A Semantic SLAM Integrated with Enhanced YOLOv7 Target Detection Algorithm

ZhangFang Hu, FangYu Li, JiXiang Shen

Abstract—**This paper proposes a semantic SLAM integrated with an enhanced YOLOv7 target detection algorithm. To address the issue of image blurring caused by robot movement and camera shake, we have incorporated an image enhancement module before the tracking thread. Consequently, the resulting images are more clearer. In the feature extraction stage, we introduce adaptive thresholds to improve the system's capability in feature point extraction. To minimize the influence of dynamic objects on this system, we employ an enhanced YOLOv7 algorithm to detect dynamic targets. Then, we integrate it with epipolar constraint to eliminate dynamic feature points. Finally, We evaluated our system with five sequences taken from the TUM dataset, and compared with ORB-SLAM3, our system improves more than 91% in accuracy, up to 98%. Moreover, compared to similar semantic SLAM systems, our system offers improved accuracy as well as enhanced real-time performance.**

Index Terms—**semantic SLAM, image enhancement, adaptive thresholds, YOLOv7, epipolar constraint**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM) refers to a process where a robot, without any prior information, simultaneously localizes itself and constructs a map of its surrounding area[1]. Based on the types of sensors used, researchers divide SLAM into Laser SLAM (LS) and Visual SLAM (VS). Among them, VS has the benefits of reduced expenses and access to obtain more data from surroundings, which can give mobile robots stronger environmental awareness [2]. So, vision-based simultaneous localization and map building techniques have been widely studied and applied to robot navigation [3], unmanned driving [4] and virtual reality [5].

In practical applications, visual SLAM needs to be real-time and robust. Traditional visual SLAM primarily relies on understanding the environment through geometric features of images [6], which has high real-time performance because it only focuses on geometric features in the environment. Traditional VS relies on the assumption of a stationary surroundings. But it doesn’t hold true in real-world scenarios where moving entities, such as walkers and automobiles, are unavoidably present. Dynamic environments generate numerous incorrect data associations [7], leading to a reduction in the accuracy of VS systems. Consequently, traditional VS systems exhibit lower robustness. Furthermore, traditional visual SLAM lacks the capability to comprehend the environment at a high level and fails to meet the demands for human-computer interaction in the current era of intelligent technology. To address the limitations of traditional visual SLAM, visual SLAM designed for dynamic environments has emerged.

In dynamic environments, the primary objective of VS is to minimize the influence of moving entities on the SLAM system by excluding feature points associated with these objects. This challenge is addressed through two distinct approaches: geometric-based dynamic visual SLAM and deep learning-based dynamic visual SLAM, each utilizing specific techniques designed for this purpose. Geometry-based dynamic visual SLAM employs geometric information of the environment to eliminate dynamic features, and a prevalent approach is the maximum consistency scheme, such as the Random Sample Consensus Algorithm (RANSAC [8]). In addition, many visual SALM systems use multi-sensor fusion to detect dynamic targets in the environment, such as ORB-SLAM3 [9]. However, these methods are effective only when dynamic objects are few, and they fail to capture high-level information about the environment, resulting in an insufficient understanding of the surroundings. Dynamic visual SLAM based on deep learning, also known as semantic SLAM, is capable of acquiring both geometric information about unfamiliar environments and the motion states of robots. Moreover, it can detect and recognize targets in the surroundings, allowing for the filtering out of dynamic feature points (dfp). This capability enables the robot to enhance its comprehension of its surroundings.Moreover it also allows robots to function effectively in more complex environments.

Although many semantic SLAM methods perform well in dynamic environments, they still suffer from certain issues. Firstly, many semantic SLAM methods overlook the problem of image blurring caused by camera shake during robot movement, resulting in insufficient extraction of both geometric and semantic information by the system. Secondly, approaches utilizing deep learning methods, such as semantic segmentation and target detection exhibit limitations. Some

Manuscript received on April 4, 2024; revised on August 27, 2024.

This work was supported in part by the Youth Fund Program of the National Natural Science Foundation of China (Grant No. 61703067), the Chongqing Basic Science and Frontier Technology Research Program (Grant No. Cstc2017jcyjAX0212), and the Science and Technology Research Program of Chongqing Municipal Education Commission (KJ1704072).

Zhangfang Hu is a Professor at the Key Laboratory of Optical Information Sensing and Technology, School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China (e-mail: 3565207151@qq.com)

Fangyu Li is a graduate student of the School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China (corresponding author phone: 183-8094-2351; e-mail: s220431046@stu.cqupt.edu.cn)

Jixiang Shen is a graduate student of the School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China (e-mail: s220432006@stu.cqupt.edu.cn)

approaches only remove feature points by relying on the results of semantic segmentation or object detection, as demonstrated by methods such as DS-SLAM [10], DynaSLAM [11], and SaD-SLAM [12]. This approach can lead to two main issues: misidentification of stationary feature points as moving, which reduces the number of useful features and impairs position estimation capability; and incomplete removal of some dynamic objects, which compromises the precision and robustness of this system. In addition, these methods use segmentation models such as SegNet [13], Mask R-CNN [14], etc. Although these models have high accuracy, they are more complex in structure and take longer to process the data, which fails to meet the criteria for real-time performance. Lastly, the conventional ORB (Oriented FAST and Rotated BRIEF) feature extraction method [15] employs a fixed threshold that is sensitive to changes in environmental lighting conditions. This dependency can lead to challenges such as failures in feature point extraction and redundancy in local feature points.

To address the aforementioned challenges in SLAM systems, we have developed a semantic SLAM system utilizing the ORB-SLAM3 framework, which ensures both efficiency and reliability in complex dynamic environments. Firstly, to tackle the problem of image blurriness resulting from camera shake and rapid motion of dynamic objects, we propose the implementation of an image enhancement module. This module utilizes the DeblurGANv2 network [16] to process blurry images, thereby improving image quality and facilitating the subsequent modules' operation. Furthermore, to alleviate the effects of dynamic entities on the SLAM system, we incorporate a parallel object detection thread within the ORB-SLAM3 framework, utilizing epipolar constraints and enhanced YOLOv7 to eliminate the dfp. In the object detection threads, we employ the lightweight YOLOv7 object detection network to derive semantic details from images. Simultaneously, we utilize this thread to identify the location of the target within the image. Additionally, to address the low precision issue of the YOLOv7 network, we integrate the SimAM attention mechanism into its feature extraction process. Lastly, we enhance the traditional ORB feature extraction method by adaptively adjusting the detection threshold of FAST corners based on the grayscale values of different regions in the image. A comparison with conventional ORB feature extraction, which uses a fixed threshold, demonstrates that our approach produces a greater number of useful feature points with a more uniform distribution. This enhancement ultimately increases the accuracy of subsequent pose estimation tasks.

The following parts of this article are structured in the manner outlined below. In Section II, we present some semantic SLAMs for dynamic environments, summarizing their results and shortcomings. In Section III, we elaborate on the system architecture. In Section IV, we perform experiments utilizing the TUM dataset and analyze the experimental results. And in Section V, we summarize the work in this paper.

II. Related Work

Traditional VS systems are predicated on a quiescent hypothesis, leading to poor localization and mapping performance in complex moving surroundings. In order to enhance the robustness of visual SLAM systems and enable better human-machine interaction, researchers have proposed Semantic SLAM. This approach utilizes deep learning-based object detection and semantic segmentation algorithms to remove dynamic regions within images. Subsequently, pose estimation and the development of maps that incorporate semantic information rely solely on sfp.

Object detection plays a crucial role in the domain of computer vision [17], aimed at localizing and classifying objects. It involves locating desired objects within given images or videos, marking their positions with bounding boxes, and determining their categories. With the progression of deep learning methodologies, numerous deep learning-based object detection algorithms have emerged, achieving impressive results in this field. Examples include Fast R-CNN [18] and the YOLO [19] series of algorithms.

Researchers have incorporated object detection algorithms into SLAM systems to mitigate the impact of dynamic entities on the performance of these systems. In 2018, Zhong F et al. proposed Detect-SLAM [20], that integrates SLAM with deep neural network (DNN)-based object detectors. This integration enhances the ability of robots to perform tasks effectively and reliably in unfamiliar and dynamic environments. Detect-SLAM combines SLAM with DNN-based detectors, simultaneously accomplishing three tasks: enhancing SLAM robustness in dynamic environments, improving object detection performance, and constructing semantic maps. In highly dynamic scenes, the trajectory estimated by the Detect-SLAM system closely approximates the true trajectory. However, compared to ORB-SLAM2, Detect-SLAM fails to yield desirable results in static scenes. This is because in static scenes, Detect-SLAM filters out some static information that is beneficial for camera pose estimation and subsequent mapping, thereby affecting the overall system's localization accuracy.

Semantic segmentation is a method of image segmentation utilized within the domain of computer vision. It focuses on the classification of each pixel in an image into predefined categories. Different from traditional image segmentation techniques, semantic segmentation not only divides an image into several regions but also classifies each pixel, thereby obtaining more precise image segmentation results. It is utilized in multiple domains, including autonomous driving, medical image analysis, and robotic vision.

In 2018, Bescos et al. introduced the DynaSLAM system. It is founded on ORB-SLAM2 framework. This system encompasses interfaces designed for monocular, stereo, and RGB-D camera configurations. When we utilize monocular and stereo cameras, Mask R-CNN is employed to perform segmentation of dynamic entities in every frame acquired by the camera. Thereby, this approach circumvents the necessity for feature extraction of dynamic entities within the SLAM framework. When utilizing an RGB-D camera as a sensor, the system integrates multi-view geometry methods to achieve more precise segmentation of dynamic objects, thereby enhancing system performance in dynamic environments. However, this approach involves significant computational overhead and does not meet real-time requirements.

The same year, C. Yu et al. introduced DS-SLAM, a method derived from ORB-SLAM2. Its main innovation lies in the addition of a independent real-time semantic segmentation module within the framework of ORB-SLAM2. This thread is capable of removing dynamic objects from the environment and creating a dense semantic octree map containing environmental semantic information, enabling the robot to perform higher-level tasks.

The above-mentioned methods have improved the property of SLAM systems to some extent, but mechanically removing dynamic objects can result in the loss of many usable feature points in the system. To tackle this problem, Cabon, Y et al. proposed the SLAMANTIC system [21], which does not require motion detection. Instead, it introduces confidence by assigning different probabilities of motion to each object to ascertain whether the object is in a state of motion. As a result, this methodology possesses the ability to distinguish between objects that might be regarded as dynamic, while they are, in reality, stationary. In addition, This system integrates semantic label distribution with the consistency of map point observations to evaluate the reliability of each 3D measurement point. Subsequently, this information is utilized for pose estimation and subsequent map optimization steps.

In summary, semantic SLAM outperforms traditional visual SLAM in overall performance. However, some methods focus solely on improving accuracy while overlooking real-time capabilities, while others exhibit good real-time performance but lower accuracy and robustness. Therefore, enhancing the accuracy of SLAM systems while maintaining a certain level of real-time property constitutes a significant area of research.

III. System Introduction

This section offers a comprehensive analysis of the system put forward in the present study. First, we present the improvements of the system proposed within this paper on the ORB-SLAM3 framework, including the incorporation of an image enhancement module, the introduction of adaptive thresholding, and the enhanced YOLOv7 network. Then, the methods of implementing the image enhancement module and using adaptive thresholding to improve the traditional ORB feature extraction are explained in detail. Lastly, the paper elaborates on the enhancement of the YOLOv7 network and its integration with polar constraints to propose a method for filtering dynamic feature points.

A. *Framework of our system*

As illustrated in Fig. 1, ORB-SLAM3 is an open-source VS system characterized by a tracking thread, a local mapping thread, and a loop closing thread responsible for maps fusion. As demonstrated in Fig. 2, our study is an improvement upon ORB-SLAM3. Initially, in the tracking threading, following the acquisition of image frames from the camera, we introduce an image enhancement module to improve the quality of processed images, ensuring enhanced feature point extraction and minimizing the impact of blurred images on the target detection thread. In the process of object detection, we utilize the improved YOLOv7 to extract semantic information pertaining to entities in images, including labels and positions. Improvements have been made to the traditional ORB feature point extraction method by employing adaptive thresholding, enhancing the robustness of feature point extraction. Finally, the semantic information obtained from the object detection process is combined with epipolar constraint to exclude feature points that are linked to dynamic entities. In the subsequent tasks, we will exclusively employ static feature points (sfp).

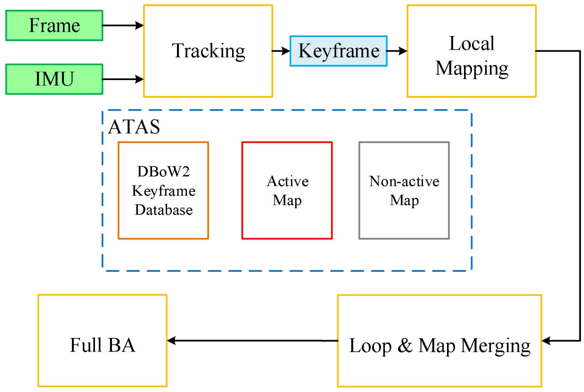


Fig.1. Framework of ORB-SLAM3

B. *Image Enhancement Module*

In the course of a mobile robot's movement, camera shake inevitably occurs, leading to blurry images. Additionally, fast-moving objects in the scene can also cause blurriness. Many previous SLAM methods have overlooked these issues, resulting in poor robustness and accuracy of SLAM systems in environments characterized by high dynamics. This paper introduces an image enhancement module into the framework of ORB-SLAM3. This module preprocesses images obtained from the camera using the DeblurGANv2 network.

The DeblurGANv2 network is an improvement over DeblurGAN [22], achieving better results. Furthermore, DeblurGANv2 utilizes the lightweight MobileNet [23] as its backbone network, resulting in a 20x speed improvement compared to DeblurGAN. It exhibits good real-time performance, meeting the real-time requirements of SLAM systems. The network architecture is illustrated in Fig. 3. The DeblurGANv2 network consists primarily of a generator and a discriminator. The generator employs the Feature Pyramid Network (FPN) structure, which gathers feature outputs from five branches and fuses them through upsampling to improve the quality of produced images. In the discriminator part, a relativistic discriminator utilizing a least-squares loss function is implemented. Additionally, it integrates global and local scale discriminator losses, ensuring a stable and efficient training process.

Before integrating the DeblurGANv2 network into our system, we enhanced its efficacy by curating a specialized dataset encompassing a diverse array of objects such as individuals, vehicles, books, and furniture. In our approach detailed in this paper, frames captured by an RGB-D camera are fed into an image enhancement module, leveraging the trained DeblurGANv2 network for deblurring. This procedure results in sharper frames, effectively mitigating blurring induced by camera shake and rapid object movements. Subsequently, the deblurred frames are fed into feature extraction and object detection modules to extract keypoints and detect objects within the frames. To assess the performance of DeblurGANv2, we conducted tests using the TUM dataset. The outcomes, illustrated in Fig.4, substantiate that the image enhancement module utilized in this study significantly ameliorates issues of image blurring attributed to camera shake and object motion.

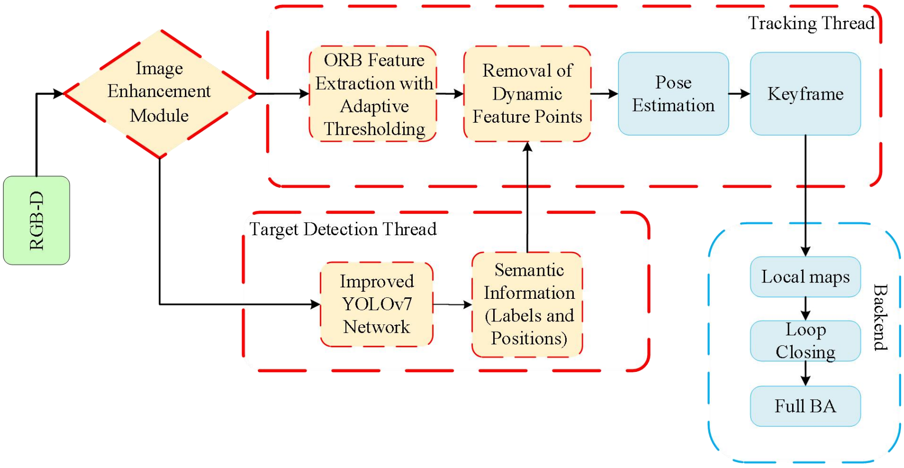


Fig.2. System Overview

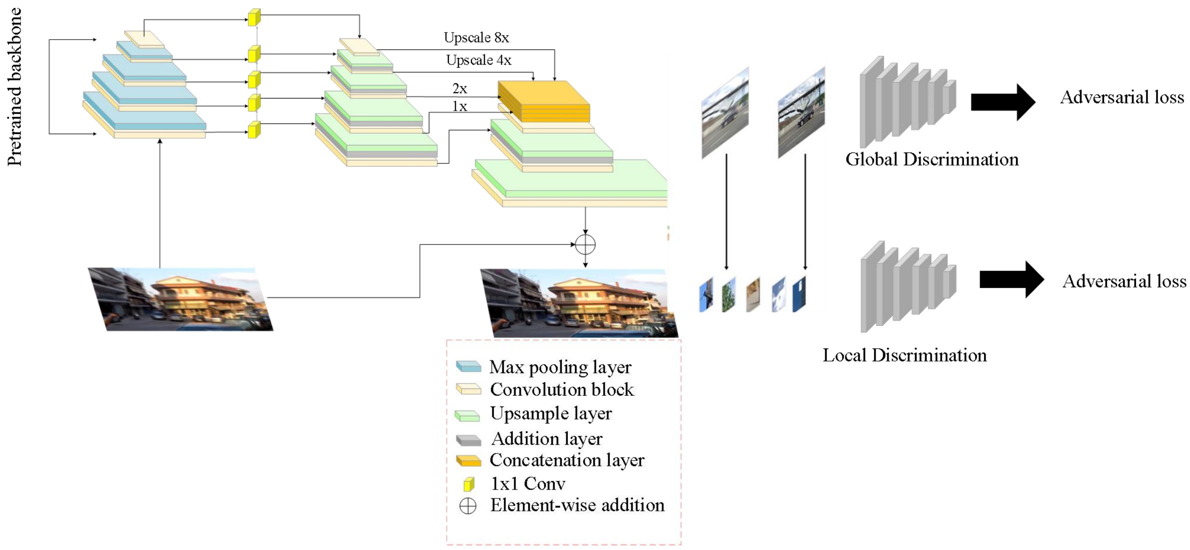
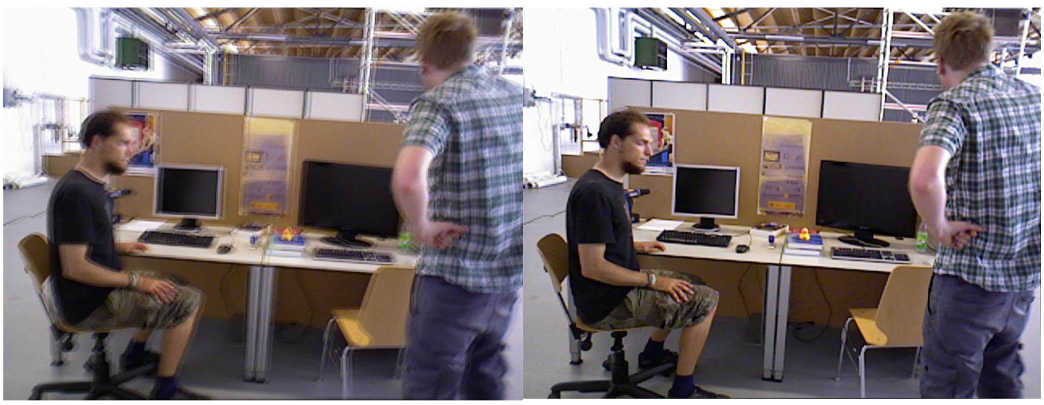


Fig.3. The architecture of DeblurGANv2 network



(a)The blurry image (b)The enhanced image

Fig. 4. Comparison images before and after enhancement using DeblurGANv2

C. ORB feature extraction based on adaptive thresholding

In the ORB-SLAM3 system, ORB (Oriented FAST and Rotated BRIEF) feature points are utilized, which consist of Oriented FAST corners and BRIEF descriptors. FAST corners primarily detect areas with significant local pixel intensity changes, as illustrated in Fig. 5. The feature point extraction process, as described in [15], involves selecting a pixel P in the image with pixel value I_p. The threshold value T is set (e.g., 20% of I_p). Then, 16 pixels are selected around P, utilizing a radius of 3 pixels. If there are N consecutive pixels on the circular path with values exceeding ($I_p + T$) or falling blow (I_p-T), then P can be classified as a feature point. It is customary to set the value of N to 12, a configuration referred to as FAST-12. Other commonly used values for N are 9 and 11, referred to as FAST-9 and FAST-11, respectively). Since FAST corners use a fixed threshold during extraction, only the points with the most significant grayscale differences in the image are selected as corners. This results in the inability to extract other useful feature points within the image. Furthermore, during subsequent feature extraction using quadtree partitioning, the method of preserving the maximum Harris response value results in all corners being concentrated in regions with richer textures. This leads to redundant local feature points. If these feature points gather on dynamic entities, removing dynamic feature points may result in fewer available feature points. In severe cases, this can cause localization failure in the SLAM system.

After the above analysis, selecting the appropriate threshold is essential for extracting ORB features. In order to enhance the robustness of feature point extraction in complex environments for the SLAM system, we adopt an adaptive thresholds selection method that is grounded in KSW entropy value [24], determining the global threshold T_g based on the grayscale distribution of the image. The KSW entropy method refers to calculating the entropy of the grayscale histogram of an image and utilizing conditional probability to describe the distribution of grayscale values for objects and backgrounds in the image, thereby defining the entropy for both the objects and the backgrounds. The method employed in this paper approximates the probability of each grayscale value to represent the likelihood distribution of grayscale values and extracts global grayscale information from the image. The method primarily consists of two steps: first, utilizing the grayscale histogram of the image to provide an approximate estimation of the probability density function of grayscale values. Subsequently, we utilize this function in conjunction with the principles of entropy to construct an objective function for threshold selection. This enables the selection of adaptive thresholds based on the grayscale value distribution of the image, thereby enhancing the adaptability of the ORB feature extraction algorithm.

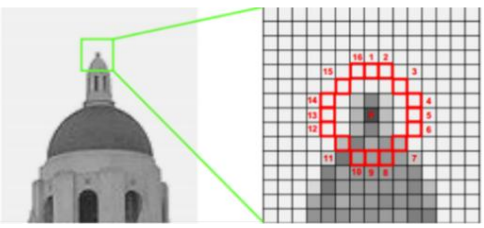


Fig. 5. ORB feature extraction

First, establish a threshold value t to divide the image with grayscale values in the range $[0, L-1]$ into two categories: $[0, t]$ and $[t+1, L-1]$. Let S_1 and S_2 represent the respective pixel probability distributions. Consequently, S1 and S2 can be articulated in the following formulas:

$$S_1 = \{P_0, P_1, P_2, \ldots P_t\} \tag{1}$$

$$S_2 = \{P_{t+1}, P_{t+2}, P_{t+3}, \ldots P_{L-1}\} \tag{2}$$

Where P_i presents the probability of each grayscale level occurrence. Then, let $P_t = P_0 + P_1 + \cdots + P_t$, the entropy of S_1 and S_2 can be articulated as:

$$H(S_1) = -\sum_{i=0}^{t} \frac{P_i}{P_t} \ln \frac{P_i}{P_t} \tag{3}$$

$$H(S_2) = -\sum_{i=t+1}^{L-1} \frac{P_i}{1-P_t} \ln \frac{P_i}{1-P_t} \tag{4}$$

Consequently, the overall entropy of the image can be expressed as the cumulative of the two types of entropy, denoted as:

$$H(S) = H(S_1) + H(S_2) \tag{5}$$

Next, iterate over all grayscale levels t within the range of the grayscale histogram, calculating the sum of entropies for the two classes. Then, the grayscale level corresponding to the maximum value is denoted as T_{max} and the grayscale level corresponding to the minimum value is denoted as T_{min}. Therefore, the global optimal threshold T_g can be represented as:

$$T_g = k \cdot |T_{\max} - T_{\min}| \tag{6}$$

Where k represents the scaling factor. Although the global threshold Tg can be adaptively selected based on the distribution of grayscale values in the image, it assumes that all regions of the image are under the same lighting conditions. When local regions of the image have varying contrasts due to shadows, the global threshold becomes inadequate, leading to a decrease in the effectiveness of feature extraction.

Therefore, in situations where lighting conditions vary significantly, our study adopts a local adaptive threshold T_l to address this issue. Assuming point A (x_0, y_0) is a potential feature point in the image. A square region N, centered at A, is selected with a side length of L. Then, the local adaptive threshold T_l can be represented as:

$$T_l = k \cdot \frac{\left[\frac{1}{n} \sum_{i-1}^{n} I_{i_{\max}} - \frac{1}{m} \sum_{i-1}^{m} I_{i_{\min}} \right]}{I_{i_{aver}}} \tag{7}$$

In equation (7), we define the maximum gray value as I_{imax} and the minimum gray value as I_{imin} in region N. Moreover, I_{iaver} denotes the grayscale average value of region N. The scaling factor k is typically chosen as 3.

D. The Improved YOLOv7 Network

To reduce the influence of dynamic objects present in real-world environments on the SLAM system, our paper improves the real-time target detection algorithm YOLOv7 [25] to enhance its detection accuracy. Subsequently, it is utilized to identify the positions of dynamic objects in images and extract semantic information.

YOLOv7 algorithm, as a typical representative of One-Stage target detection algorithm [26], has good real-time performance, and its framework architecture is illustrated in Fig. 6. First, YOLOv7 introduces reparameterization (RepConv) [27] in the network structure, which decreases the parameter count and computational demands, thereby improving the network's operational efficiency. Then, YOLOv7 improved ELAN by proposing E-ELAN (Extended-ELAN), which utilizes extension, stochastic disruption, and merging bases to achieve continuous enhancement of the network's learning capabilities while preserving the integrity of the original gradient pathways. Also, E-ELAN can instruct different computational modules to learn more different features. The label assignment methodology employed by YOLOv7 integrates the cross-grid search technique utilized in YOLOv5, along with the matching strategy adopted in YOLOx. In addition to this, the training approach incorporating an auxiliary head is used in YOLOv7, which enhances the accuracy by elevating the training expenses and does not affect the inference time.

Although YOLOv7 runs faster, its detection accuracy is lower, so this paper introduces SimAM (Spatial information Attention Mechanism) [28] in YOLOv7 to enhance its precision.

The Attention Mechanism is a technique widely employed in computer science and machine learning, where the output of each neuron is influenced not only by the outputs of all neurons in the preceding layer but also weighted based on various aspects of the input data. This enables the network to focus more on the specific information within the input sequence, thereby enhancing both the accuracy and efficiency of the model. SimAM is a powerful attentional mechanism that has been widely used in the domain of computer vision, including target detection, image segmentation, and image generation. Different from existing channel and spatial attention modules, SimAM generates 3D attention weights for feature maps without the need to introduce supplementary parameters. An additional benefit of the module is that majority of the operations are predicated on the selection of specified energy functions, reducing the necessity for structural alterations. SimAM has the following features:

1) SimAM is able to model relationships between features at multiple levels, and it can focus on both low-level features and high-level semantic features, enabling the model to understand the input data more comprehensively.
2) SimAM not only focuses on the features of each channel, but is also able to mine the relationships between different channels. This attention mechanism enables SimAM to better learn the semantic connections between features and improves the model's representational capabilities.
3) SimAM introduces spatial information and focuses on the spatial distribution of features. This attention mechanism enhances the model's ability to precisely localize the target region, increasing the precision of target detection and segmentation.
4) SimAM is adaptive in that it is able to automatically adjust its focus of attention according to the different characteristics of the input data. This adaptability allows SimAM to be applied to a variety of different types of data, improving the module's capacity for generalization. As illustrated in Fig. 7, SimAM estimates three-dimensional weights compared to the channel attention module and the spatial attention module.

Attention mechanism is a plug-and-play module, in order to increase the detection precision of YOLOv7 we inserts the SimAM module on the feature extraction reinforcement network of YOLOv7 as illustrated in Fig. 8.

Fig. 9 illustrates the effect of the improved YOLOv7 algorithm, indicating that the modified YOLOv7 is able to detect the target in the image more accurately.

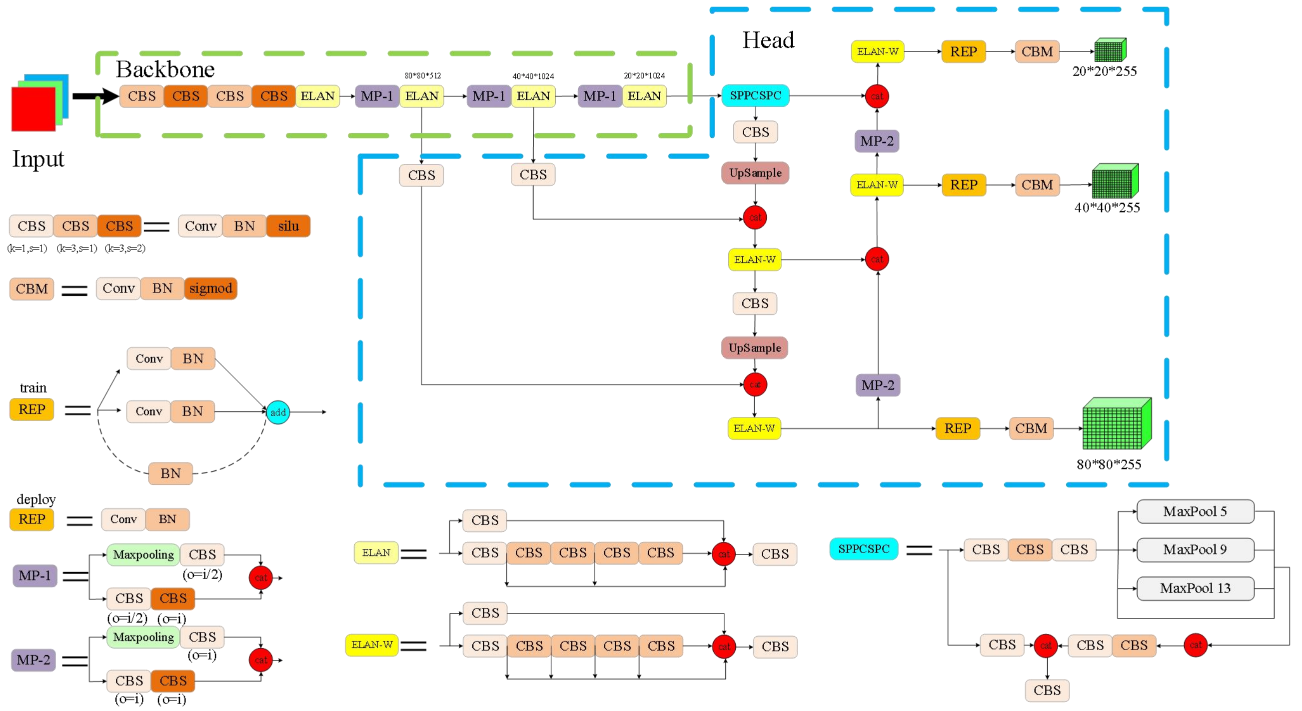


Fig. 6. YOLOv7's network structure

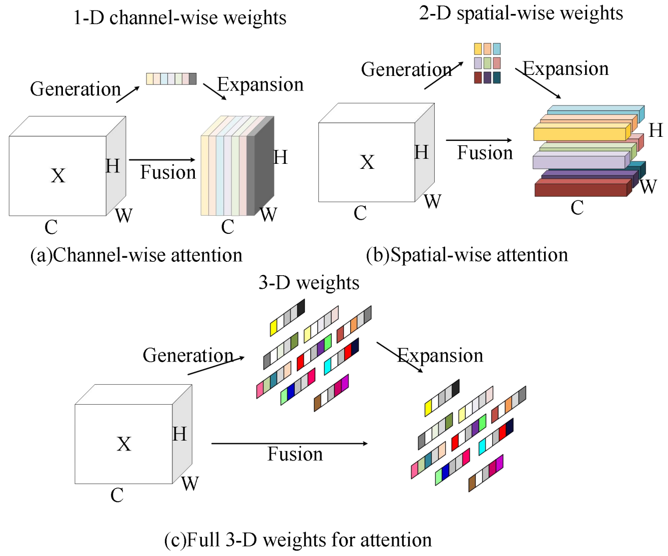


Fig. 7. Comparison of different attention types

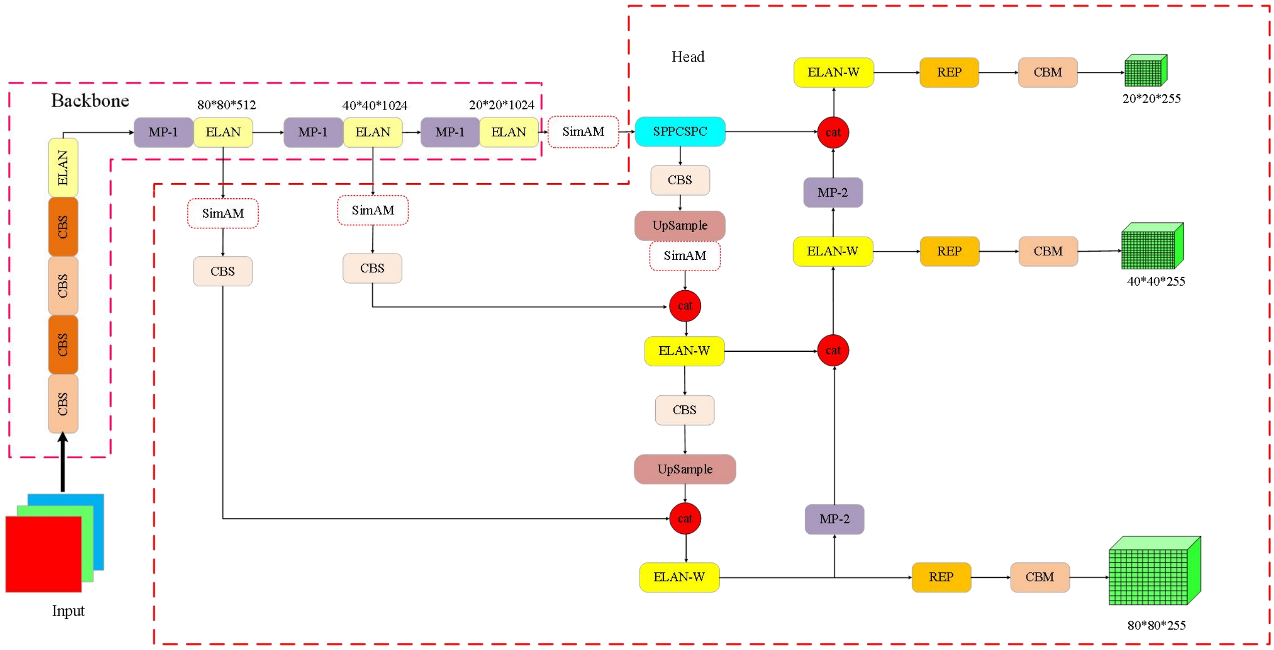
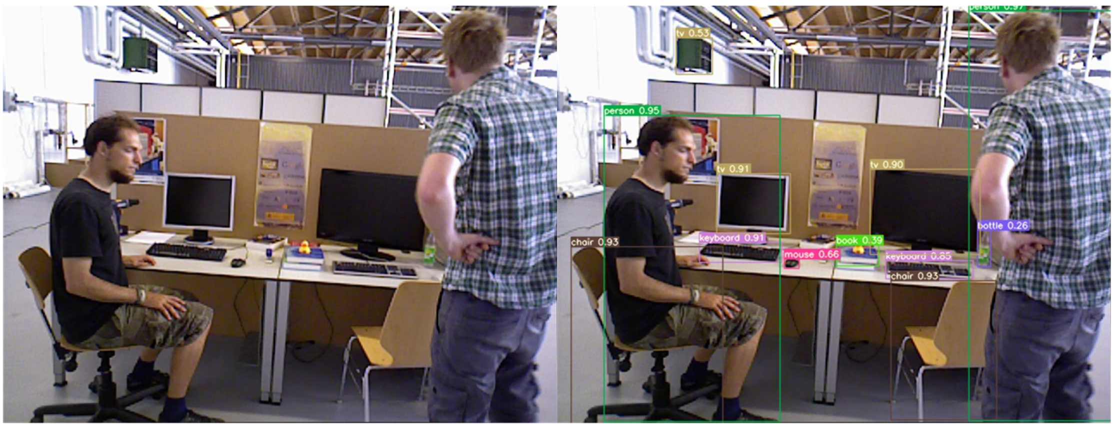


Fig. 8. Comparison of different attention types



(a) (b)

Fig. 9. Target detection results. Figure (a) shows the input image and Figure (b) shows the detection result

E. Target Detection Combined with Eippolar Line Constraint to Reject Dynamic Feature Points

Previous SLAM methods based on target detection relied exclusively on these results to eliminate dynamic feature points. However, target detection networks often struggle to ascertain whether inherently mobile objects, such as cars, are currently in motion. Consequently, even when a car remains stationary in the environment, the system may erroneously discard associated feature points, markedly reducing the pool available for pose estimation. Moreover, these approaches frequently fail to filter out feature points attributed to static objects like books or chairs that are being displaced by individuals, resulting in inaccurate data associations and a significant degradation in SLAM system precision.

In order to tackle these challenges, we integrate target detection with an epipolar constraint approach to eliminate dfp. First, we need to align the feature points from two consecutive frames and use them to calculate the fundamental matrix. Next, we can assess the distance between the feature points in the current frame and their related epipolar lines. Greater distances indicate a higher likelihood of a dynamic feature point.

For achieving an accurate fundamental matrix, this study identifies feature points designated as static targets using semantic information. The fundamental matrix (F) between two frames is then computed employing a seven-point method within a RANSAC framework. According to the pinhole camera model, as illustrated in Fig. 10, O_1 and O_2 denote the optical centers of a camera. The movement of camera between successive frames, M_1 and M_2, is described by R and t. P represents an arbitrary point in space. P_1 and P_2 denote the projections of the point P on M_1 and M_2, l_1 and l_2 denote the two polar lines.

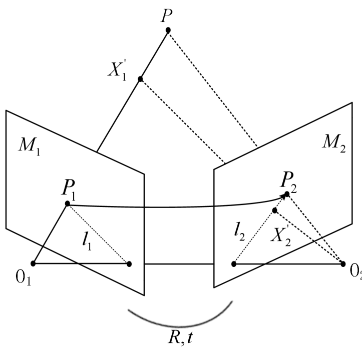


Fig. 10. Epipolar constraint

The chi-square coordinates of P_1 and P_2 can be expressed as:

$$P_1 = [x_1, y_1, 1]\, , P_2 = [x_2, y_2, 1] \tag{8}$$

In which x and y represent the pixel coordinate values of the feature points, the polar line l_2 can be expressed as:

$$l_2 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = F \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{9}$$

X, Y, Z in Eq. 9 denote the line vectors. As indicated in the Literature [29], we express the epipolar constraint as:

$$p_2^T F p_1 = p_2^T l_2 = 0 \tag{10}$$

Then, we express the distance between p_2 and the epipolar line as d, which is also known as the offset dist. It is calculated as follows:

$$d = \frac{\left| p_2^T F p_1 \right|}{\sqrt{X^2 + Y^2}} \tag{11}$$

From Eq. 11, in an ideal scenario, when $d = 0$, the feature point p_2, representing the current frame, is situated on l_2. Therefore, it may be considered static.

However, in practice, the presence of various forms of noise typically results in the offset dist exceeding 0, yet remaining below a defined empirical threshold, denoted as ε. In this paper, the threshold is selected as 0.6, and when d is less than ε, we regard this feature point as static.

To summarize, the steps for removing dynamic feature points in our study are as follows:

Step 1: We employ the improved YOLOv7 to identify various targets in deblurred images and ascertain their positions within these images, while also delineating detection boxes. In addition, we extract semantic information about the targets to provide necessary data for subsequent tasks.

Step 2: We need to determine whether those feature points in the dynamic object detection frame satisfy epipolar constraint. If the feature points in the detection frame don't satisfy epipolar constraint, they are considered as dynamic feature points and are no longer used in the subsequent tracking threads.

IV. EXPERIMENTS

In this section, we assess the efficacy of our system by conducting evaluations employing the TUM dataset. Subsequently, we compare the outcomes with those obtained from ORB-SLAM3. In addition, this paper also compares with advanced VS algorithms that utilize object detection in dynamic surroundings such as AHY-SLAM and RDS-SLAM. All experiments in this paper were conducted on a computer system featuring an Intel i5 CPU, RTX1050Ti GPU, and 16GB of memory.

A. TUM dataset and evaluation metrics

The TUM RGB-D dataset [30], provided by the Computer Vision Group at the Technical University of Munich, represents a large-scale resource that has set a new standard for evaluating SLAM systems. We utilized five data packages that encompass a significant number of dynamic objects to test our system's performance, respectively fr3/walking/xyz, fr3/walking/half, fr3/walking/static, fr3/walking/rpy, and fr3/sitting/static. The initial four data packets include a greater number of dynamic objects, whereas the final data packet has a smaller quantity of dynamic objects.

We utilize two metrics to assess the effectiveness of the method we proposed: Absolute Trajectory Error (ATE), which measures the disparity between estimated and ground truth trajectories, and Relative Pose Error (RPE), utilized to quantify rotational drift and translational drift.

B. Experimental Results Analysis

This paper presents improvements upon ORB-SLAM3. To illustrate the benefits of our system, we conducted a comparative analysis of the experimental data acquired from our system against these obtained from ORB-SLAM3, as shown in Table I to III. In these tables, RMSE (Root Mean Square Error) quantifies the disparity between predicted and actual values. A lower RMSE signifies closer approximation to the true values. S.D. (Standard Deviation) gauges the spread of values within a dataset, with a smaller standard deviation indicating reduced variability and enhanced system stability. Tables I through III demonstrate that our system outperforms ORB-SLAM3 by over 91% in most high-dynamic sequences, confirming the superior performance of our system in highly dynamic environments. The performance in low dynamic surroundings is slightly inferior to ORB-SLAM3, possibly due to our system's dynamic feature point removal process inadvertently discarding some useful static feature points.

Fig. 11, 12, and 13 illustrate the estimated trajectories and ground truth trajectories for both systems across the five data packages. Dashed lines represent the actual trajectories, while solid lines depict the estimated trajectories. If the estimated trajectory is closer to the true trajectory on the ground, it indicates that the system's tracking and positioning performance is better. From the figure, it is evident that in dynamic environments, the trajectory estimated by our system closely matches the ground truth trajectory. Therefore, the proposed system demonstrates superior tracking and localization performance.

To further validate the advancements of our system, we compared it with recently published SLAM algorithms, such as RDS-SLAM and AHY-SLAM. As shown in Table IV, our system exhibits significantly lower RMSE and S.D. across the five sequences compared to these algorithms, demonstrating its superior performance and advanced capabilities. As a crucial component of mobile robotics, real-time capability is essential for SLAM systems. Hence, we also measured the runtime of the SLAM systems. The runtime of each SLAM system is illustrated in Table V.

As indicated in Table V, we can see that our system exhibits better real-time performance compared to RDS-SLAM and AHY-SLAM, averaging 75ms per frame. However, there is an increase in processing time of 13ms compared to ORB-SLAM3, which is attributed to the inclusion of the image enhancement module and the thread for the removal of dfp. In summary, our system demonstrates good real-time performance and is applicable for deployment in real-world environments.

TABLE I. Comparison of ATE between ORB-SLAM3 and Ours in TUM sequences

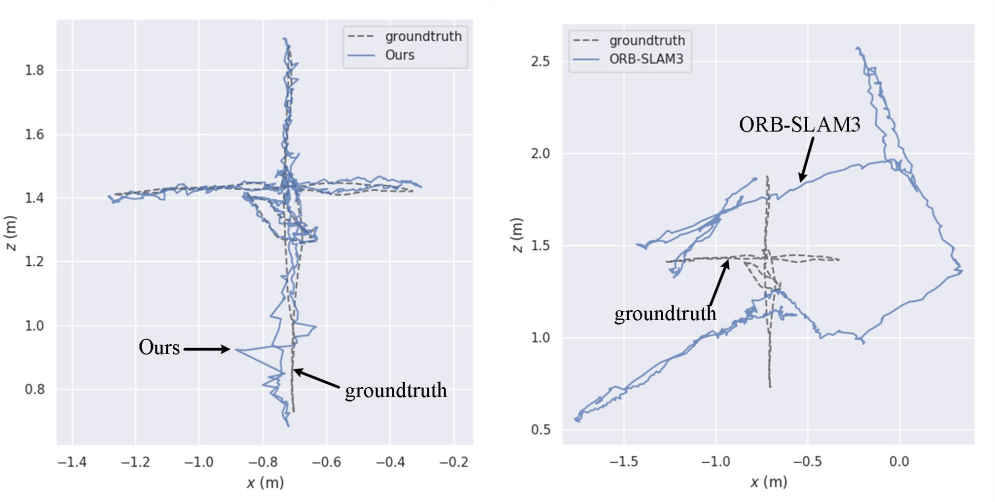
Sequence	ORB-SLAM3		Ours		Improvement/%	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3_walking_xyz	0.7126	0.3672	0.0163	0.0086	97.71	97.65
fr3_walking_half	0.3942	0.2854	0.0207	0.0158	94.47	94.46
fr3_walking_static	0.4001	0.0640	0.0072	0.0038	98.2	94.06
fr3_walking_rpy	0.4223	0.3321	0.0443	0.0367	89.52	88.96
fr3_sitting_static	0.0075	0.0044	0.0078	0.0045	-4.3	-3.2

TABLE II. Results of metric rotational drift (RPE)

Sequence	ORB-SLAM3		Ours		Improvement/%	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3_walking_xyz	6.2841	3.4841	0.2928	0.1972	95.34	94.34
fr3_walking_half	6.8735	5.4233	0.4728	0.4598	93.12	91.52
fr3_walking_static	2.7134	2.2098	0.2374	0.2134	91.25	90.34
fr3_walking_rpy	5.3785	3.4785	0.7142	0.5725	86.72	83.54
fr3_sitting_static	0.1687	0.0087	0.1781	0.0094	-5.6	-8.2

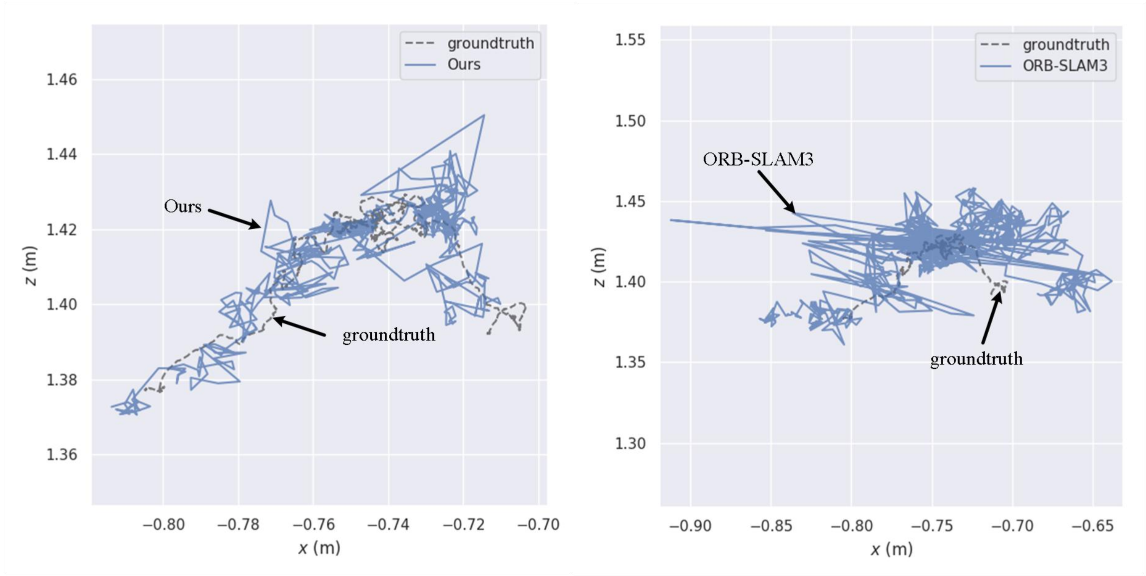
TABLE III. Results of metric translational drift (RPE)

Sequence	ORB-SLAM3		Ours		Improvement/%	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3_walking_xyz	0.3523	0.2243	0.0170	0.0117	95.16	94.78
fr3_walking_half	0.2879	0.2165	0.0219	0.0170	92.38	92.14
fr3_walking_static	0.1744	0.1543	0.0044	0.0027	97.44	98.23
fr3_walking_rpy	0.3728	0.2631	0.0438	0.0354	88.24	86.53
fr3_sitting_static	0.0081	0.0042	0.0084	0.0044	-4.2	-2.7

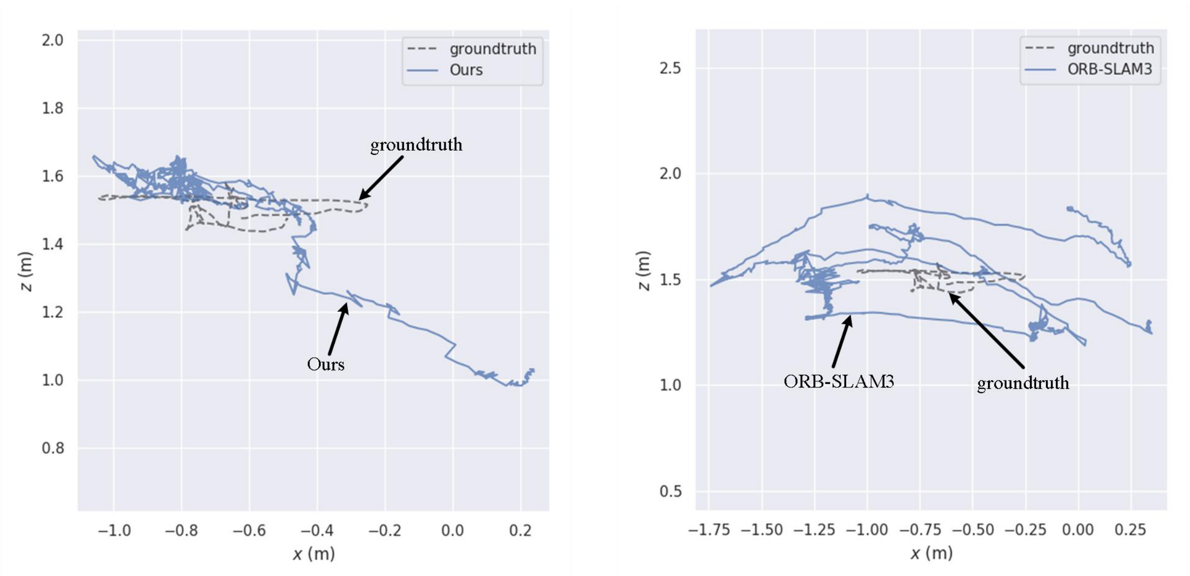


(a)fr3/walking_xyz

Fig. 11. Comparison chart of estimated trajectory and actual trajectory

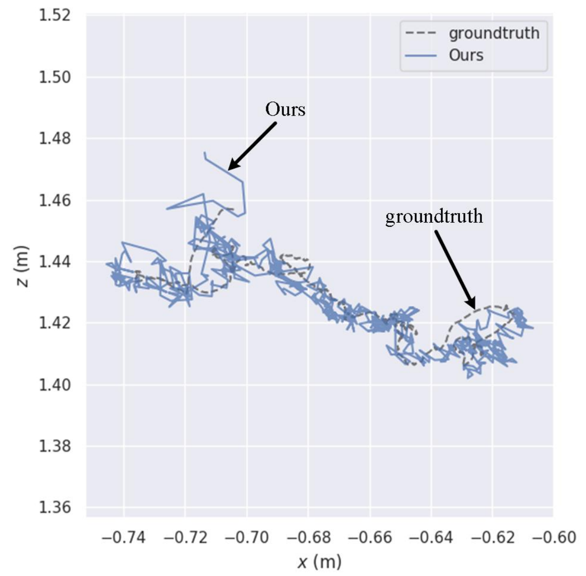


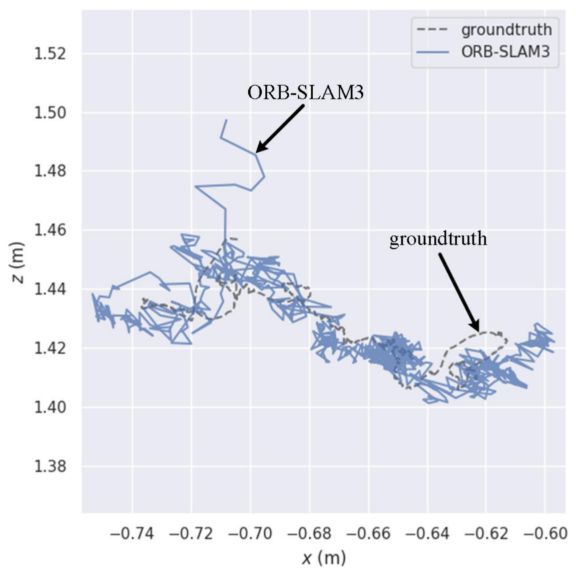
(c)fr3/walking_static



(d)fr3/walking_rpy

Fig. 12. Comparison chart of estimated trajectory and actual trajectory





(e)fr3/sitting_static

Fig. 13. Comparison chart of estimated trajectory and actual trajectory

TABLE IV. Comparison of absolute trajectory errors between Ours and other similar SLAM methods.(ATE)

Sequence	RDS-SLAM		AHY-SLAM		Ours	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3_walking_xyz	0.0571	0.0229	0.0182	0.0098	0.0163	0.0086
fr3_walking_half	0.0807	0.0454	0.0321	0.0174	0.0207	0.0158
fr3_walking_static	0.0206	0.0120	0.0081	0.0043	0.0072	0.0038
fr3_walking_rpy	0.1604	0.0873	0.1938	0.1588	0.0443	0.0367
fr3_sitting_static	0.0084	0.0043	0.0089	0.0049	0.0069	0.0041

TABLE V. Time analysis

Systems	Average Processing Time Per Frame(ms)
ORB-SALM3	62
Ours	75
AHY-SLAM	103
RDS-SLAM	82

V. Conclusion

This paper introduces a real-time semantic SLAM system integrating an enhanced YOLOv7 network. Initially, we introduce an image enhancement module into the ORB-SLAM3 framework, leveraging the DeblurGANv2 network to deblur camera-captured images, thereby enhancing the quality of image frames. Subsequently, we incorporate a thread dedicated to the detection of objects into the system. We refine the YOLOv7 network to optimize its performance and integrate it within the object detection thread to obtain object positions and semantic information from the image frames. Additionally, we introduce an adaptive threshold in the traditional ORB feature extraction method to bolster feature extraction capabilities, laying a robust foundation for pose estimation. Finally, to mitigate the influence of dynamic feature points on system performance, We integrate the outputs from the object detection process with epipolar constraints in order to eliminate dynamic the dfp. Experimental results show that our proposed system achieves over 90% improvement in accuracy compared to some existing SLAM algorithms. Furthermore, by utilizing the enhanced YOLOv7 algorithm, our system enhances precision while simultaneously preserving real-time operational efficiency.

Although the system proposed in this paper has numerous advantages, it still exhibits limitations. For instance, during the elimination of the dfp, some useful sfp may inadvertently be discarded, leading to decreased accuracy. Furthermore, the system exhibits competent functionality in indoor settings where the dynamic objects is minimal. However,its efficacy diminishes when faced with numerous fast-moving objects in outdoor settings. To address the issue of mistakenly removing static feature points, we will continue researching more rational methods for dynamic feature point removal, such as employing advanced selection algorithms. Additionally, we will investigate techniques to enhance the system's ability to detect a high volume of fast-moving objects, such as further improving the structure of the detection network to enhance its performance.

REFERENCES

[1] Smith, Randall C., and Peter Cheeseman. "On the representation and estimation of spatial uncertainty." The International Journal of Robotics Research 5.4 (1986): 56-68.

[2] Chen, Weifeng, et al. "An overview on visual slam: From tradition to semantic." Remote Sensing 14.13 (2022): 3010.

[3] Seok, Hochang, and Jongwoo Lim. "ROVINS: Robust omnidirectional visual inertial navigation system." IEEE Robotics and Automation Letters 5.4 (2020): 6225-6232.

[4] Zhai, Chaoyang, et al. "Robust vision-aided inertial navigation system for protection against ego-motion uncertainty of unmanned ground vehicle." IEEE Transactions on Industrial Electronics 68.12 (2020): 12462-12471.

[5] Li, Peiliang, et al. "Monocular visual-inertial state estimation for mobile augmented reality." 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). IEEE, 2017.

[6] Gupta, Abhishek, and Xavier Fernando. "Simultaneous localization and map** (slam) and data fusion in unmanned aerial vehicles: Recent advances and challenges." Drones 6.4 (2022): 85.

[7] Yin, Hesheng, et al. "Dynam-SLAM: An accurate, robust stereo visual-inertial SLAM method in dynamic environments." IEEE Transactions on Robotics 39.1 (2022): 289-308.

[8] Nistér, David. "Preemptive RANSAC for live structure and motion estimation." Machine Vision and Applications 16.5 (2005): 321-329.

[9] Campos, Carlos, et al. "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam." IEEE Transactions on Robotics 37.6 (2021): 1874-1890.

[10] Yu, Chao, et al. "DS-SLAM: A semantic visual SLAM towards dynamic environments." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.

[11] Bescos, Berta, et al. "DynaSLAM: Tracking, map**, and inpainting in dynamic scenes." IEEE Robotics and Automation Letters 3.4 (2018): 4076-4083.

[12] Yuan, Xun, and Song Chen. "Sad-slam: A visual slam based on semantic and depth information." 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020.

[13] Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." IEEE Transactions on Pattern Analysis and Machine Intelligence 39.12 (2017): 2481-2495.

[14] He, Kaiming, et al. "Mask r-cnn." Proceedings of The IEEE International Conference on Computer Vision. 2017.

[15] Kupyn, Orest, et al. "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better." Proceedings of The IEEE/CVF International Conference on Computer Vision. 2019.

[16] Rublee, Ethan, et al. "ORB: An efficient alternative to SIFT or SURF." 2011 International conference on computer vision. Ieee, 2011.

[17] Voulodimos, Athanasios, et al. "Deep learning for computer vision: A brief review." Computational Intelligence and Neuroscience 2018 (2018).

[18] Girshick, Ross. "Fast r-cnn." Proceedings of The IEEE International Conference on Computer Vision. 2015.

[19] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[20] Zhong, Fangwei, et al. "Detect-SLAM: Making object detection and SLAM mutually beneficial." 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2018.

[21] Schorghuber, Matthias, et al. "SLAMANTIC-leveraging semantics to improve VSLAM in dynamic environments." Proceedings of The IEEE/CVF International Conference on Computer Vision Workshops. 2019.

[22] Kupyn, Orest, et al. "Deblurgan: Blind motion deblurring using conditional adversarial networks." Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[23] Mijwil, Maad M., et al. "MobileNetV1-Based Deep Learning Model for Accurate Brain Tumor Classification." Mesopotamian Journal of Computer Science 2023 (2023): 32-41.

[24] Bei, Qiancheng, et al. "An Improved ORB Algorithm for Feature Extraction and Homogenization Algorithm." 2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI). IEEE, 2021.

[25] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[26] Liu, Wei, et al. "Ssd: Single shot multibox detector." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016.

[27] Ding, **aohan, et al. "Repvgg: Making vgg-style convnets great again." Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[28] Yang, Lingxiao, et al. "Simam: A simple, parameter-free attention module for convolutional neural networks." International Conference on Machine Learning. PMLR, 2021.

[29] Kundu, Abhijit, K. Madhava Krishna, and Jayanthi Sivaswamy. "Moving object detection by multi-view geometric techniques from a single camera mounted robot." 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009.

[30] Sturm, Jürgen, et al. "A benchmark for the evaluation of RGB-D SLAM systems." 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012.