Real-time Visual SLAM Based on Lightweight PSPNet Network

Yuan Luo, Jixiang Shen, Fangyu Li

Abstract—**In the context of SLAM, the implementation of the PSPNet architecture has demonstrated efficacy in the removal of dynamic objects, which subsequently enhances the localization precision of visual SLAM systems operating within dynamic environments. Nonetheless, this methodology is characterized by suboptimal performance in real-time applications. This study introduces a real-time visual SLAM system that employs a streamlined PSPNet architecture to address the identified challenges. This system enhances real-time performance by optimizing the PSPNet network and incorporating a key frame selection module. It also utilizes semantic segmentation results to refine the homography matrix and optical flow methods to improve system accuracy. Comparative experiments performed on publicly available datasets demonstrate that the proposed system not only attains a high level of localization accuracy in dynamic environments but also exhibits superior real-time performance, thereby fulfilling practical requirements.**

Index Terms—**dynamic environment, PSPNet, semantic segmentation, VSLAM**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM) refers to the process by which a mobile robot concurrently determines its own position and constructs a map of its environment, all without relying on any prior information [1][2]. SLAM has emerged as a prominent area of research within the domain of artificial intelligence applications, particularly in relation to unmanned vehicles and mobile robotics. SLAM can be classified into two primary categories based on the types of sensors employed: laser SLAM and visual SLAM (VSLAM). Among these, VSLAM is predominantly utilized due to its capacity to gather a greater amount of information regarding the external environment.

Manuscript received on April 2, 2024; revised on September 12, 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).

Yuan Luo 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: luoyuan@cqupt.edu.cn).

Jixiang Shen is a graduate student of the School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China (corresponding author phone: 198-2381-3316; e-mail: S220432006@stu.cqupt.edu.cn).

Fangyu Li is a graduate student of the School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China (e-mail: S220431046@stu.cqupt.edu.cn).

Currently, researchers have proposed numerous outstanding VSLAM algorithms tailored for static environments (SE). Among the various options available, the ORB-SLAM series has become one of the most extensively utilized solutions, attributed to its real-time performance on central processing units (CPUs) and its reliable functionality [3]. ORB-SLAM [4], introduced by MUR-Artal et al. in 2015, represents a significant advancement in the field. This algorithm is considered a notable successor to PTAM [5], distinguished by its innovative use of three concurrent threads: tracking, local map building, and loopback detection. This multi-threaded approach effectively minimizes cumulative error and enhances both processing speed and map-building accuracy, yielding superior results. Building upon ORB-SLAM, the team has subsequently introduced ORB-SLAM2 [6] and ORB-SLAM3 [7]. ORB-SLAM2 constitutes a resilient SLAM framework that is suitable for use with monocular (MON), stereo (STE), and RGB-D camera systems. This system facilitates the reuse of maps, detection of loop closures, and the process of re-localization. ORB-SLAM2 functions in real-time on conventional CPUs, rendering it applicable to a diverse array of contexts, including compact handheld devices utilized in indoor settings, industrial environments, as well as unmanned aerial vehicles and autonomous transportation systems navigating urban landscapes. ORB-SLAM3 represents the inaugural SLAM system that integrates visual, visual-inertial, and set-value mapping functionalities. The methodology utilizes both pinhole and fisheye lens models across a range of camera configurations, which encompass MON, STE, and RGB-D systems. In comparison to ORB-SLAM2, its primary advantage resides in its capacity to proficiently leverage short-term, medium-term, long-term, and multi-map data, thereby attaining levels of accuracy that were not attainable with ORB-SLAM2. Owing to its elevated precision and immediate operational capabilities in unchanging settings, ORB-SLAM3 has become a foundational framework for numerous scholars in recent years. Researchers have utilized ORB-SLAM3 as a basis for enhancements, aiming to develop VSLAM systems that perform robustly in dynamic environments (DE). In this study, ORB-SLAM3 serves as the foundational framework for enhancement.

ORB-SLAM3 is fundamentally structured around three principal threads: the tracking thread, the local mapping thread, and the thread responsible for loop closure and map merging. The tracking thread is tasked with the identification and correspondence of local map feature points. It employs the Bundle Adjustment (BA) algorithm to reduce the reprojection error, which facilitates the precise estimation of the camera's position for each individual frame. The local map construction process can enhance the optimization of the

camera position and the feature point cloud by employing the local BA algorithm, which utilizes the outcomes derived from the tracking thread. Ultimately, the loopback and map merge thread is capable of identifying loopback occurrences and mitigating cumulative drift errors through bitmap optimization. Following this optimization process, the global BA algorithm thread is initiated to calculate the optimal configuration of the entire system and the corresponding motion results. ORB-SLAM3 demonstrates commendable performance in SE; however, a significant disparity persists between the estimated trajectory and the actual trajectory in DE.

In order to reduce the influence of dynamic objects on position estimation within ORB-SLAM3 and to improve its accuracy in environments characterized by dynamic elements, this study proposes the incorporation of a semantic thread that operates concurrently with the tracking thread of ORB-SLAM3. By employing the Pyramid Scene Parsing Network (PSPNet) [8] semantic segmentation network (SSNet), input image frames are segmented to identify and exclude dynamic objects. This approach aims to further optimize the camera's position estimation. PSPNet is a pyramidal scene analysis network designed to integrate contextual information from various regions, thereby enhancing the model's capacity to comprehend global contextual data. However, it exhibits suboptimal performance in real-time applications. To address the issue of the prolonged duration required for semantic segmentation using PSPNet, this study proposes a modification to the PSPNet architecture by substituting its backbone network, ResNet [9], with the more lightweight deep neural network MobileNetV2 [10]. This alteration aims to enhance the efficiency and speed of the semantic segmentation process. To further enhance real-time performance to meet application demands, this paper introduces a key frame selection module preceding the tracking and semantic threads. This module classifies input image frames into two distinct categories: key frames and non-key frames. Key frames undergo semantic segmentation, while non-key frames utilize segmentation results from key frames and dynamic keypoints from optical flow tracking to directly generate semantic images. This approach aims to optimize the efficiency of semantic segmentation. Given the potential impact of PSPNet's lightweight design and keyframe selection on subsequent pose estimation accuracy, this study further refines the solution of the single response matrix and optical flow methods. This study incorporates findings from semantic segmentation into these methodologies to strengthen the reliability of the derived single response matrix and to enhance the precision of optical flow detection. The objective of this approach is to improve the overall accuracy of the system. The primary contributions of this paper are outlined as follows:

1) The PSPNet SSNet has been validated and enhanced. Modifications were made to the PSPNet architecture to reduce its complexity, thereby increasing the speed of semantic segmentation processes.
2) A novel keyframe selection strategy has been introduced. The proposed method filters the input image frames according to the established keyframe selection criteria, thereby significantly enhancing the system's real-time performance.
3) The outcomes of semantic segmentation are utilized in addressing the single response matrix, thereby enhancing the robustness of the required matrix and increasing the localization accuracy of the system.
4) The results of semantic segmentation are incorporated with the dynamic detection algorithm utilizing the optical flow method, which subsequently diminishes the computational burden associated with dense optical flow. This methodology more effectively retains static feature points (SFP), resulting in improved accuracy and real-time performance of the system.

The following sections of this paper are organized in the manner outlined below. Section II provides an extensive examination of the current literature related to VSLAM within DE. Section III provides a detailed discussion of the architecture of the proposed system. In Section IV, a comparative analysis is conducted between the proposed system and other leading VSLAM systems utilizing the TUM RGB-D dataset, thereby assessing its accuracy and real-time performance. In conclusion, Section V provides a comprehensive summary and analysis of the research findings.

II. Related work

Traditional VSLAM systems predominantly operate under the assumption of SE. However, the presence of dynamic objects is a common occurrence in indoor settings. This reliance on static scene assumptions significantly constrains the advancement of VSLAM technology and limits the practical application of VSLAM systems in real-world scenarios. The integration of deep learning networks into VSLAM has demonstrated a significant capacity to filter out dynamic objects, thereby enhancing the resilience of VSLAM systems in environments characterized by dynamic elements. Presently, VSLAM methodologies that leverage deep learning in dynamic contexts can be primarily classified into three distinct categories: VSLAM that utilizes target detection, VSLAM that employs semantic segmentation, and VSLAM that is based on instance segmentation. The target detection network is capable of extracting both target and spatial information from images, enabling the identification of object categories through the delineation of candidate bounding boxes [11]. This approach is characterized by its high efficiency and rapid processing capabilities; however, it is often constrained by limitations in accuracy. Prominent examples of target detection networks include R-CNN [12], Faster R-CNN [13], YOLO [14], and SSD [15]. Furthermore, VSLAM systems that integrate these networks generally demonstrate enhanced performance in real-time applications. Semantic segmentation demonstrates superior accuracy in comparison to object detection, as it analyzes images at the pixel level, thereby extracting comprehensive information about the visual content. However, this process is often time-intensive. Common SSNet include Deeplabv2 [16], SegNet [17], PSPNet, ICNet [18], etc. The integration of SSNet into VSLAM systems can achieve high accuracy in DE; however, it often falls short in real-time performance, thereby failing to meet the requirements of practical applications. Instance segmentation, which provides both

pixel-level classification and spatial information regarding distinct objects, is capable of identifying multiple instances of the same object. Consequently, instance segmentation demonstrates superior segmentation accuracy compared to semantic segmentation. Common instance segmentation networks are Mask R-CNN [19], Yolact [20], PolarMask [21], and SOLO [22]. The instance segmentation network has the highest segmentation accuracy among the three networks, but as with semantic segmentation, real-time performance cannot be guaranteed. Each of the three deep learning networks exhibits distinct strengths and limitations. The present emphasis of research is on the efficient utilization of these networks to filter dynamic objects, which in turn improves localization precision and real-time performance in dynamic settings within VSLAM systems.

Currently for the problem that traditional VSLAM cannot adapt to DE, researchers have proposed many VSLAM systems that perform well in DE in combination with deep learning networks. In 2018, Berta Bescos and colleagues developed DynaSLAM [23], a VSLAM system that builds upon the ORB-SLAM2 framework. DynaSLAM integrates modules for dynamic object detection and background reconstruction, thereby improving the system's resilience in MON, STE, and RGB-D configurations when operating in DE. In the same year, Chao Yu and colleagues introduced a resilient semantic VSLAM system, referred to as DS-SLAM, designed for DE [24]. This system integrates a SSNet with mobile coherence detection to mitigate the impact of dynamic objects, thereby enhancing the localization accuracy of the system in such environments. Additionally, in 2020, Long X and associates presented the PSPNet network and developed a semantic SLAM framework, termed PSPNet-SLAM, which utilizes a pyramid scene parsing network for the detection of dynamic objects [25]. This study incorporates semantic threads organized in a pyramid structure alongside geometric threads utilizing an inverse ant colony search strategy within the ORB-SLAM2 framework. This integration enhances the overall system's localization accuracy and robustness; however, there remains a need for improvements in real-time performance. In 2021, Yu et al. introduced DRSO-SLAM [26], a dynamic RGB-D SLAM system that leverages semantic information and optical flow. The system underwent validation utilizing the TUM dataset, demonstrating an average improvement in root-mean-square error of 95.02% when compared to ORB-SLAM2 in highly DE. In 2023, Wu et al. introduced a dynamic scene VSLAM system known as AHY-SLAM, which employs adaptive threshold homogenization for feature extraction and incorporates YOLOv5 for target detection [27]. In comparison to ORB-SLAM2, AHY-SLAM demonstrates a substantial enhancement in the accuracy of position estimation across various dynamic scene sequences within the TUM open dataset, achieving an improvement of up to 97% in absolute pose estimation accuracy. However, this advancement is accompanied by increased computational time relative to ORB-SLAM2, attributable to the additional target detection process involved.

The ongoing advancements in computational capabilities and deep learning methodologies have led to the emergence of numerous VSLAM systems that leverage deep learning techniques. These systems significantly enhance the localization accuracy and robustness of VSLAM in DE; however, they still exhibit certain limitations that require further refinement. Consequently, the challenge of achieving an optimal balance between accuracy and real-time performance in VSLAM systems operating within dynamic contexts has emerged as a prominent area of research interest.

III. System description

Figure 1 presents the architecture of the proposed system, which enhances ORB-SLAM3 through the integration of parallel semantic threads, a keyframe selection module, a homography optimization module, and an optical flow optimization module. The initial processing of incoming image frames is conducted by the keyframe selection module. Subsequently, the selected keyframes are directed to the semantic and tracking threads for the purposes of semantic segmentation and feature point extraction, respectively. Non-keyframes undergo feature point extraction and await keyframe updates, generating semantic images via optical flow tracking of dynamic keypoints. Following the execution of semantic segmentation on keyframes, SFP located outside the semantic bounding boxes are employed to compute the homography matrix, thereby augmenting its robustness. Additionally, dynamic feature points (DFP) situated within the semantic bounding boxes undergo further filtration via the optical flow technique to preserve a greater number of SFP. Ultimately, DFP are eliminated based on the detection outcomes, and only the residual SFP are utilized for pose estimation, which enhances the overall localization accuracy of the system.

A. PSPNet

Current scene parsing frameworks primarily utilize Fully Convolutional Networks (FCN) [28] as their foundational architecture. However, FCN encounters several issues when performing image segmentation. Firstly, FCN lacks the capability to infer based on contextual information. Secondly, it cannot correlate labels through inter-class relationships. Additionally, FCN models tend to overlook small objects, and large objects may exceed FCN's receptive field. Consequently, FCN fails to effectively capture global information and handle inter-scene relationships [29]. To address these issues, researchers have proposed the PSPNet. The PSPNet integrates intricate scene information characteristics within the FCN prediction framework, thereby enabling the synthesis of both local and global features. Furthermore, it implements an optimization approach that incorporates a moderate supervision loss, which facilitates the network's ability to assimilate global scene information and effectively manage inter-scene relationships.

The PSPNet architecture is comprised of four distinct modules, as illustrated in Figure 2. Initially, an input image undergoes feature extraction utilizing a pre-trained ResNet model in conjunction with a null network strategy [30][31], resulting in a feature map that is scaled to one-eighth the dimensions of the original input image. The resultant feature maps are then subjected to a pyramid pooling module, which is specifically engineered to aggregate contextual information. This module employs pooling kernels that cover

fractional, half, and entire image regions, integrating them into a unified global prior. Finally, this prior is fused with the original feature maps to enrich their contextual understanding. Subsequently the final prediction map is generated using convolutional layer. The PSPNet architecture demonstrates a high level of segmentation accuracy and exhibits effective image processing capabilities, even in intricate scenes. However, its processing time may be excessive, potentially hindering its suitability for real-time applications in VSLAM. In order to tackle this issue, the present study enhances the PSPNet architecture by substituting its backbone network, ResNet, with MobileNetV2, a more lightweight network, thereby aiming to increase the efficiency of semantic segmentation.

B. Lightweight Network MobileNetV2

In 2018, the Google team developed MobileNetV2, which is an advancement of MobileNetV1 [32]. MobileNetV2 maintains the straightforward architecture of MobileNetV1, eliminating the necessity for specialized operators, while markedly improving accuracy and successfully addressing a range of image classification and detection tasks suitable for mobile applications. The primary innovations of MobileNetV2 lie in the introduction of Inverted Residuals and Linear Bottlenecks, which enhance the network's representational capacity.

Compared to traditional residual structures, the transformations in inverted residuals occur in two main aspects: 1) Alteration of Dimensionality. In conventional residual architectures, a 1×1 convolution is employed initially to reduce dimensionality, which is subsequently followed by a 3×3 convolution aimed at feature extraction, and ultimately, another 1×1 convolution is applied to increase dimensionality. Conversely, in inverted residual structures, the process begins with a 1×1 convolution for dimensionality enhancement, followed by a 3×3 depthwise convolution for feature extraction, and concludes with a 1×1 convolution for dimensionality reduction. This inverts the order of dimensionality reduction and increase, and replaces the standard 3×3 convolution with a depthwise convolution. 2) Variation in Activation Functions. Conventional residual architectures typically employ the ReLU activation function uniformly. In contrast, inverted residual structures utilize the ReLU6 activation function for the initial two convolutional layers, while the final convolutional layer employs a linear activation function.

The linear bottleneck architecture pertains to a specific configuration in which the concluding convolutional layer employs a linear activation function. This alteration, which involves substituting the ReLU activation function with a linear function in the final convolutional layer, represents a significant advancement introduced in MobileNetV2. The use of the ReLU function in MobileNetV1 often led to information loss by zeroing out parts of the convolutional kernels in the depthwise convolutions. In response to this issue, researchers implemented a linear activation function within MobileNetV2 to reduce the loss of information.

The architectural configuration of MobileNetV2 is presented in Table I. In this context, 't' denotes the expansion factor, indicating the multiplicative increase of the convolutional kernel in the initial 1×1 convolutional layer. The variable 'c' signifies the number of channels, while 'n' refers to the frequency of bottleneck repetitions. Additionally, 's' represents the stride, with varying stride values corresponding to distinct modules, as illustrated in Fig. 3. The MobileNetV2 network architecture comprises 17 bottleneck layers, one standard convolutional layer, and two pointwise convolutional layers. Each bottleneck layer is composed of two pointwise convolutional layers and one depthwise convolutional layer, resulting in a cumulative total of 54 trainable parameter layers within the model. The inverse residuals and linear bottleneck structure optimize the network, making the layers deeper, but the model is smaller and faster. MobileNetV2 has a reduced amount of parameters and better results compared to MobileNetV1 in ImageNet image classification tasks. Figure 4 illustrates the impact of substituting the backbone network of PSPNet with MobileNetV2 for the purpose of semantic segmentation. As depicted in the figure, the PSPNet architecture, following lightweight optimization, demonstrates a high level of accuracy in segmenting individuals and objects within the image.

C. Keyframe Selection Module

To enhance system efficiency, this study introduces a keyframe selection module preceding the tracking and semantic threads. Prior to feature point extraction and semantic segmentation, each image frame is evaluated based on a predefined keyframe selection strategy. If classified as a keyframe, it undergoes feature point extraction and semantic segmentation within the tracking and semantic threads, respectively. In contrast, non-key frames undergo feature point extraction and utilize optical flow tracking of dynamic keypoints based on segmentation results from preceding keyframes to directly generate semantic images. Given that the selection of keyframes influences both the processing duration of subsequent semantic threads and the overall localization accuracy of the system, this study establishes specific selection criteria to ensure the appropriateness of keyframe selection. The selection principles are as follows:

1) The image frame refers to either the initial frame or the first frame subsequent to a repositioning event.
2) The quantity of DFP present in the preceding frame falls below a specified threshold.
3) The image frame is the last frame.
4) Images that have passed 5 frames in a row.

Initially, if the image frame is identified as either the inaugural frame or the first frame subsequent to relocalization, it ought to be designated as a key frame. In the absence of timely processing, semantic segmentation is postponed until five frames have elapsed, leading to a degradation of information. Furthermore, in instances where the preceding frame exhibits a lack of DFP or contains a minimal number of such points, it is advisable to designate the current frame as a key frame. This is because there are scenarios where the system might not detect dynamic objects or potential dynamic objects, necessitating semantic segmentation to rule out such cases. Additionally, if the current frame is the last frame, it should also be selected as a key frame, as it may contain critical information for camera

pose estimation. Ultimately, the selection of a key frame at intervals of every five frames achieves a compromise between precision and real-time operational efficiency. By following these principles for key frame selection, the system's real-time performance can be maximized while ensuring accuracy.

In instances where an image frame is not classified as a keyframe, the DFP recorded in the most recent keyframe, along with those in the current frame, are monitored utilizing the optical flow technique. This process, facilitated by the keyframe update module, determines the spatial position of DFP within the current frame. The 3×3 grid of DFP monitored in the present frame is annotated to facilitate the direct creation of a new segmentation frame that encompasses the semantic information of the current frame. Unlike the semantic segmentation frames of keyframes, the semantic segmentation frames of non-keyframes are efficiently generated without PSPNet segmentation, and the generated segmentation frames only work in the current frame. Therefore, the semantic images generated by optical flow tracking can quickly detect and eliminate DFP in non-critical frames [33].

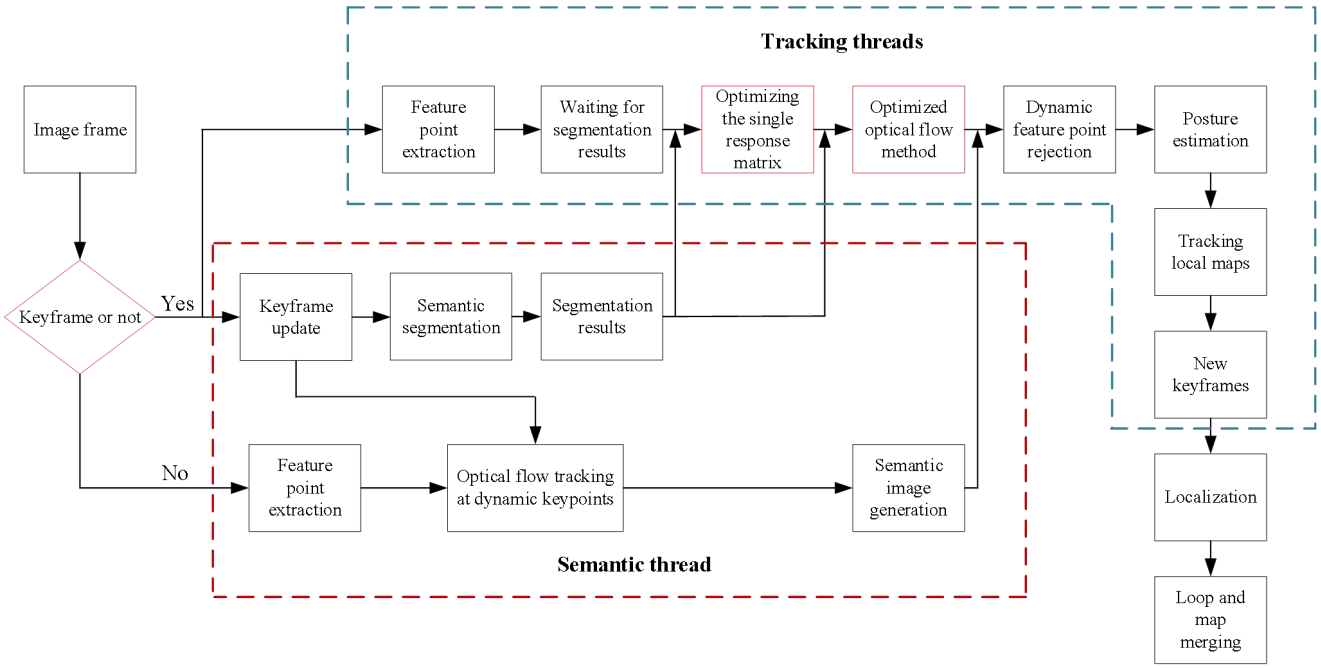


Fig. 1. System diagram of this article.

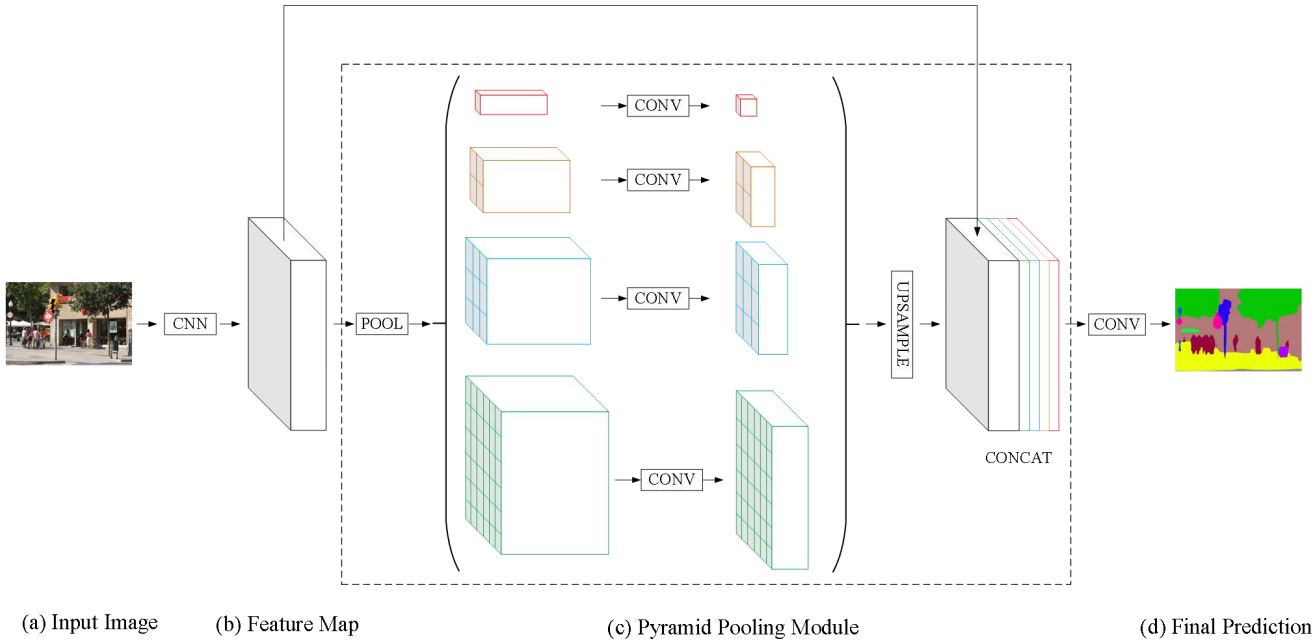


Fig. 2. PSPNet network architecture.

TABLE I
NETWORK ARCHITECTURE OF MOBILENETV2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1×1	-	1280	1	1
$7^2 \times 1280$	avgpool 7×7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1×1	-	k	-	

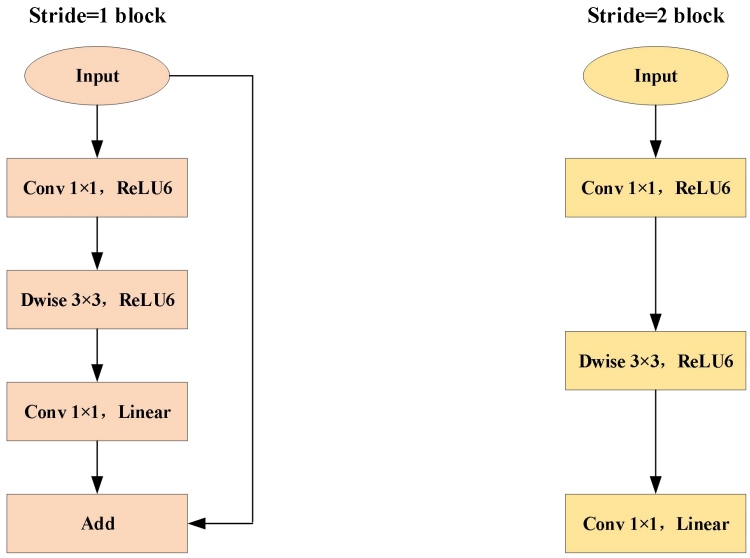


Fig. 3. Different modules for different strides.

（a） （b）

Fig. 4. (a) Image before semantic segmentation. (b) Image after semantic segmentation.

D. Optimization of the Single Response Matrix Module

To accurately estimate the camera position and calculate the corresponding optical flow vectors, it is essential to understand the mapping relationship among the matched feature points. The single response matrix delineates the correspondence between two distinct planes, for example: Consider a rectangular box situated in space, with the central point of its front face designated as point *O*. When two photographs are captured from distinct angles, point *O* is identifiable in both images. The pixel coordinates of point *O* in the first image are represented as $X_1(u_1, v_1)$, while in the second image, they are denoted as $X_2(u_2, v_2)$, Then the relationship between the two coordinates is a uni-responsive transformation, i.e.:

$$X_1 = HX_2 \tag{1}$$

where *H* is called the singular response matrix, which is converted to chi-square coordinates:

$$\begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = H \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} \tag{2}$$

In summary, the calculation of the homography matrix is crucial for accurately estimating camera pose and for the subsequent derivation of optical flow vectors. The existence of multiple DFP and noise within the image can undermine the stability of the computed homography matrix. In order to mitigate the impact of dynamic features and noise on the computation of the homography matrix, this study integrates the results of semantic segmentation into the tracking methodology. Feature points situated within the semantic bounding box are primarily characterized by their dynamic properties; therefore, these points are categorized as hypothesized DFP. Conversely, points located beyond the semantic bounding box are classified as proposed SFP, as illustrated in Figure 5.

In addition to SFP, areas outside the semantic box may also contain noise points generated during camera pose transformations. In order to mitigate the impact of noise points, this research utilizes the Random Sample Consensus (RANSAC) algorithm to enhance the removal of outliers. RANSAC is especially proficient in handling datasets characterized by a substantial presence of outliers, as it iteratively approximates the mathematical model of the data while systematically excluding the outliers.

Assuming that the homography matrix *H* serves as the mathematical model to be optimized through the RANSAC algorithm, the process involves the iterative refinement of this model by systematically incorporating random pairs of points that are selected from outside the defined semantic boundaries. RANSAC employs an iterative processing approach to identify and exclude feature point pairs that do not align with the established mathematical model, categorizing them as noise points. This approach guarantees that the feature points selected for pose estimation exhibit reduced vulnerability to noise interference, consequently enhancing the overall robustness and precision of the VSLAM system.

Following the removal of DFP pairs and noise point pairs, it is essential to utilize the remaining feature point pairs to fit the single response matrix in order to formulate the error equation:

$$\varepsilon = \sum_i ((x_t - \frac{H_{11}x_{t-1} + H_{12}y_{t-1} + H_{13}}{H_{31}x_{t-1} + H_{32}y_{t-1} + H_{33}}) + (y_t - \frac{H_{21}x_{t-1} + H_{22}y_{t-1} + H_{23}}{H_{31}x_{t-1} + H_{32}y_{t-1} + H_{33}})) \tag{3}$$

In this context, ε signifies the reprojection error associated with a pixel, while x_t and y_t denote the pixel coordinates in frame *t*, and x_{t-1} and y_{t-1} correspond to the pixel coordinates in frame *t-1*. At this juncture, the resolution of the single response matrix is reformulated as a least squares problem aimed at minimizing the error. Subsequently, the single response matrix undergoes optimization through the Levenberg-Marquardt (L-M) algorithm, with its incremental equation articulated as follows:

$$\left(J_k^T J_k + I\right)\Delta x = -J_k f\left(x\right) \tag{4}$$

$$H_e \approx J_k^T J_k + I \tag{5}$$

In this context, $f(x)$ denotes the cost function, while Δx signifies the incremental change during the iterative solution process. The L-M algorithm employs the first derivative of $f(x)$ with respect to J_k to approximate the second derivative of $f(x)$ concerning H_e Subsequently, a confidence matrix, denoted as *I*, is incorporated to guarantee the reversibility of the computed matrix. Through this approach, the optimized single response matrix, referred to as *H*, can ultimately be derived. The detailed procedural flow of the algorithm is illustrated in Figure 6(a).

E. Optimized Optical Flow Method Module

The presence of SFP on dynamic objects poses a challenge, as the complete removal of all feature points within the semantic frame via semantic segmentation may inadvertently lead to the exclusion of numerous SFP. This unintended consequence results in the loss of valuable static feature information, which can subsequently compromise the accuracy of position estimation [34]. In order to improve the precision of positioning and the real-time functionality of the system, this research utilizes an optimized single response matrix to reduce the impact of camera motion occurring between two consecutive frames. Subsequently, the methodology establishes the rejection threshold for feature points by analyzing the optical flow values of pixel points confined to the semantic frame. This approach not only diminishes the computational load associated with dense optical flow, thereby enhancing the system's real-time performance, but also preserves a greater number of SFP, which contributes to an increase in the accuracy of pose estimation. The detailed algorithmic process is illustrated in Figure 6(b).

Let us denote the optical flow value of a given pixel point as *P*. This value can be mathematically represented as:

$$P = u^2 + v^2 \tag{6}$$

In this context, *u* and *v* represent the velocity vectors corresponding to the optical flow along the x-axis and y-axis, respectively. Prior to determining the optical flow value for a given pixel, a mask matrix is generated, which is of identical

dimensions to the image matrix of the current frame. All entries within this mask matrix are initialized to a value of 1. Due to the significant disparity in optical flow values between pixel points associated with moving objects and those linked to stationary objects, a threshold (θ) is established. If the optical flow value (P) exceeds this threshold ($P > \theta$), the pixel point is classified as dynamic, and the corresponding value in the mask matrix is assigned a value of zero. Upon analyzing all the pixels within the image, a comprehensive optical flow mask representing the dynamic objects can be generated. If the feature points identified in the current frame are located within this optical flow mask, they will be discarded, while the SFP that remain will be preserved for future position estimation.

IV. Experimentation and Analysis

To assess and quantify the enhancements realized by the system presented in this paper within DE, comparative experiments were performed between the proposed system and ORB-SLAM3, along with other notable VSLAM systems in dynamic scenarios, utilizing the TUM RGB-D dataset [35]. All experiments were executed on a computer equipped with an Intel i7 CPU, an RTX3060 GPU, and 16 GB of RAM.

A. TUM RGB-D Dataset

The TUM RGB-D dataset serves as a valuable resource for assessing the precision of camera localization. It comprises several sequences captured in DE, each recorded by an RGB-D camera operating at a frame rate of 30 frames per second and a resolution of 640×480 pixels. This study involves the selection of five sequences to assess the system's performance within a dynamic environment. Among these, one sequence is characterized as low-dynamic, while the remaining four are classified as high-dynamic. The sequences utilized are fr3_sitting_static (f_s_s), fr3_walking_xyz (f_w_x), fr3_walking_static (f_w_s), fr3_walking_rpy (f_w_r), and fr3_walking_half (f_w_h). The designations xyz, static, rpy, and half correspond to four distinct categories of camera self-motion; for instance, "xyz" denotes movement of the camera along the x, y, and z axes.

To quantitatively assess the performance of the system presented in this paper, the Root Mean Square Error (RMSE) of the Absolute Trajectory Error (ATE) for each sequence is employed to facilitate comparisons among various VSLAM systems. The ATE serves as an indicator of the global consistency of the trajectory, while its RMSE provides a measure of the system's accuracy. The computation of the RMSE can be expressed as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (X_{g,i} - X_{c,i})^2}{n}} \tag{7}$$

where n represents the total number of observations; i is the *ith* observation; $X_{g,i}$ is the ground truth of the *ith* observation; $X_{c,i}$ is the calculation result of the *ith* observation.

B. Experimental Results

To demonstrate the advancements made by the algorithm presented in this paper, three VSLAM systems are selected to compare with this paper's system for the experiments, namely: ORB-SLAM3, DS-SLAM, and PSPNet-SLAM. ORB-SLAM3 is recognized as one of the most effective SLAM systems for SE, and the system presented in this paper builds upon its framework. Additionally, DS-SLAM incorporates the SSNet SegNet, positioning it as one of the leading SLAM systems for highly DE. PSPNet-SLAM adopts the same SSNet PSPNet as the system in this paper, and achieves high accuracy in DE. The four systems were subjected to experimentation across each of the aforementioned five sequences, and the resulting experimental data are presented in Table II.

As demonstrated in Table II, the system presented in this study markedly enhances the localization accuracy of VSLAM in highly DE when compared to ORB-SLAM3, achieving improvements between 91.7% and 95% across four distinct highly dynamic sequences. In comparison to DS-SLAM, the system presented in this study demonstrates superior localization accuracy in high dynamic sequences. Additionally, when evaluated against PSPNet-SLAM, which similarly employs the PSPNet architecture for semantic segmentation, the proposed system achieves localization accuracy that is on par with that of PSPNet-SLAM in high DE. Nevertheless, in sequences characterized by low dynamicity, the accuracy of the proposed system is marginally inferior to that of ORB-SLAM3. This discrepancy is due to the fact that the system is designed for high DE and may mistakenly remove SFP as DFP in low dynamic conditions, leading to reduced localization accuracy. Furthermore, to improve the system's real-time capabilities, the proposed approach optimizes PSPNet and integrates a key frame selection module. While these improvements reduce the system’s time consumption, they also have a certain impact on accuracy. Figure 7 illustrates a comparison of the estimated trajectories between the proposed system and ORB-SLAM3, indicating that the proposed system markedly improves localization accuracy in environments characterized by high dynamics.

VSLAM systems are developed for practical applications, wherein real-time performance is of paramount importance. The system introduced in this study strikes a balance between accuracy and real-time efficiency, resulting in a notable enhancement of the system's real-time capabilities while preserving a high level of accuracy. The execution time of the proposed system is evaluated in comparison to ORB-SLAM3, DS-SLAM, and PSPNet-SLAM, as presented in Table III. The data clearly indicate that the proposed system demonstrates a significant decrease in execution time relative to both DS-SLAM and PSPNet-SLAM. Specifically, in comparison to PSPNet-SLAM, which employs an identical SSNet, the proposed system demonstrates a reduction in execution time exceeding 100 milliseconds across all five sequences, achieving times that are consistently below 90 milliseconds. This performance aligns with the criteria for real-time processing. Nevertheless, the execution duration of the proposed system remains greater in comparison to that of ORB-SLAM3. This occurs due to the implementation of semantic threading within the system, which serves to

remove dynamic objects, thereby resulting in an increase in the overall execution time.

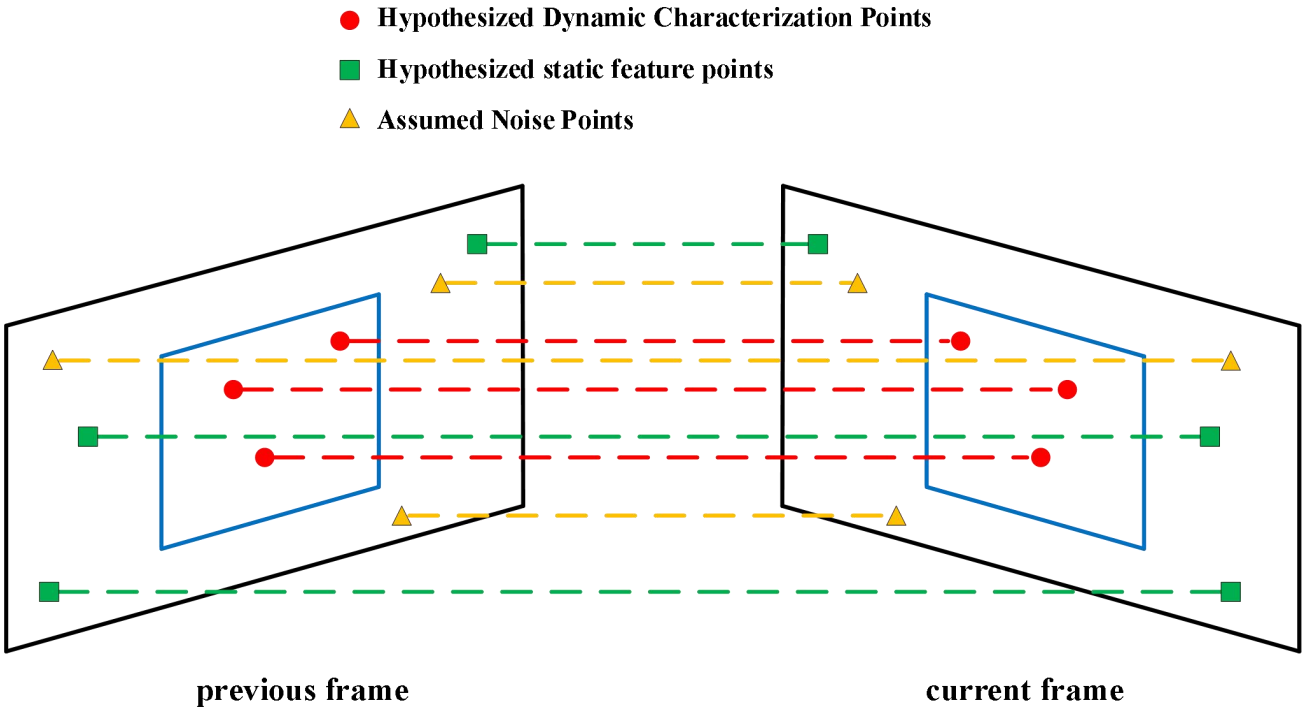
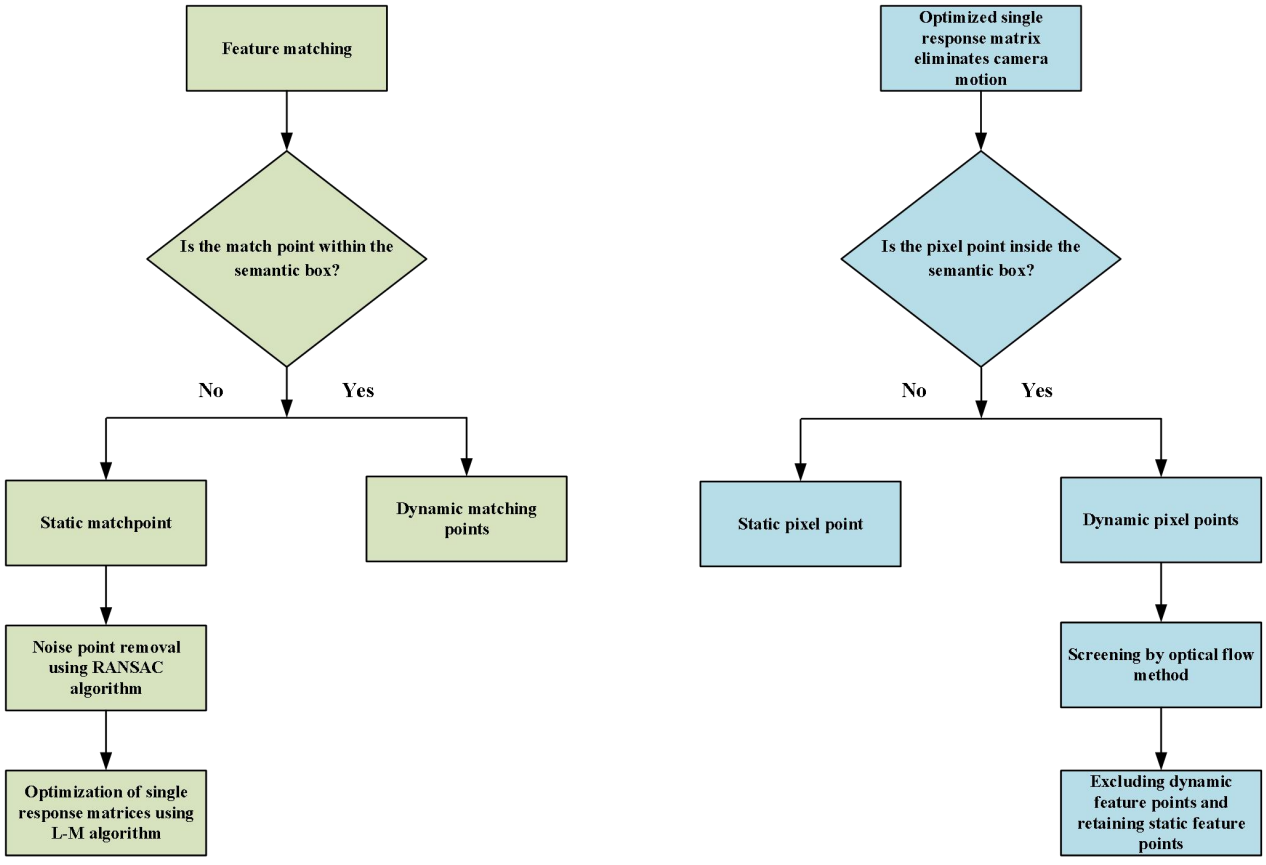


Fig. 5. The feature points from the preceding frame are mapped onto the current frame utilizing a singular response matrix.



(a) Optimizing the single response matrix

(b) Optimized optical flow method

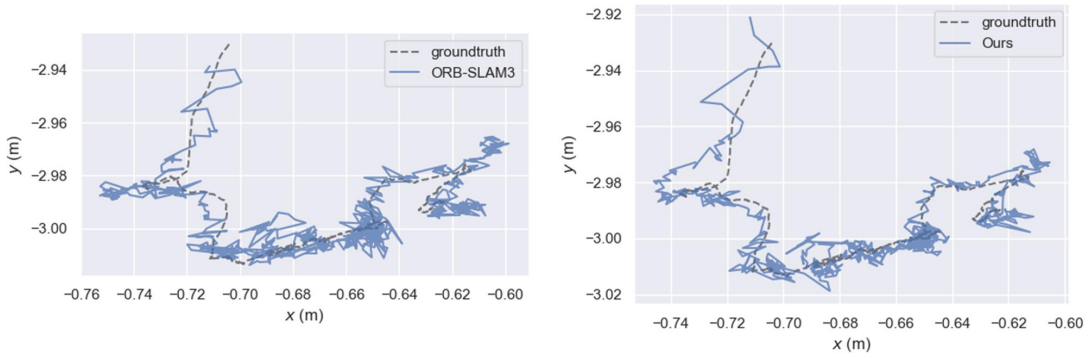
Fig. 6. Flowchart of the optimization algorithm.

TABLE II
RMSE OF ABSOLUTE TRAJECTORY DEVIATIONS ACROSS VARIOUS SYSTEMS

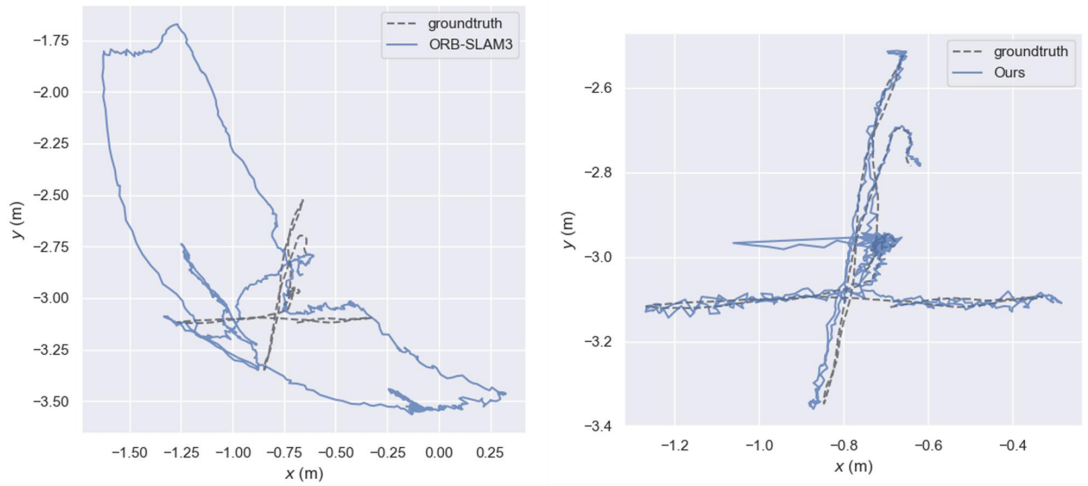
Sequences	ORB-SLAM3	DS-SLAM	PSPNet-SLAM	Ours
f_s_s	0.0093	0.0072	0.0310	0.0107
f_w_x	0.6238	0.0348	0.0263	0.0311
f_w_s	0.4360	0.0117	0.0091	0.0103
f_w_r	0.7652	0.0634	0.0472	0.0501
f_w_h	0.5367	0.0510	0.0413	0.0448

TABLE III
TIME CONSUMPTION OF DIFFERENT SYSTEMS (UNIT:MS)

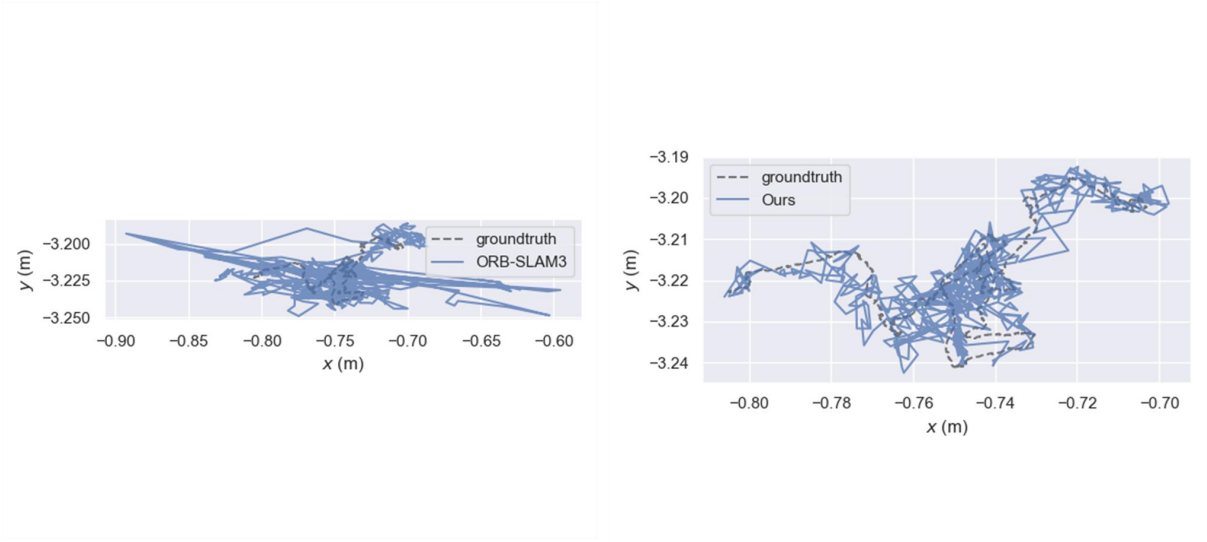
Sequences	ORB-SLAM3	DS-SLAM	PSPNet-SLAM	Ours
f_s_s	48.73	141.12	175.32	71.60
f_w_x	54.61	140.73	273.26	82.41
f_w_s	51.20	143.26	194.13	74.12
f_w_r	55.36	141.67	221.73	80.53
f_w_h	53.81	146.56	256.17	86.32



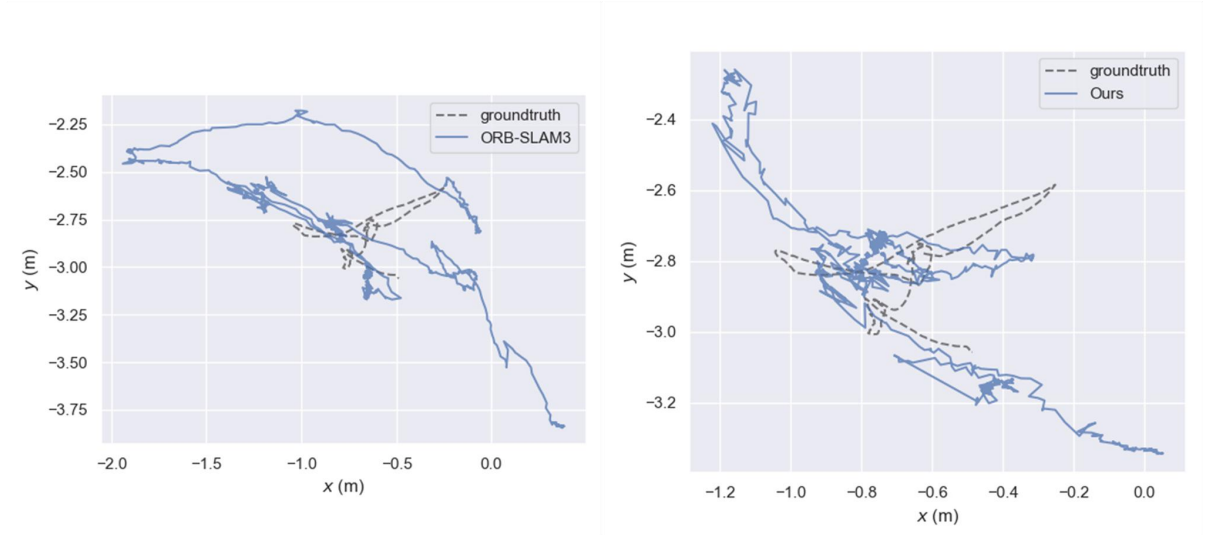
(a) f_s_s



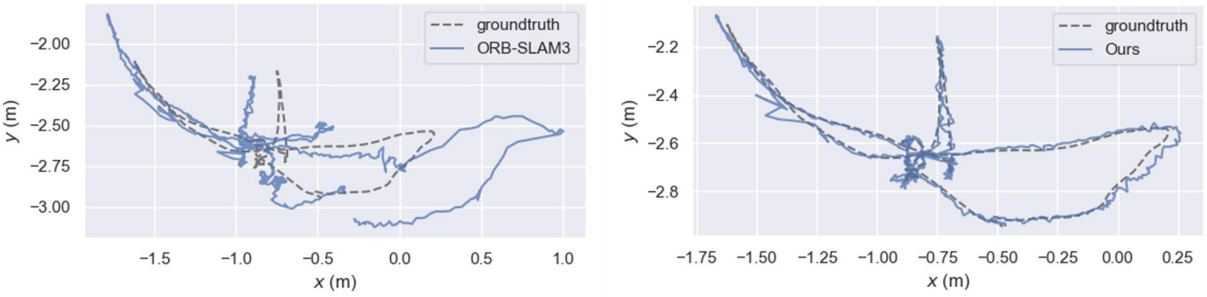
(b) f_w_x



（c）f_w_s



（d）f_w_r



（e）f_w_h

Fig. 7. A comparative analysis of the projected trajectories generated by the system presented in this paper and those produced by ORB-SLAM3 (the left portion of the figure presents a comparison between ORB-SLAM3 and the actual trajectories, while the right portion illustrates a comparison between the system proposed in this paper and the real trajectories).

V. Conclusion

This study presents a real-time VSLAM system that is founded on a lightweight PSPNet. To improve the efficiency of image semantic segmentation, the original backbone network, ResNet, of the PSPNet is substituted with the more lightweight MobileNetV2. The outcomes of the semantic segmentation process are subsequently employed within the tracking component of the system. Feature points located outside the semantic segmentation boundary are utilized to address the single response matrix, thereby enhancing its robustness. Simultaneously, feature points situated within the semantic segmentation boundary undergo additional screening through the optical flow method, which effectively discards DFP while preserving a greater number of SFP. In conclusion, the retained SFP are employed for subsequent position estimation, thereby enhancing the localization accuracy of the system within DE. Comparative experiments were performed between the proposed system and other prominent VSLAM systems using the TUM RGB-D dataset. The results of these experiments indicate that the proposed system not only achieves high localization accuracy in dynamic settings but also satisfies real-time operational requirements.

The system presented in this paper exhibits potential for enhancement. While it demonstrates superior real-time performance compared to other VSLAM systems in DE, there remains a need for further improvement in its accuracy to effectively address more intricate dynamic scenarios. Conversely, the system's applicability across various contexts, including outdoor settings, requires enhancement to fulfill diverse practical requirements. Future research endeavors will focus on addressing these two dimensions.

References

[1] Bresson, Guillaume, et al. "Simultaneous localization and mapping: A survey of current trends in autonomous driving." IEEE Transactions on Intelligent Vehicles 2.3 (2017): 194-220.

[2] Li, Ruihao, Sen Wang, and Dongbing Gu. "Ongoing evolution of visual slam from geometry to deep learning: Challenges and opportunities." Cognitive Computation 10.6 (2018): 875-889.

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

[4] Mur-Artal, Raul, Jose Maria Martinez Montiel, and Juan D. Tardos. "ORB-SLAM: a versatile and accurate monocular SLAM system." IEEE Transactions on Robotics 31.5 (2015): 1147-1163.

[5] Klein, Georg, and David Murray. "Parallel tracking and mapping for small AR workspaces." 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE, 2007.

[6] Mur-Artal, Raul, and Juan D. Tardós. "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras." IEEE Transactions on Robotics 33.5 (2017): 1255-1262.

[7] 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.

[8] Zhao, Hengshuang, et al. "Pyramid scene parsing network." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

[9] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[10] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[11] Shao, Feifei, et al. "Deep learning for weakly-supervised object detection and localization: A survey." Neurocomputing 496 (2022): 192-207.

[12] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[13] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in Neural Information Processing Systems 28 (2015).

[14] 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.

[15] 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.

[16] Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE Transactions on Pattern Analysis and Machine Intelligence 40.4 (2017): 834-848.

[17] 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.

[18] Li, Gongyang, Zhi Liu, and Haibin Ling. "ICNet: Information conversion network for RGB-D based salient object detection." IEEE Transactions on Image Processing 29 (2020): 4873-4884.

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

[20] Bolya, Daniel, et al. "Yolact: Real-time instance segmentation." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[21] Xie, Enze, et al. "Polarmask: Single shot instance segmentation with polar representation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[22] Wang, Xinlong, et al. "Solo: Segmenting objects by locations." Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16. Springer International Publishing, 2020.

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

[24] 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.

[25] Long, Xudong, Weiwei Zhang, and Bo Zhao. "PSPNet-SLAM: A semantic SLAM detect dynamic object by pyramid scene parsing network." IEEE Access 8 (2020): 214685-214695.

[26] Yu, Naigong, et al. "Drso-slam: A dynamic rgb-d slam algorithm for indoor dynamic scenes." 2021 33rd Chinese Control and Decision Conference (CCDC). IEEE, 2021.

[27] Gong, Han, et al. "AHY-SLAM: Toward faster and more accurate visual SLAM in dynamic scenes using homogenized feature extraction and object detection method." Sensors 23.9 (2023): 4241.

[28] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[29] Han, Shuangquan, and Zhihong Xi. "Dynamic scene semantics SLAM based on semantic segmentation." IEEE Access 8 (2020): 43563-43570.

[30] Chen, Liang-Chieh, et al. "Semantic image segmentation with deep convolutional nets and fully connected crfs." ArXiv Preprint ArXiv:1412.7062 (2014).

[31] Yu, Fisher, and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions." ArXiv Preprint ArXiv:1511.07122 (2015).

[32] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." ArXiv Preprint ArXiv:1704.04861 (2017).

[33] Zhao, Yao, et al. "KSF-SLAM: a key segmentation frame based semantic SLAM in dynamic environments." Journal of Intelligent & Robotic Systems 105.1 (2022): 3.

[34] Su, Peng, Suyun Luo, and Xiaoci Huang. "Real-time dynamic SLAM algorithm based on deep learning." IEEE Access 10 (2022): 87754-87766.

[35] 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.