 2022, Art. no. 90. doi: 10.1007/s10846-022-01697-y.

[20] C. Zhang, R. Zhang, S. Jin, and X. Yi, “PFD-SLAM: A new RGB-D SLAM for dynamic indoor environments based on non-prior semantic segmentation,” *Remote Sensing*, vol. 14, no. 10, 2022, Art. no. 2445. doi: 10.3390/rs14102445.

[21] R. Long, C. Rauch, T. Zhang, V. Ivan, T. L. Lam and S. Vijayakumar, “RGB-D SLAM in indoor planar environments with multiple large dynamic objects,” *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8209–8216, Jul. 2022. doi: 10.1109/LRA.2022.3186091.

[22] K. Tateno, F. Tombari, I. Laina, and N. Navab, “CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6243–6252.

[23] M. Kaneko, K. Iwami, T. Ogawa, T. Yamasaki, and K. Aizawa, “Mask-SLAM: Robust feature-based monocular SLAM by masking using semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 258–266.

[24] C. Yu *et al.*, “DS-SLAM: A semantic visual SLAM towards dynamic environments,” in *2018 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, IEEE, 2018, pp. 1168–1174.

[25] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, “DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4076–4083, 2018. doi: 10.1109/LRA.2018.2860039.

[26] H. Xu, C. Yang, and Z. Li, “OD-SLAM: Real-time localization and mapping in dynamic environment through multi-sensor fusion,” in *2020 5th Int. Conf. Adv. Robot. Mechatron. (ICARM)*, IEEE, 2020, pp. 172–177.

[27] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017. doi: 10.1109/TPAMI.2016.2644615.

[28] F. Zhong, S. Wang, Z. Zhang, and Y. Wang, “Detect-SLAM: Making object detection and SLAM mutually beneficial,” in *2018 IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, IEEE, 2018, pp. 1001–1010.

[29] Y. Liu and J. Miura, “RDS-SLAM: Real-time dynamic SLAM using semantic segmentation methods,” *IEEE Access*, vol. 9, pp. 23772–23785, 2021. doi: 10.1109/ACCESS.2021.3050617.

[30] J. Peng, “RO-SLAM: A robust SLAM for unmanned aerial vehicles in a dynamic environment,” *Comput. Syst. Sci. Eng.*, vol. 47, no. 2, pp. 2275–2291, 2023. doi: 10.32604/csse.2023.039272.

[31] W. Wu, L. Guo, H. Gao, Z. You, Y. Liu and Z. Chen, “YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint,” *Neural Comput. Appl.*, vol. 34, no. 8, pp. 6011–6026, Apr. 2022. doi: 10.1007/s00521-021-06764-3.

[32] S. Cheng, C. Sun, S. Zhang, and D. Zhang, "SG-SLAM: A real-time RGB-D visual SLAM toward dynamic scenes with semantic and geometric information," *IEEE Trans. Instrum. Meas.*, vol. 72, no. 3, pp. 1–12, 2023. doi: 10.1109/TIM.2022.3228006.

[33] J. Jin, X. Jiang, C. Yu, L. Zhao, and Z. Tang, "Dynamic visual simultaneous localization and mapping based on semantic segmentation module," *Appl. Intell.*, vol. 53, no. 16, pp. 19418–19432, Aug. 2023. doi: 10.1007/s10489-023-04531-6.

[34] L. Chen, Z. Ling, Y. Gao, R. Sun, and S. Jin, "A real-time semantic visual SLAM for dynamic environment based on deep learning and dynamic probabilistic propagation," *Complex Intell. Syst.*, vol. 9, no. 5, pp. 5653–5677, Oct. 2023. doi: 10.1007/s40747-023-01031-5.

[35] J. Liu, X. Li, Y. Liu, and H. Chen, "RGB-D inertial odometry for a resource-restricted robot in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9573–9580, Oct. 2022. doi: 10.1109/LRA.2022.3191193.