IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 February 2024, accepted 14 March 2024, date of publication 19 March 2024, date of current version 28 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3379269

APPLIED RESEARCH

DRV-SLAM: An Adaptive Real-Time Semantic Visual SLAM Based on Instance Segmentation Toward Dynamic Environments

QIANG JI, ZIKANG ZHANG, YIFU CHEN, AND ENHUI ZHENG
School of Mechanical and Electrical Engineering, China Jiliang University, Hangzhou 310018, China
Corresponding author: Enhui Zheng (ehzheng@cjlu.edu.cn)

ABSTRACT Traditional simultaneous localization and mapping (SLAM) methodologies predominantly rely on the assumption of a static environment. This constraint limits the applicability of most visual SLAM systems in various real-world scenarios. In this paper, we introduce a real-time semantic visual SLAM algorithm tailored for complex dynamic environments (DRV-SLAM). DRV-SLAM leverages image analysis to identify potential moving objects and determine their current motion states. By dynamically adjusting the rejection of unreliable dynamic feature points based on the proportion of potential moving objects in the environment, DRV-SLAM significantly enhances the system's localization accuracy and robustness in complex dynamic environments. Additionally, DRV-SLAM employs a dense mapping approach that combines global downsampling and targeted object data enhancement. This method effectively reduces the memory footprint of dense point cloud maps, enabling DRV-SLAM to efficiently construct large-scale dense point cloud maps in diverse scenarios. Experimental results show that in a highly dynamic environment, the DRV-SLAM algorithm shows an order of magnitude performance improvement compared to the traditional ORB-SLAM2 algorithm. The performance index of absolute trajectory error is significantly improved by more than 98%, and DRV-SLAM is currently one of the most real-time, accurate and robust systems for dynamic scenes.

INDEX TERMS Visual simultaneous localization and mapping (SLAM), dynamic environment, motion states, mapping.

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) addresses the challenge of estimating one's own position and constructing a map using sensory data from the environment, solely relying on the sensors carried, without prior knowledge of the surroundings. Modern visual SLAM frameworks are well-established and typically consist of several modules: sensor data acquisition, front-end visual odometry, back-end nonlinear optimization, loop closure detection, and map creation. Furthermore, some advanced visual SLAM algorithms have demonstrated superior performance, such as ORB-SLAM2 [1], LSD-SLAM [2], and others [3], [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Laura Celentano.

However, these vision SLAM algorithms have been primarily studied under the assumption of a static environment. While motion constraints can be used to treat feature points on a small portion of dynamic objects as anomalous static points, when the surrounding environment contains a higher number of dynamic objects, it leads to significant drift in self-pose estimation. In reality, the environments encountered by robots are complex and dynamic, making it challenging to ensure constant static conditions. Therefore, vision SLAM algorithms based on static assumptions limit their application in scenarios involving service robots, autonomous driving, Augmented Reality/Virtual Reality(AR/VR), and other complex dynamic environments.

In addition, dynamic environments pose a significant challenge to the robustness of visual SLAM systems. When

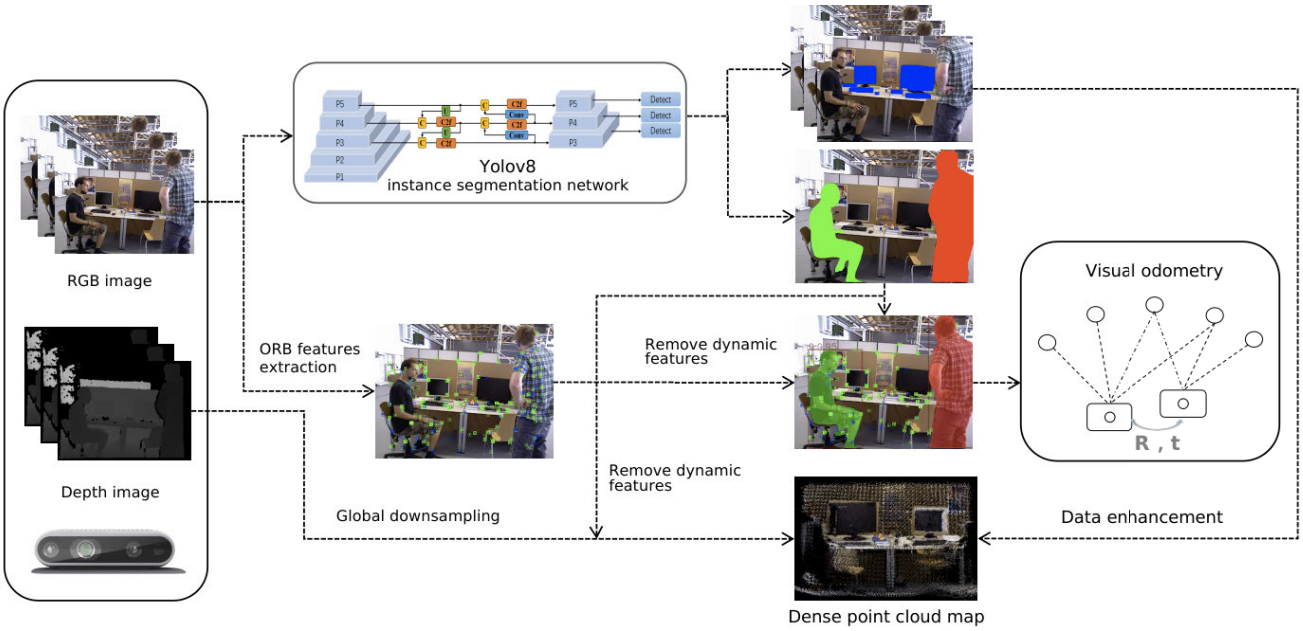

FIGURE 1. The overview of DRV-SLAM. The raw RGB image is utilized to instance segmentation and motion region constraint simultaneously. Then remove outliers and estimate pose. Dense map is built in an independent thread based on the pose, depth image, and instance segmentation results.

a large portion of the extracted features in the system originates from moving objects, utilizing these features for position estimation not only significantly increases the errors in position estimation but also compromises system stability, potentially leading to system failures. Experiments have demonstrated that when ORB-SLAM2 [1] is applied to datasets with dynamic environments, such as the TUM RGB-D DynamicObjects dataset [5], the system frequently encounters failures, resulting in unsuccessful and incomplete operation.

This paper introduces a method that combines target detection, instance segmentation, and motion region constraints, utilizing both semantic a priori information and geometric instance segmentation information. The aim is to mitigate the impact of dynamic objects in the visual SLAM system, thereby enhancing the system's localization accuracy and robustness in complex dynamic environments. Furthermore, the paper offers sparse and dense point cloud maps to cater to robots in various scenarios. An overview of the DRV-SLAM system is illustrated in Fig. 1.

The main contributions of this paper include:

1) DRV-SLAM is a complete adaptive real-time semantic visual SLAM system based on the ORB-SLAM2 [1] framework. It is specially designed for complex dynamic environments and can dynamically adjust the feature extraction strategy according to the surrounding environment. Improving the pose estimation accuracy and robustness of the system. The performance of the system is evaluated on the TUM RGB-D DynamicObjects dataset [5]. The results show that DRV-SLAM is significantly better than ORB-SLAM2 [1] and ORB-SLAM3 [3] in terms of accuracy and robustness of pose estimation in highly dynamic environments. Its performance is superior to other advanced visual dynamic SLAM systems.

2) This paper proposes a method for partitioning potential moving objects into motion states, which can classify perceived potential moving objects into those in motion and those in a static state. When the proportion of potential moving objects in the surrounding environment is high, the system only removes feature points on highly dynamic potential moving objects. This method significantly improves the robustness and accuracy of the system in dynamic scenes. We also employed TensorRT for optimizing the deployment of the instance segmentation network, thereby enhancing the inference speed post-deployment, ensuring real-time compliance for the instance segmentation network.

3) Additionally, this paper presents a dense map-building method employing global downsampling and targeted object data enhancement. This approach reduces the system memory utilization for dense point cloud maps, thereby enabling the DRV-SLAM system to create dense point cloud maps across a broader range of scenarios. Consequently, the DRV-SLAM system can function in a more extensive array of scenarios.

II. RELATED WORKS

Dealing with the impact of dynamic objects on the system in visual SLAM has been a widely researched topic in recent years. Research ideas can be broadly categorized into two types: one relies on traditional computer vision methods for dynamic point detection, such as motion estimation, optical flow computation, multi-view geometry, and so on. The other approach involves the use of deep learning methods for dynamic object detection.

The former approach typically involves detecting moving objects or regions through the analysis and modeling of pixel or feature point motion in an image sequence.

- Kundu et al. [6] define geometric constraints by constructing basis matrices that are considered dynamic if matching features in subsequent frames are far from the poles.
- Zou and Tan [7] project features from the previous frame to the current frame by analyzing the triangulation consistency and calculating the reprojection error of the feature tracking; if the error is large, the map points are considered dynamic.
- Wang et al. [8] designed a moving target discrimination model based on statistical features by clustering the depth images into several objects, counting the number and percentage of features on each object, and eliminating all features on the model if the target is considered to be moving.

However, this type of method cannot acquire the semantic a priori information for individual pixels, potentially resulting in incorrect detections or missed detections, particularly in environments characterized by complex motion patterns, noise, occlusions, and dynamic lighting conditions. Consequently, it fails to deliver precise dynamic object detection.

Deep learning-based methods enable machines to learn higher-level feature representations, typically resulting in more accurate detection outcomes. As a result, the current mainstream approach in the field of SLAM for dynamic object detection involves the use of deep learning methods. Notably, methods like DynaSLAM and others [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20] all employ deep learning strategies for motion object recognition.

- DynaSLAM [9] uses multi-view geometry to help the Mask R-CNN [21] instance segmentation network to perform segmentation in a way to get better segmentation results, to detect the moving objects in the image and repair the background around the moving objects.
- Detect-SLAM [10] by performing target detection only on keyframes and propagating the movement probability through feature matching and match point expansion The method solves the real-time problem by removing feature points on potentially moving objects, assigning an ID to each object and constructing an object model in the map.
- DS-SLAM [11] uses SegNet [22] semantic segmentation network in combination with the optical flow method to remove dynamic feature points and construct an octree semantic map.
- YOLO-SLAM [23] uses the YOLOv3 [24] target detection network to monitor a priori potential moving objects in the scene, and then combines it with the RANSAC [25] method to eliminate feature points within the dynamic object range.
- SG-SLAM [26] utilizes the SSDlite [27] target detection network in combination with epipolar constraints between image frames to discern motion objects.

However, when dealing with moving objects, most of their work focuses on eliminating the feature points on the detected potential moving objects, and the research on judging whether the potential moving objects are moving is not in-depth enough. Some of these methods, outliers detected by the optical flow method are used to determine whether the segmented potential moving objects are moving. However, due to the influence of light changes, noise, etc., it is easy to misjudge the potential moving objects as moving. When there are many potential moving objects in the surrounding environment, if it cannot be accurately determined whether they are moving, directly eliminating feature points on potential moving objects will waste many useful feature points, thus reducing the robustness of the SLAM system. Especially when potential moving objects occupy a high proportion in the image, it is easy to cause the SLAM system to fail to track.

In addition, there are research efforts dedicated to designing robust methods for dynamic SLAM based on filters and probabilistic models, effectively enhancing the robustness of SLAM algorithms in dynamic environments.

- Demim et al. [28] proposed a method based on Adaptive Smoothed Variable Structure Filter (SVSF) to address cooperative SLAM problems. By introducing a covariance matrix to evaluate the uncertainty of adaptive SVSF, the method enhances its performance and expands its useful applications, effectively improving system robustness when encountering complex environments.
- Tang et al. [29] proposed an improved H-Infinity unscented FastSLAM (IHUFastSLAM) with adaptive genetic resampling. The H-Infinity unscented Kalman filter algorithm was enhanced using an adaptive factor and applied as importance sampling in particle filtering. Subsequently, process noise and measurement noise were estimated using a time-varying noise estimator. Additionally, an adaptive genetic algorithm was employed for particle filter resampling. Finally, the improved IHUFastSLAM with adaptive genetic resampling was introduced for robot tracking. The proposed algorithm enables accurate robot tracking with robustness in complex environments.
- Zhang et al. [30] proposed a semi-supervised point cloud registration (PCR) method for accurately estimating point correspondences and handling large-scale transformations using limited prior datasets. The method treats two point clouds as implementations of Gaussian Mixture Models (GMM), and PCR between the two point clouds is achieved by minimizing the KL divergence between these probability distributions. Subsequently, an augmented regression network is employed to estimate the correspondence between the point clouds and latent GMM components. Finally, the parameters of the GMM are updated based on the correspondence, and the transformation matrix is computed using weighted singular value decomposition (SVD).

Extensive experiments on synthetic and real-world data validated the superior performance of the proposed method compared to state-of-the-art registration methods. These experiments also highlighted the method's advantages in terms of accuracy, robustness, and generalization.

Although methods based on filters and probabilistic models can improve robustness and stability to some extent, they often involve complex numerical optimization problems that require significant computational resources and time. Moreover, in practical applications, they are affected by factors such as environmental changes, sensor noise, and model errors, making it difficult to establish accurate mathematical models and leading to poor generalization ability. Therefore, their practical application in the field of SLAM is subject to many limitations.

III. SYSTEM INTRODUCTION

In this section, the framework of the DRV-SLAM system is described in detail. This section contains five aspects, firstly, we introduce the DRV-SLAM system overview diagram, secondly, we briefly introduce the YOLOv8 real-time instance segmentation network used in DRV-SLAM and the improvements we have made, then we introduce our proposed method for motion region constraint, then we show how to perform dynamic feature point culling, and lastly we introduce the method of constructing a dense map.

A. FRAMEWORK OF DRV-SLAM

The DRV-SLAM system builds upon the existing framework of ORB-SLAM2 [1] and introduces several enhancements, including an instance segmentation module, dynamic feature point filtering module, and dense mapping module. During system operation, five threads run in parallel: the tracking thread, instance segmentation thread, local mapping thread, loop detection thread, and dense mapping thread. The overall system architecture is illustrated in Figure 1.

After entering the image data input system, it undergoes parallel processing and is simultaneously sent to the tracking thread and the instance segmentation thread. In the instance segmentation thread, the YOLOv8 instance segmentation network performs target detection and segmentation on the images, outputting object detection boxes, and object area masks. Meanwhile, in the tracking thread, feature points are extracted from the images, awaiting the segmentation results from the instance segmentation thread. Subsequently, utilizing the instance segmentation results in combination with the method of motion region constraint, the potential moving objects in the images are categorized based on their motion levels. The range of motion levels for feature point removal is dynamically adjusted according to the proportion of pixels occupied by potentially moving objects in the image. The retained feature points are used for subsequent pose estimation and optimization. In the dense map-building thread, dynamic points in the depth map are first removed using the instance segmentation results. Then, a dense point cloud map is constructed using global downsampling and targeted object data enhancement methods.

B. INSTANCE SEGMENTATION NETWORK

Currently, target detection and semantic segmentation methods based on convolutional neural networks continue to make breakthroughs in speed and accuracy. Object detection algorithms need to classify different objects in the image and also need to give the location of each object. Traditional target detection methods, such as R-CNN [31] series (such as SPP-Net [32], Fast R-CNN [33], Faster R-CNN [34], etc.), using convolutional neural networks to automatically learn features, avoiding the limitations of manually designing features. They achieve target detection through multiple steps such as candidate frame extraction, feature extraction and classification. Although these methods have high accuracy, they are time-consuming because candidate box extraction and target classification are performed separately.Traditional semantic segmentation networks also use convolutional neural networks to learn features. Different from target detection, the goal of semantic segmentation is to assign each pixel in the image to a specific semantic category, such as FCN [35], SegNet [22], U-Net [36], etc., but due to the use of a fully convolutional structure, the computational complexity is high, and they are also not suitable for real-time applications.

In contrast, the YOLO [37] algorithm abandons the intermediate step of generating candidate regions, directly processes the regression problem of each bounding box through a single convolutional neural network, and predicts the probability of the corresponding category, thereby achieving high speed and Very good accuracy is maintained. YOLOv8 provides a new SOTA model that can achieve target detection and instance segmentation at the same time. DRV-SLAM uses the YOLOv8 instance-level pixel semantic segmentation network. It can segment 80 types of objects by training on the MS COCO data set [38], and deploys and optimizes the model through TensorRT. Testing on a computer equipped with a GTX 1050ti graphics card, the inference speed reaches 42 frames per second, meeting the requirements for real-time detection.

C. MOTION REGION CONSTRAINT

The inference results of instance segmentation models have two attributes: object bounding boxes and segmentation masks. In conjunction with the motion consistency constraint method, we introduce a motion region constraint method to assess the motion status of potentially moving objects. This method encompasses three key aspects: motion consistency detection, instance-segmented object ID assignment, and classification of the motion state of instance-segmented objects.

Fig. 2 illustrates the schematic diagram of the motion region constraint method. The left and right sections represent

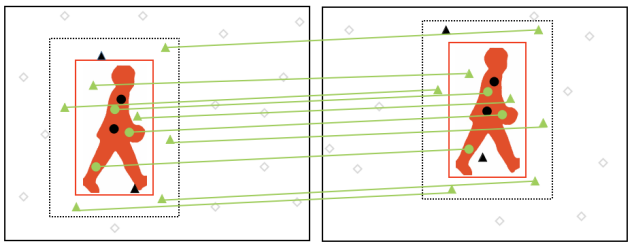

FIGURE 2. Schematic diagram of motion region constraints.

two consecutive frames of images within the dataset. In the figure, the red mask represents potential motion objects segmented by the instance segmentation network. The red solid-line box denotes the target detection box for potential motion objects, and the black dashed-line box is the expansion box that enlarges the region around the target detection box. Circular dots represent feature points extracted from potentially moving objects, triangular dots represent feature points extracted from regions outside the mask area within the expansion frame, and diamond dots represent feature points extracted from regions outside the expansion rectangular box. Successfully matched feature points are depicted in green and connected by solid green lines, while unsuccessful matches are shown in black.

1) MOTION CONSISTENCY DETECTION

In this section, we employ the motion consistency detection method to identify moving points within an image. This process is executed in the following steps: First, we compute an optical flow pyramid to obtain feature point pairs that have been successfully matched between the current frame and the previous one. We discard the matched pairs if they are located within potentially moving objects or at the image edges. Subsequently, we utilize the remaining matched point pairs to estimate the fundamental matrix. Using the fundamental matrix, we compute the epipolar line of the matched point pairs in the current frame. Finally, we assess whether the distance from the matching point to the epipolar line in the current frame is below a predefined threshold. If it is less than the threshold, the point is considered stationary; otherwise, it is regarded as having moved.

The figure3 is an epipolar geometric constraint diagram, where P represents a point in the three-dimensional space, O_1 and O_2 represent the camera centers of the two frames, l_1 and l_2 represent epipolar lines, p_1 and p_2 represent the matching points in the previous frame and the current frame. According to [6], the fundamental matrix F can be calculated by the following formula:

$$p_2^T F p_1 = 0 \tag{1}$$

The epipolar line l_2 can be calculated by the following formula:

$$l_2 = FP_2 = F \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \tag{2}$$

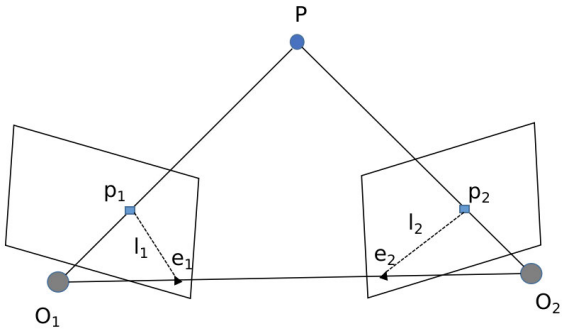

FIGURE 3. Epipolar constraints.

2) INSTANCE-SEGMENTED OBJECTS ID ASSIGNMENT

The process of assigning IDs to instance-segmented objects in this paper can be divided into the following steps.

The first step involves utilizing the instance segmentation result from the current frame in combination with the extracted feature points to obtain feature points situated within the segmented regions of each object in the current frame.

The second step involves identifying the object in the current frame with the highest count of matching pairs within the mask of its target region from the previous frame for each object. If these two objects share the same attribute and the count of matching pairs exceeds a predefined threshold, they are deemed to represent the same object.

The third step involves assigning an ID to the object. If the object was not found in the previous frame, a new ID is assigned to the current frame object. If the object was found in the previous frame, it is checked whether the object already has an ID from the previous frame. If it does have an ID, the current frame object is assigned the same ID. If it does not have an ID, both the object from the previous frame and the current frame are assigned new IDs simultaneously.

3) MOTION STATE CLASSIFICATION OF POTENTIAL MOVING OBJECTS

In this paper, objects with dynamic attributes, such as people and cars, are referred to as potentially moving objects. These potentially moving objects are classified into two states: stationary and in motion, based on their movement speed. Objects inherently lacking dynamic attributes, such as books, chairs, computers, and similar items, are denoted as static objects. The method employed for categorizing instance-segmented objects into these motion states is as follows:

Step 1: For each potential moving object in the current frame, check whether there is an object with the same ID in its previous frame.

Step 2: If there is a potential motion object with the same ID, then the motion state is divided by the following method. Firstly, calculate the number of successfully matched feature points in the region other than the segmentation area within

the expanded rectangle box of the same object in the current frame and the previous frame, denoted as m. Then calculate the number of matched points in the region other than the segmentation area within the expanded rectangle box of the object in the current frame, denoted as n. Finally, calculate the ratio of the number of matched feature points m to the number of matched points n, denoted as δ. The potentially moving objects are categorized into stationary and in-motion states based on the interval in which δ is located. The formula is as follows:

$$\delta = \frac{m}{n} \tag{3}$$

Step 3: In the absence of an object with the same ID, motion consistency detection is conducted on the current frame. If the number of detected moving points on the potential motion object surpasses a specific threshold, it is categorized as a potentially moving object in the motion state; otherwise, it is classified as a potentially moving object in the stationary state.

D. STRATEGY FOR DYNAMIC FEATURE POINT REMOVAL

When the proportion of pixels occupied by potential moving objects in the image is relatively high, simply removing the feature points on potential moving objects can potentially lead to tracking thread failures due to the reduced number of available feature points.

To address the aforementioned issues, this paper presents the following methods. Firstly, prior to feature point extraction, the proportion of pixels occupied by potential moving objects in the previous frame is calculated, and if this proportion is significant, the number of feature points extracted in the current frame is increased. Secondly, prior to dynamic feature point extraction, the proportion of pixels occupied by potential motion objects in the current frame image is calculated. If this proportion is small, all feature points on potential motion objects are removed. Conversely, if the proportion is large, only the feature points on highly dynamic potential motion objects are eliminated. These methods not only mitigate the impact of highly dynamic moving objects on the position estimation of the visual SLAM system but also substantially enhance the system's robustness.

E. DENSE MAPPING

The ORB-SLAM2 [1] system generates sparse maps with a limited number of point clouds, which fail to capture detailed information about the reconstructed environment. In contrast, DRV-SLAM employs RGB images and depth maps to produce dense point clouds with accurate depth values while retaining the sparse point cloud maps.

DRV-SLAM is well-suited for localization and mapping in dynamic scenes. Typically, our goal is to conduct dense mapping on objects with static attributes, excluding any potentially moving objects in the point cloud map. During the dense mapping process, map points that are associated with potentially moving objects are removed to ensure the accurate construction of dense maps.

In large-scale SLAM scenarios, dense mapping increases the global point cloud count significantly. This has a dual impact: it prolongs the dense mapping phase, reducing overall SLAM system speed, and imposes heavy memory demands, risking memory overflow and reducing system robustness. Typically, the focus is on the detailed representation of prominent objects in the dense point cloud, with less emphasis on background elements like walls and tabletops.

Based on the above situation and requirements, this paper proposes a global downsampling and targeted object data enhancement scheme for dense mapping. Global downsampling refers to downsampling all the point clouds in the depth point cloud map of the keyframe and then transforming the remaining point clouds to the world coordinate system to save them as the global point cloud map. The method of targeted object data enhancement is based on global downsampling and then through the image information of several common frames nearest to the keyframe and its instance segmentation results, only the point clouds on the segmented static key objects are transformed to the world coordinate system and saved as the global point cloud map, which enhances the details of the targeted objects in the dense map.

Figure 4 for the targeted object data enhancement method schematic, here assuming that the book on the table is the "key object", the instance segmentation network has been detected in the figure and identified the book with a red mask. The right image is the key frame image, the points above are the point cloud in the camera coordinate system obtained by downsampling the depth map of the current keyframe, in which the red points represent the point clouds on the book, the left multi-frame image is the image of several frames in front of the keyframe, the red points on each frame represent the point clouds of the pixels on the book in the camera coordinate system of each frame. the point cloud on the left frames and the point cloud on the right frame are added together into the global point cloud map to realize the effect of the targeted object data enhancement.

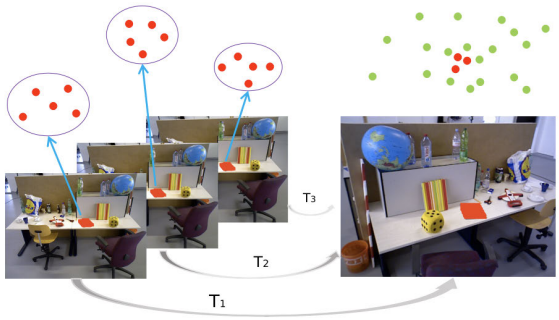

FIGURE 4. Schematic diagram of data enhancement methodology for targeted objects.

TABLE 1. Results of metrics absolute trajectory error (ATE).

Sequences	ORB-SLAM2				DRV-SLAM				Improvements			
	RMSE	S.D.	Mean	Median	RMSE	S.D.	Mean	Median	RMSE	S.D.	Mean	Median
fr3_walking_xyz	**0.7521**	**0.3759**	**0.6492**	**0.5857**	**0.0128**	**0.0064**	**0.0111**	**0.0098**	**98.29%**	**98.29%**	**98.29%**	**98.32%**
fr3_walking_static	0.3900	0.1602	0.3554	**0.3087**	0.0137	0.0070	0.0118	**0.0108**	96.48%	95.63%	96.67%	**98.32%**
fr3_walking_rpy	**0.8705**	**0.4520**	0.7425	0.7059	**0.0314**	**0.0189**	0.0250	0.0202	96.39%	95.81%	96.63%	97.13%
fr3_walking_half	0.4863	0.2290	0.4272	0.3964	0.0211	0.0102	0.0185	0.0174	95.66%	95.54%	95.66%	95.61%
fr3_sitting_static	**0.0087**	0.0043	0.0076	**0.0066**	**0.0050**	0.0027	0.0042	**0.0035**	**42.52%**	37.20%	44.73%	**46.96%**

TABLE 2. Comparison results of absolute trajectory error (ATE) between drv-slam and classical slam algorithms.

Sequences	ORB-SLAM3		DynaSLAM		DS-SLAM		YOLO-SLAM		SG-SLAM		DRV-SLAM(Ours)	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3_walking_xyz	0.2978	0.1288	0.0156	0.0079	0.0247	0.0161	0.0146	0.0070	0.0152	0.0075	0.0128	0.0064
fr3_walking_static	0.3214	0.1139	0.0064	0.0031	0.0081	0.0067	0.0073	0.0035	0.0073	0.0034	0.0137	0.0070
fr3_walking_rpy	0.1580	0.0753	0.0325	0.0194	0.4442	0.2350	0.2164	0.1001	0.0324	0.0187	0.0314	0.0189
fr3_walking_half	0.6353	0.2576	0.0261	0.0123	0.0303	0.0159	0.0283	0.0138	0.0268	0.0134	0.0211	0.0102
fr3_sitting_static	0.0071	0.0037	0.0067	0.0028	0.0065	0.0033	0.0066	0.0033	0.0060	0.0029	0.0050	0.0027

IV. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of DRV-SLAM in dynamic environments using the publicly available TUM RGB-D dataset [5] provided by the Technical University of Munich. The TUM RGB-D dataset includes multiple sequences in dynamic environments and provides accurate camera pose variations obtained through an external motion capture system. In these sequences, individuals in extreme motion scenarios occupy more than half of the image area, posing significant challenges to the operation of visual SLAM systems. This evaluation provides a rigorous test of the accuracy and robustness of visual SLAM algorithms in extremely dynamic environments.

All experimental parts of this paper were conducted on a local laptop with an Intel i5-8300H CPU, 8GB of RAM, and a GTX 1050ti graphics card with 4GB of video memory.

A. CULLING DYNAMIC OBJECTS EXPERIMENT

To illustrate the efficacy of DRV-SLAM's motion region constraint method in eliminating dynamic feature points, Experiment 1 was conducted.

Figure 5 in (a) depicts two individuals in the image, both of whom are potentially moving objects. In (b), the image feature points extracted by the conventional ORB-SLAM2 system are shown, and it can be observed that many of these feature points are distributed on the bodies of the people. (c) presents the instance segmentation mask map generated by the instance segmentation network for the potentially moving objects in the image. The mask for the potentially moving objects in the moving state (standing person) is colored in red, while the mask for the potentially moving objects in the stationary state (sitting person) is highlighted in green, achieved through the application of the motion region constraint method. (d) illustrates a scenario where potential moving objects occupy a significant portion of the image. In this case, only the feature points on the potential motion objects in the motion state are eliminated and the feature points on the potential motion objects in the stationary state are retained.

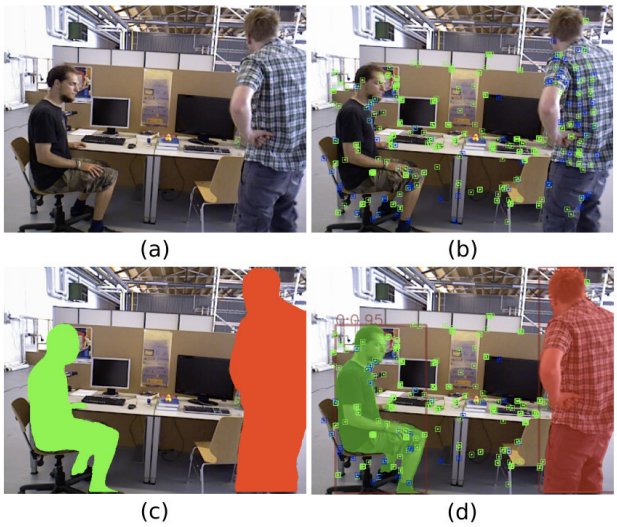

FIGURE 5. Motion object feature point rejection flowchart. Original RGB image (a), feature point distribution image extracted by ORB-SLAM (b), and mask image obtained by instance segmentation combined with motion region constraint method (c), as well as the feature point distribution image after removing dynamic feature points by DRV-SLAM (d).

B. PERFORMANCE EVALUATION OF DRV-SLAM IN A DYNAMIC ENVIRONMENT

In this section, we conduct experimental comparisons involving DRV-SLAM, ORB-SLAM2, and other state-of-the-art

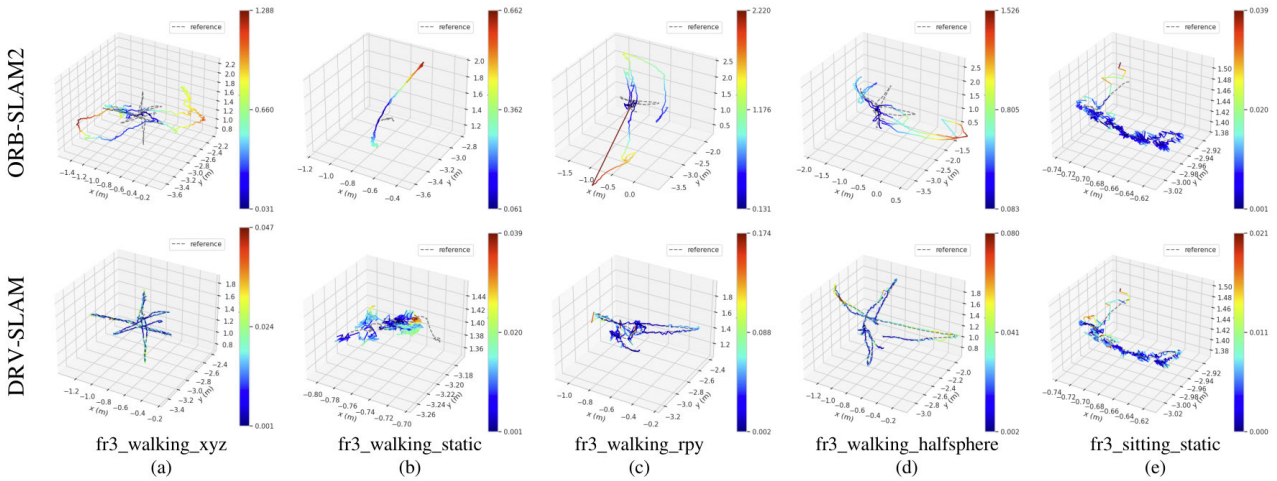

FIGURE 6. ATE graphs of DRV-SLAM and ORB-SLAM2 running five sequences. (a) fr3_walking_xyz. (b) fr3_walking_static. (c) fr3_walking_rpy. (d) fr3_walking_halfsphere. (e) fr3_sitting_static.

SLAM algorithms. The performance metric used for these comparisons is the Absolute Trajectory Error (ATE) [5].

Our entire Experiment 2 was conducted in the TUM RGB-D dataset, and the sequences of the TUM RGB-D dataset where the experiments were conducted were: fr3_walking_xyz, fr3_walking_static, fr3_walking_rpy, fr3_walking_half and fr3_sitting_static, where walking and sitting represent the scenarios of a person walking in an image sequence and a person sitting in an image sequence, respectively, and xyz, static, rpy, and half represent the four types of self-motion of the camera, e.g., xyz indicates that the camera is moving along the xyz axis.

The results of the absolute trajectory error comparison between DRV-SLAM and ORB-SLAM2 [1] are shown in Table 1, where we give the Root Mean Square (RMSE), Standard Deviation (S.D.), Mean, and Median. In addition, we show the trajectory enhancement rate of DRV-SLAM concerning the original ORB-SLAM2 localization, and the enhancement rate in the table is calculated as follows [5]:

$$\eta = \frac{o - r}{o} \times 100\% \tag{4}$$

where o denotes the absolute trajectory error of ORB-SLAM2, r denotes the absolute trajectory error of DRV-SLAM, and η represents the trajectory enhancement rate of DRV-SLAM relative to the original ORB-SLAM2 localization.

As can be seen from Tables 1, the performance exhibited by DRV-SLAM in high dynamic sequences can be improved by an order of magnitude compared to that with ORB-SLAM2, with the highest improvement values of 98.28%, 98.27%, 98.31%, 98.29% in the four metrics, namely, RMSE, Mean, Median, and S.D. In low dynamic sequences, such as the fr3_sitting_static sequence, the performance also gets a good boost, with improvement values of 42.02%, 43.96%, 46.41%, and 37.16% in the four metrics of RMSE, Mean, Median, and S.D., respectively.

TABLE 3. Time analysis.

Systems	Neural Network	Average Inference Time Per Frame (ms)	Hardware Platform
DS-SLAM	SegNet	37.57	Intel i7 CPU,P4000 GPU
Dynamic-SLAM	Mask R-CNN	195	Nvidia Tesla M40 GPU
SG-SLAM	SSDLite	76.32	Intel i5 CPU,Nvidia GTX 1050ti GPU
DRV-SLAM(Ours)	YOLOv8-seg	24.66	Intel i5 CPU,Nvidia GTX 1050ti GPU

Figures 6 and 7 display the ATE plots of ORB-SLAM2 and DRV-SLAM on the five high-dynamic sequences [5], respectively. In these figures, the gray dashed line represents the true reference value, while the colored solid line represents the estimated pose by the SLAM system. We can observe that DRV-SLAM significantly reduces trajectory errors compared to ORB-SLAM2.

To further assess DRV-SLAM's performance, this paper conducts a comparative analysis with ORB-SLAM3, DynaSLAM, DS-SLAM, YOLO-SLAM and SG-SLAM using the TUM RGB-D dataset. The results of the experimental comparison of absolute trajectory errors across five sets of sequences are presented in Table 2. The findings indicate that DRV-SLAM significantly outperforms traditional visual SLAM algorithms like ORB-SLAM3 in highly dynamic environments, achieving an order of magnitude improvement in performance. Additionally, DRV-SLAM exhibits greater efficiency compared to most state-of-the-art SLAM systems designed for dynamic scenes, such as DynaSLAM, DS-SLAM, YOLO-SLAM and SG-SLAM, while maintaining superior performance.

The deep learning network model inference experimental time-consuming results and hardware platforms are shown in Table 3. In the experiments, DRV-SLAM's instance segmentation network demonstrates exceptional performance with an average inference speed of up to 24 milliseconds per image. This speed is significantly faster than the

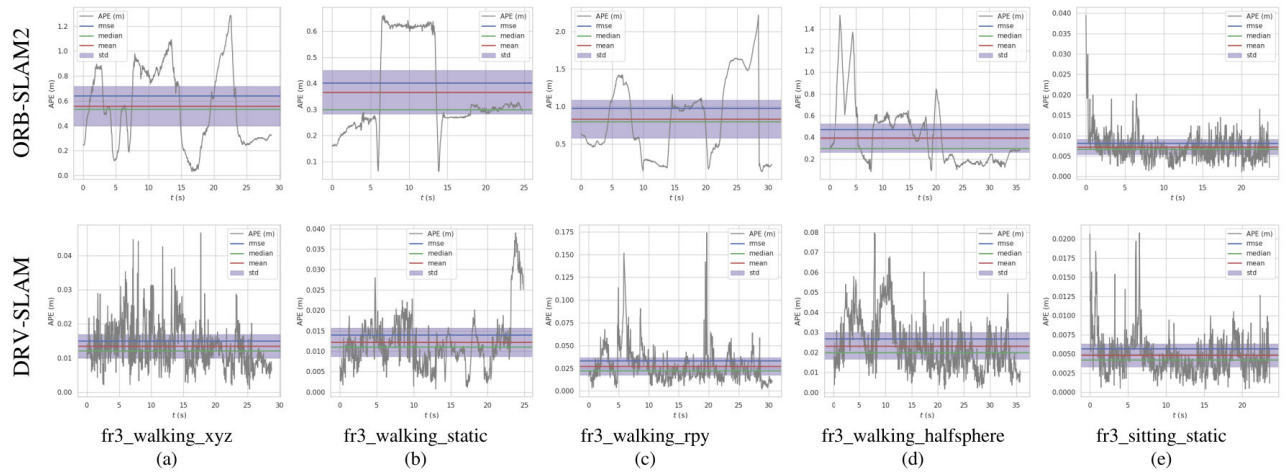

FIGURE 7. ATE results of DRV-SLAM and ORB-SLAM2 running five sequences. (a) fr3_walking_xyz. (b) fr3_walking_static. (c) fr3_walking_rpy. (d) fr3_walking_halfsphere. (e) fr3_sitting_static.

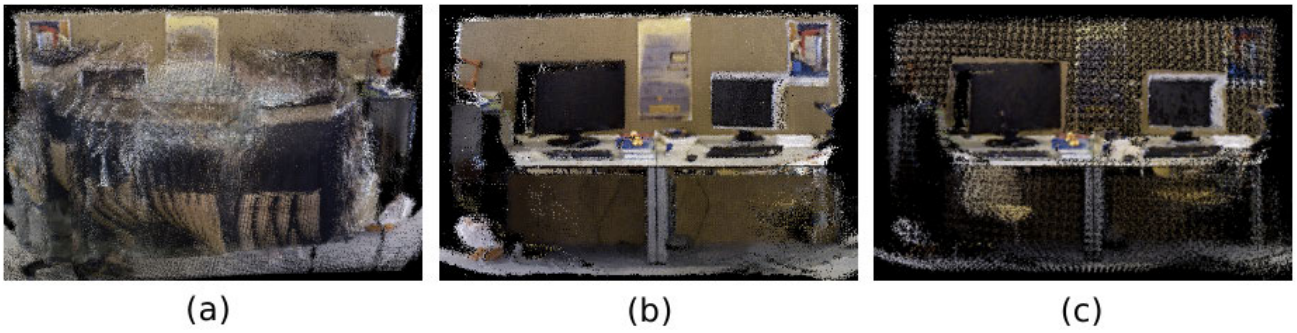

FIGURE 8. Comparison of dense mapping results. Dense map without dynamic object removal (a), dense map after removing dynamic objects (b), and Dense maps using global downsampling and target object data augmentation methods (c).

segmentation speed of other SLAM systems designed for dynamic environments.

C. DRV-SLAM DENSE MAPPING EXPERIMENT

To evaluate the dense mapping performance of the DRV-SLAM system, we conducted Experiment 3 using the TUM RGB-D dataset. Figure 8 presents a comparison of the dense mapping results. In Figure (a), the dense mapping results without point cloud rejection for potentially moving objects in the map-building module are displayed. It is observed that a large number of point clouds are mapped onto potentially moving objects in the dense point cloud map. Due to the dynamic state of these objects, they cause a 'smearing' effect when mapped into the dense point cloud map, significantly degrading the map's quality. Figure (b) illustrates the dense mapping results after the point cloud removal of potentially moving objects within the mapping module. In comparison to Figure (a), it is evident that point clouds on potentially moving objects have been largely eliminated, resulting in a noticeable enhancement in the quality of the dense mapping.

In this paper, a dense mapping scheme is proposed in the dense mapping module, featuring global downsampling and targeted object data enhancement. Figure (c) in Fig. 8 illustrates the effects of the dense mapping scheme with global downsampling and targeted object data enhancement in DRV-SLAM. From Figure (b), it is evident that the sampling frequency of the dense map point cloud is consistent throughout the entire scene, resulting in a high number of point clouds in the map. In Figure (c), it is evident that a higher density of point clouds is exclusively present on targeted objects, such as computers, keyboards, books, and more, located on the table within the dense point cloud map. This configuration offers a more detailed representation of these specific details. Conversely, other regions of the map contain a lower density of point clouds, resulting in minimal or almost no detail representation. Figure (b) contains 952,382 point clouds in the dense point cloud map, while Figure (c) contains 326,987 point clouds. The dense mapping scheme involving global downsampling and targeted object data enhancement reduces the total number of created point clouds by approximately 65.7% compared to the global dense mapping scheme, effectively reducing the system memory footprint occupied by the dense point cloud map.

V. CONCLUSION

In this paper, we propose a complete adaptive real-time semantic visual SLAM (DRV-SLAM) system for complex dynamic scenes.DRV-SLAM employs an improved instance segmentation network YOLOv8 combined with motion region constraints, which can detect potentially moving objects and determine their motion states. According to the proportion of potential moving objects in the image, the system can dynamically adjust the range of dynamic feature points that need to be rejected, thus improving the robustness and accuracy of the system in dynamic environments. In the dense mapping session, the system adopts the strategy of global downsampling and targeted object data enhancement to reduce the memory occupied by the dense point cloud map, which makes it suitable for dense point cloud mapping in spatially large environments. It is worth noting that this mapping strategy is not commonly seen in previous SLAM mapping algorithms. However, its effectiveness is evident. This mapping method provides a direction for reducing point cloud memory usage and focusing on target reconstruction in the field of SLAM. The experimental results show that DRV-SLAM significantly outperforms ORB-SLAM2 and ORB-SLAM3 in terms of accuracy and robustness in complex dynamic environments, and slightly outperforms DynaSLAM and DS-SLAM, in addition, DRV-SLAM significantly outperforms other advanced visual dynamic SLAM systems in terms of image inference speed for segmented networks.

There are still some disadvantages of the system that need to be addressed in the future. For example, when multiple potential moving objects overlap in the field of view of the image, it may lead to errors in determining the IDs of potential moving objects. Additionally, the instance segmentation model's ability to recognize object types and accuracy needs further improvement.

In the future, our research will focus on building semantic octree maps [39], and applying them to robot navigation tasks in order to perform more advanced tasks in complex dynamic environments. This direction will help to improve the intelligent navigation of mobile robots and cope with diverse dynamic scenarios.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[2] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 834–849. [Online]. Available: https://springer.dosf.top/chapter/10.1007/978-3-319-10605-2_54

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[4] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[5] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

[6] A. Kundu, K. M. Krishna, and J. Sivaswamy, "Moving object detection by multi-view geometric techniques from a single camera mounted robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 4306–4312.

[7] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, Feb. 2013.

[8] R. Wang, W. Wan, Y. Wang, and K. Di, "A new RGB-D SLAM method with moving object detection for dynamic indoor scenes," *Remote Sens.*, vol. 11, no. 10, p. 1143, May 2019.

[9] 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, Oct. 2018.

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

[11] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1168–1174.

[12] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, "Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment," *Robot. Auto. Syst.*, vol. 117, pp. 1–16, Jul. 2019.

[13] I. Ballester, A. Fontan, J. Civera, K. H. Strobl, and R. Triebel, "DOT: Dynamic object tracking for visual SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11705–11711.

[14] L. Cui and C. Ma, "SOF-SLAM: A semantic visual SLAM for dynamic environments," *IEEE Access*, vol. 7, pp. 166528–166539, 2019.

[15] K. Lv, Y. Zhang, Y. Yu, Z. Wang, and J. Min, "SIIS-SLAM: A vision SLAM based on sequential image instance segmentation," *IEEE Access*, vol. 11, pp. 17430–17440, 2023.

[16] Y. Liu and J. Miura, "RDS-SLAM: Real-time dynamic SLAM using semantic segmentation methods," *IEEE Access*, vol. 9, pp. 23772–23785, 2021.

[17] B. Bescos, C. Campos, J. D. Tardos, and J. Neira, "DynaSLAM II: Tightly-coupled multi-object tracking and SLAM," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5191–5198, Jul. 2021.

[18] J. Zhang, M. Henein, R. Mahony, and V. Ila, "VDO-SLAM: A visual dynamic object-aware SLAM system," 2020, *arXiv:2005.11052*.

[19] 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/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 258–266.

[20] R. Long, C. Rauch, V. Ivan, T. Lun Lam, and S. Vijayakumar, "RGB-D-inertial SLAM in indoor dynamic environments with long-term large occlusion," 2023, *arXiv:2303.13316*.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[22] 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, Dec. 2017.

[23] 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.

[24] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[25] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[26] 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, pp. 1–12, 2023.

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision—ECCV*, Amsterdam, The Netherlands. Springer, 2016, pp. 21–37. [Online]. Available: https://springer.dosf.top/chapter/10.1007/978-3-319-46448-0_2

[28] F. Demim, A. Boucheloukh, A. Nemra, E. Kobzili, M. Hamerlain, and A. Bazoula, "An adaptive SVSF-SLAM algorithm in dynamic environment for cooperative unmanned vehicles," *IFAC-PapersOnLine*, vol. 52, no. 15, pp. 394–399, 2019.

[29] M. Tang, Z. Chen, and F. Yin, "An improved H-infinity unscented FastSLAM with adaptive genetic resampling," *Robot. Auto. Syst.*, vol. 134, Dec. 2020, Art. no. 103661.

[30] Z. Zhang, E. Lyu, Z. Min, A. Zhang, Y. Yu, and M. Q.-H. Meng, "Robust semi-supervised point cloud registration via latent GMM-based correspondence," *Remote Sens.*, vol. 15, no. 18, p. 4493, Sep. 2023.
[31] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[33] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.
[35] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[36] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*. Munich, Germany. Springer, 2015, pp. 234–241. [Online]. Available: https://springer.dosf.top/chapter/10.1007/978-3-319-24574-4_28
[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
[38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision–ECCV*, Zurich, Switzerland. Springer, 2014, pp. 740–755. [Online]. Available: https://springer.dosf.top/chapter/10.1007/978-3-319-10602-1_48
[39] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auto. Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.

QIANG JI received the B.S. degree in vehicle engineering from Shandong Jiaotong University, Jinan, China, in 2022. He is currently pursuing the M.S. degree in electronic information with China Jiliang University, Hangzhou, China. His main research interests include simultaneous localization and mapping (SLAM), computer vision, and robotics.

ZIKANG ZHANG received the B.S. degree in automation engineering from Wanjiang University of Technology, Maanshan, China, in 2022. He is currently pursuing the M.S. degree in electronic information with China Jiliang University, Hangzhou, China. His main research interests include simultaneous localization and mapping (SLAM), computer vision, and robotics.

YIFU CHEN received the B.S. degree in electrical engineering and automation from China Jiliang University, Hangzhou, China, in 2022, where he is currently pursuing the M.S. degree in electronic information. His research interests include deep learning and object detection based on unmanned aerial vehicle (UAV).

ENHUI ZHENG received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2006. He is currently an Associate Professor with the Department of Automation, China Jiliang University, Hangzhou. He is the Deputy Secretary-General of Zhejiang Model Radio Sports Association. His research interests include UAV application, image processing, deep learning, and simultaneous localization and mapping.