Real-Time Visual-Inertial Localization Using Semantic Segmentation Towards Dynamic Environments

XINYANG ZHAO\(^1,2\), CHANGHONG WANG\(^1\), (Senior Member, IEEE), AND MARCELO H. ANG, JR.\(^2\), (Senior Member, IEEE)

\(^1\)School of Astronautics, Harbin Institute of Technology, Harbin 150001, China
\(^2\)Advanced Robotics Centre, Department of Mechanical Engineering, National University of Singapore, Singapore 117608

Corresponding author: Changhong Wang (cwang@hit.edu.cn)

This work was supported in part by the China Scholarship Council under Grant 201906120060.

ABSTRACT Simultaneous localization and mapping (SLAM), focusing on addressing the joint estimation problem of self-localization and scene mapping, has been widely used in many applications such as mobile robot, drone, and augmented reality (AR). However, traditional state-of-the-art SLAM approaches are typically designed under the static-world assumption and prone to be degraded by moving objects when running in dynamic scenes. This article presents a novel semantic visual-inertial SLAM system for dynamic environments that, building on VINS-Mono, performs real-time trajectory estimation by utilizing the pixel-wise results of semantic segmentation. We integrate the feature tracking and extraction framework into the front-end of the SLAM system, which could make full use of the time waiting for the completion of the semantic segmentation module, to effectively track the feature points on subsequent images from the camera. In this way, the system can track feature points stably even in high-speed movement. We also construct the dynamic feature detection module that combines the pixel-wise semantic segmentation results and the multi-view geometric constraints to exclude dynamic feature points. We evaluate our system in public datasets, including dynamic indoor scenes and outdoor scenes. Several experiments demonstrate that our system could achieve higher localization accuracy and robustness than state-of-the-art SLAM systems in challenging environments.

INDEX TERMS Simultaneous localization and mapping, dynamic environment, semantic, visual-inertial system.

I. INTRODUCTION

In the past thirty years, with the rapid development of computer science and sensors, simultaneous localization and mapping (SLAM) has become an indispensable technology in many fields, like robotics [1], [2], autonomous driving [3], [4], and Augmented Reality (AR) [5], [6]. Benefited from the development of computer vision, visual SLAM (V-SLAM) has attracted the attention of many researchers and companies with its advantages of low cost, low power consumption, and the ability to provide rich information of sceneries. It has become a research hotspot in the field of SLAM. Although the current state-of-the-art V-SLAM algorithms work well in static environments, they are always prone to failure when confronted with dynamic scenes. Real-world environments, such as shopping malls, streets, and stations, usually have various moving objects. Therefore, to improve the localization accuracy and robustness of the SLAM system in dynamic environments, it is crucial to avoid the interference of dynamic objects on the system effectively.

Most traditional V-SLAM approaches are designed based on the assumption that the environment is static. They just simultaneously estimate camera pose and 3D landmarks through extracting feature points, both static and dynamic, from images. Identifying and excluding feature points on dynamic objects is an effective way to eliminate the impact of dynamic environments on the system [7].
Recent studies [8], [9] have shown that feature points can be effectively classified based on the results of semantic segmentation. And then, only feature points located on static objects are allowed to participate in subsequent calculations. The application of deep learning technology can improve the performance of the SLAM system in dynamic environments. However, the deep learning method can only judge whether the object is dynamic or not according to the manual set prior knowledge, such as pedestrians, animals, and any other objects that have the moving tendency. It could not identify the dynamic feature points on objects with uncertain motion states such as books, chairs, and vehicles. Besides, due to the expensive computation of the convolutional neural network, its integration will significantly increase the latency of the SLAM system. Therefore, once the camera moves rapidly, the pose estimation is likely to drift due to the feature point tracking failure.

To address these problems above, we propose a real-time SLAM system for dynamic environments, which combines semantic segmentation and multi-view geometric constraints, can effectively identify and avoid using the feature points located on dynamic objects. This system adopts a novel visual front-end based on VINS-Mono [10], which is consists of three threads, RGB-image manager, semantic segmentation manager, and feature point processing. The first two threads are mainly used to store the RGB-images from the input sensor and pixel-wise result from the semantic segmentation network. Feature point processing thread, running independently of the input sensor, utilizes the optical flow to track feature points on the frames obtained from the RGB-image manager. When the pixel-wise image is output from semantic segmentation thread, the corresponding feature points located on the prior dynamic objects(such as people and animals) will be marked dynamic labels. After that, we adopt the multi-view geometric constraint to detect and mark other dynamic feature points furtherly. Finally, if the number of static feature points is insufficient, new feature points will be extracted from the static area of the image. In this way, only static feature points are allowed to estimate the camera state. On the other hand, the fusion of IMU data, which tightly-coupled with monocular, can make up for the low frequency of camera pose estimation caused by the semantic segmentation thread, ensuring the accuracy and robustness of localization and mapping. In summary, the main contributions of this article are listed as follows:

1) The expensive computation in semantic segmentation is prone to decrease the running frequency of the SLAM system and result in the feature point tracking failure. An efficient feature points processing framework is proposed in this article. When the semantic segmentation thread is busy, it can continuously track the feature points of the latest image frame for the next execution loop. Therefore, while waiting for the completion of semantic segmentation, each image frame of the camera can be fully utilized to ensure the tracking effect of feature points.

2) Since the feature points on dynamic objects will significantly degrade the SLAM performance, two steps are used in this system to eliminate the influence of dynamic objects. Firstly, we present the dynamic feature point detection method that combined deep learning technology and multi-view geometric constraints to exclude dynamic feature points. Secondly, we adopt the feature points extraction algorithm based on the semantic mask in this system to extract new feature points in the static area more uniformly when the static feature points are insufficient.

3) A complete semantic visual-inertial SLAM system is constructed based on VINS-Mono in this article, which can significantly reduce the influence of dynamic objects on the system, achieving better accuracy and robustness in the challenging dataset, ADVIO.

The rest of this article is structured as follows. We first discuss the related work of our system in Section II. The main work is introduced and demonstrated in Section III, then a series of experiments and evaluation of results are shown in Section IV. Finally, Section V concludes the article and discusses directions for future work.

II. RELATED WORK

In recent years, visual SLAM has attracted a large number of researchers with its advantages of low cost and wide application, has made rapid development. MonoSLAM [11], which was proposed in 2003, is considered to be the first pure visual SLAM system that can complete camera pose estimation and feature measurement in real-time on a desktop PC through the Bayesian framework. In 2007, Klein et al. proposed PTAM [12], which is composed of two thread: feature points tracking and mapping. The front-end of the system tracks the feature points in real-time, and the back-end uses nonlinear optimization to achieve pose estimation. Since then, this combination of front-end and back-end framework has been widely used in subsequent visual SLAM systems [13], [14]. ORB-SLAM [15], [16], which was proposed by Mur-Artal et al. in 2015 and 2017, is the remarkable SLAM system. It innovatively utilizes three threads: tracking, local Bundle Adjustment(BA), and global BA to realize pose estimation and landmark measurement, achieving remarkable tracking and mapping results. However, the vision-only approaches only rely on the observation of environmental features to achieve pose estimation. So, the performance of the visual-only SLAM system will be hindered by the drift of pose estimation due to the rapid motion, or dramatic change in illumination. Besides, the lack of scale information of the real world also limits the application of the monocular SLAM system in robot navigation. As a sensor that can measure its angular velocity and acceleration, IMU has long been used to assist systems to achieve localization [17], [18], [19], and it has complementary sensing capabilities with the camera. This visual and inertial fusion system is called VIO(Visual Inertial Odometry). At present, the main...
framework of VIO is to use tightly-coupled methods to fuse the state of the IMU and the camera, which jointly construct the motion equation and the observation equation to realize the state estimation. Based on Forster’s pre-integration theory [20] of IMU, Mur-Artal et al. proposed a novel tightly coupled visual-inertial SLAM system ORB-VISLAM [16]. This system has the ability of loop-closure detection and map reuse, which can achieve higher accuracy and robustness. Recently, another tightly-coupled visual-inertial system based on monocular and IMU, VINS-Mono [10], has been proposed. This method can achieve camera-IMU extrinsic calibration and IMU bias correction online, have good relocalization, pose graph reuse, and loop detection capabilities. VINS-Mono has played an outstanding role in the application of drones and augmented reality (AR). However, it is still necessary to continue exploring based on these state-of-the-art methods, especially to improve its performance in dynamic scenarios.

In order to avoid the interference of dynamic objects on the system, the system usually divides the feature points into two clusters, static and dynamic. Only static feature points are used for pose estimation and 3D landmarks reconstruction. Standard visual SLAM approaches, such as ORB-SLAM and VINS-Mono, usually using Random Sample Consensus (RANSAC) [21] to exclude feature points that do not conform with the geometric model. However, RANSAC will work well only in the condition that the static feature points are the majority, rather than the situation that the dynamic objects in front of the camera are dominant. Recently, several methods have been used to improve the performance of the SLAM problem towards dynamic environments. These methods can be roughly divided into two types: geometric constrains and deep learning.

The former approach utilizes the epipolar geometric constraints [22] defined by multi-view geometry for static scenes to exclude the dynamic feature points that violate the rules. Kundu et al. [23] use two different multi-view geometric constraints to divide the pixels into two categories: dynamic and static. The first constraint is from epipolar geometry, which requires static points to lie on its epipolar line in subsequent images. The second one utilizes the robot motion to detect the dynamic points that do not meet the estimated bound in the image pixel position along the epipolar line. Lin and Wang [24] proposed a stereo-based SLAM and moving object tracking (SLAMMOT) system instead of that using a single camera. Dynamic feature point was detected by testing the system performance with and without adding this point into the pose estimation. Although the system can classify dynamic and static feature points with high accuracy, it could not cope with the sudden appearance objects. Sun et al. [25] using RGB-D data-based motion removal approach to exclude the feature points located on moving objects. This approach can effectively improve RGB-D SLAM performance in dynamic environments without prior knowledge from moving objects. However, this approach could not cope with the situation that dynamic objects dominated the scene as the possible foreground and used to update the foreground model.

The deep learning method adopts semantic information to detect dynamic feature points. With the rapid development of computer vision, object detection and semantic segmentation based on deep learning have higher accuracy and are widely used in SLAM systems. The new image frame will be processed by CNN architectures, like SegNet [26], Mask R-CNN [27], YOLOv3 [28]. The static and dynamic regions in the image will be segmented according to the results of CNN networks. Zhang et al. [29] use YOLO to get the semantic label of each feature point and exclude those located on the dynamic objects defined by prior knowledge. Xiao et al. [30] proposed the Dynamic-SLAM system based on SSD [31] convolution neural network. They use a selection tracking algorithm to detect the dynamic points, which dramatically improves the performance of the system in dynamic environments. Compared with the geometric method, the learning method can identify dynamic objects from a single image, enabling the system to detect prior dynamic objects in the initial stage. Its primary disadvantage is that it could not figure out the moving objects that should be static in prior knowledge [32]. Due to the complementarity of the geometric and deep learning methods, their combination is an effective method to improve the accuracy and robustness of the system in challenging environments. DS-SLAM [33] proposed by Yu et al. combining semantic segmentation network with epipolar constraints outperforms ORB-SLAM2 in dynamic environments. DynaSLAM [32] adopted Mask R-CNN together with multi-view geometry, improves the accuracy of pose estimation, and constructs a static map in dynamic scenes.

At present, although the SLAM system towards dynamic environments has achieved excellent performance, it still has two limitations. First of all, due to the time-consuming operation of the neural network, many current methods are difficult to run in real-time in mobile processors. Secondly, most current SLAM systems for dynamic environments only use visual sensors to achieve pose estimation, which will undoubtedly limit the robustness of the system in complex environments such as low texture or significant illumination changes. We integrate the novel dynamic feature point detection method with the visual-inertial SLAM system to ensure real-time performance and improve the accuracy and robustness of pose estimation.

III. SEMANTIC VISUAL-INERTIAL SYSTEM

A. SYSTEM OVERVIEW

To solve the problem that VINS-Mono is easy to degenerate by moving objects in dynamic environments, we integrate semantic segmentation network and dynamic feature point detection module into the VINS-Mono system. Fig.1 shows the flowchart of our approach. It consists of five main threads: semantic segmentation thread, feature points processing thread, tightly-coupled optimization thread, loop
FIGURE 1. The flowchart illustrating the full pipeline of the proposed SLAM system. The measurement preprocessing module as the front end of the SLAM system integrates the Semantic Segmentation, Feature Detection and Tracking, and Remove Outliers, which are highlighted with red boxes in the figure. They act as a pre-processing stage to eliminate feature points located on dynamic objects. Visual-Inertial Odometry and Loop Closure as the back end of the system optimize the states to generate high accuracy pose estimation and 3D landmarks.

FIGURE 2. Deeplab V3+ network structure based on MobileNetV2 skeleton.

detection thread, and pose graph optimization thread. Our semantic segmentation thread contains the CNN named Deeplab V3+, which is based on the work of Chen et al. [34]. It performs frame-by-frame segmentation and runs in parallel with feature points processing thread. The feature points processing thread detects and discards dynamic feature points as outliers by combining the pixel-wise results from semantic segmentation thread and multi-view geometric constraints. Meanwhile, to fully utilize the waiting time when the semantic segmentation is busy, it performs the optical flow [35] to track the remaining static feature points in the latest image from the camera for the next loop.

The details of the semantic segmentation and dynamic feature processing modules are presented in this section.

B. SEMANTIC SEGMENTATION NETWORK ARCHITECTURE
There are many excellent semantic segmentation networks in the academic field, such as HarDNet [36], LiteSeg [37], and DeepLabv3+ [34], which have excellent performance in both accuracy and real-time. After our test, DeepLabv3+ has higher real-time performance than the other two networks, making it more suitable for this system. In the semantic segmentation module, we adopt DeepLabv3+ to depict foreground objects from the background at pixel-level. DeepLabv3+ is constructed based on Deeplabv3, which introduces a novel encoder-decoder structure used for semantic segmentation, achieved 89.0% mean intersection over union (mIoU) in PASCAL VOC 2012 dataset and 82.1% mIoU in Cityscapes dataset. In that structure, it can arbitrarily control the resolution of extracted encoder features by atrous convolution to effectively capture rich contextual and detailed target boundaries with a coarse-to-fine recovery of spatial information.

As is shown in Fig. 2, at the encoder stage, DeepLabv3+ in our approach employs MobileNetV2 [38] to extract the features from the input image at arbitrary resolution. Then, the ratio of input image spatial resolution to the output resolution, in the end, is denoted by output stride. The encoder output of the encoder-decoder structure is the last feature map, which contains 256 channels and rich semantic information. At the same time, it is also the input into the global pooling layer for feature extraction. As for the decoder module, the $1 \times 1$ convolution can reduce the channel of the low-level feature map from the encoder module, and prevent the prediction results from tilting to low-level features. The encoder features are bilinearly upsampled to concatenate with the corresponding low-level features from the network backbone. After the concatenation, the $3 \times 3$ convolution is used to obtain sharper segmentation results. Finally, the four times up-sampling produces the semantic label of each pixel.

The DeeLabv3+ could use different perception backbones with different deployment platforms in mind. Considering that this system will be used on mobile devices, we adopt MobileNetV2, a lightweight model with about 10 megabytes of weights as the backbone of DeepLab v3+. We employ the DeepLab v3+, integrated with TensorFlow and Robotic Operation System (ROS) in our approach. It can produce pixel-wise semantic segmentation in the format of the ROS image message, which is convenient for the system to use. The DeepLab model we use in this architecture was trained on the PASCAL VOC dataset [39] that contains 20 classes in total (airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, tv). We consider that the objects likely to appear for most dynamic environments are included within this list. The network, trained on PASCAL VOC, could be fine-tuned with new training data if other classes were needed.

The output of the network, assuming that the input is an RGB image of size $m \times n \times 3$, is a matrix of size $m \times n \times l$, where $l$ represents the label of objects in the classes list. For each
output channel $i \in l$, a corresponding color is obtained. By combining all the channels, we can get the contour of all the potential dynamic objects in the image.

### C. EFFICIENT FEATURE POINTS TRACKING

By using DeepLab v3+, the feature points located on the dynamic objects in the image could be labeled as dynamic and not used for tracking and mapping anymore. However, since the limited computing capability of mobile robots, the expensive computation of semantic segmentation network will seriously affect the real-time performance of the system.

To deal with these problems, some recent proposed SLAM methods [32], [40], [41] using semantic segmentation thread, which runs in parallel with other threads, performing frame-by-frame segmentation from the camera. Since the methods mentioned above adopt the simple parallel strategy that the fastest thread will wait until the slowest thread completes its work. The frame-rates of feature tracking will not more than that of the semantic segmentation module. Unlike the methods mentioned above, which runs on high-performance GPU. Considering the portability, cost and power consumption, this system runs on GTX860M, a low-performance GPU. Meanwhile, due to the limited buffer space for the inputs, if the image in the cache is not processed, it will be replaced by the new input image from the camera. Therefore, in the case of low-performance GPU, if we adopt the same architecture mentioned above, it will undoubtedly significantly affect the performance of feature tracking. Once the robot moves rapidly, the parallax of the two frames processed by the tracking thread will increase, which will easily lead to the loss of feature tracking and localization failure. Fig.3 shows the comparison of feature point tracking effects between original optical flow architecture and our efficient optical flow tracking architecture. Our approach is detailed in this section.

We propose a novel feature point tracking method in our system, which uses multi-threading and mutex to make the feature tracking module track the feature points on all the images from the camera, to prevent the image loss caused by the time-consuming of semantic segmentation network. Fig.4 shows the framework of the feature point tracking module. Three threads were adopted in this framework. RGB image manager holds all of the images from the camera. The semantic image manager stores the image from the semantic segmentation module. As the core of the feature tracking module, feature point processing thread includes tracking existing feature points, extracting new feature points, identifying and excluding dynamic feature points. The following section will outline each component in more detail.

1) **OPTICAL FLOW TRACKING**

In our approach, we use a novel image management method that sets a container named ImageBuf for temporarily storing images from the camera. The system loads the raw images from imagebuf for feature tracking and semantic segmentation. Donate $I = \{I_1, I_{j+1}, \cdots, I_{k-1}, I_k\}$ as the image sequence loaded from ImageBuf to be processed.

Semantic segmentation and feature point processing are running in parallel. Feature point matchings between $I_{j-1}$ and $I_j$ are found by the Lucas-Kanade Optical Flow [35]. In the experiment, we find that semantic segmentation is still not completed after the new feature point extraction, dynamic feature point removal and feature point publishing modules. Therefore, to further improve the system performance, we use the remaining time to complete the feature point tracking from $I_{j+1}$ to $I_{k-1}$, and the result will be used for the next loop. Fig.3 shows that our method can track feature points well.

2) **SEMANTIC MASK BASED FEATURE POINTS EXTRACTION**

Only using the optical flow tracking method in SLAM systems will inevitably lose some feature points due to the scene changes caused by ego-motion. The existing feature points will be less and less until it is insufficient to support the pose calculation. New feature points need to be extracted from images to maintain a minimum number(100-300) to ensure enough points to participate in triangulation and pose estimation. However, dynamic objects (such as people, dogs, vehicles, etc.) in the environment usually have rich
FIGURE 4. The feature point tracking module is composed of three main parallel threads: RGB-image manager, semantic image manager, and feature point processing. The RGB-image and semantic image manager preprocesses the monocular input so that the feature point processing thread running independently of the input sensor. Frame Pairs is divided into two parts, one of which is the image \(I_j\), and the feature point set \(P_j\) on it is obtained by optical flow, tracking the feature point set \(P_{j-1}\) on previous frame \(I_{j-1}\). The other part is the image set \(I_{j+1}, \ldots, I_{k-1}\). It is used to track the feature points from time \(j+1\) to \(k-1\) for the next loop.

At this problem, we calculated \(\tilde{S}_i\), the percentage of pixels covered by dynamic objects in the mask. Then, the distance constraint \(d\) is expressed as:

\[
d = \rho \tilde{S}_i
\]

where \(\rho\) is the decision coefficient and \(\tilde{S}_i\) is the decision threshold, which is calculated as follows:

\[
\tilde{S}_i = \frac{N_D}{N_I}
\]

where \(N_D\) and \(N_I\) represent the number of pixel on dynamic objects and total number of pixels on the mask.

Fig. 5 shows an example of dynamic feature point detection based on deep learning, multi-view geometry, and semantic mask-based feature point extraction method. As we can see from the last column, combining the results of semantic segmentation and current static feature points, the mask can effectively help the feature extractor avoid the dynamic region and existing feature points when extracting feature points.

3) DYNAMIC FEATURES DETECTION

Using DeepLab v3+, we can detect most of the feature points pre-defined by prior knowledge on dynamic objects (such as people, animals, cars). However, not all pre-defined dynamic objects, such as sleeping dogs and parked cars, are always moving. The system is also unable to detect changes in static objects, such as tables and chairs being moved, or even changes in indoor layout in long-term SLAM. These moving objects easily lead to data association errors in the front-end of the SLAM system, affecting the performance of subsequent feature point triangulation and pose estimation. Let \(\pi_{k-1}^{k-1}\) denote the fundamental matrix transformation from two consecutive frames \(c_{k-1}\) and \(c_k\). The reprojection error for feature point matchings is expressed as:

\[
r(\lambda_{j}^{k-1}, \lambda_{j}^{k}; \pi_{k-1}^{k-1}) = \|\lambda_{j}^{k-1} - \pi_{k-1}^{k-1} \lambda_{j}^{k}\|
\]
they are dynamic. As shown in Fig. 5, we first label the feature points located on the objects belonging to Potential Dynamic as dynamic points. The remaining feature points labeled as static points are processed by RANSAC to exclude feature points that do not conform with the fundamental matrix. In this way, the dynamic objects’ influence on the calculation of the fundamental matrix can be minimized.

As mentioned above, we assume that all the feature points located on Potential dynamic objects are dynamic and then exclude them to avoid the interference of dynamic objects on the calculation of the Fundamental matrix. That means the fewer feature points are involved in subsequent triangulation and pose estimation, which will undoubtedly greatly affect the performance and robustness of triangulation, especially in environments with many dynamic objects. Considering that there are still some objects that are actually static in the scene but belong to the categories of Potentially Dynamic, the feature points on them are incorrectly classified as dynamic when calculating the fundamental matrix. Based on the fact that static feature points comply with the standard constraint defined for static scenes in multi-view geometry, we use that constraint to identify whether the feature points on Rigid Potential Dynamic objects are static. Given a pair of images, as shown in Fig. 6, for a 3D point on a moving object \( X_i \), whose projections on the first and second images are \( x_i^{k-1} \) and \( x_i^k \), respectively. And \( X_i^{k-1}, X_i^k \) are their homogeneous coordinates:

\[
\begin{align*}
X_i^{k-1} &= \begin{bmatrix} u_1, v_1 \end{bmatrix}, & x_i^{k-1} &= \begin{bmatrix} u_2, v_2 \end{bmatrix} \\
X_i^k &= \begin{bmatrix} u_1, v_1, 1 \end{bmatrix}, & x_i^k &= \begin{bmatrix} u_2, v_2, 1 \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\lambda_i^{k-1} &= \arg \min_{\pi} \sum_i r(\lambda_i^{k-1}, \pi_k^{k-1})
\end{align*}
\]

where \( \lambda_i^{k-1} \) is the location of pixel \( i \) in the frame \( c_{k-1} \), \( \lambda_i^k \) is the corresponding pixel location in the consecutive frame \( c_k \). \( \parallel \parallel \) represents Euler distance between the two pixels. \( \pi_k^{k-1} \) is found with the following formula:

where \( x_i^{k-1} \) is the location of pixel \( i \) in the frame \( c_{k-1} \), \( \lambda_i^k \) is the corresponding pixel location in the consecutive frame \( c_k \). \( \parallel \parallel \) represents Euler distance between the two pixels. \( \pi_k^{k-1} \) is found with the following formula:

The formulation in (5) is founded by epipolar geometry. The less the proportion of outlier feature points matching in the calculation, the higher its accuracy. At present, the state-of-the-art SLAM approaches like [10], [16] adopt Random Sample Consensus (RANSAC) [21] to exclude feature points that do not conform the fundamental matrix. This algorithm works well in static cases but is prone to be corrupted when the dynamic objects in front of the camera are dominant. We combine deep learning methods and geometric constraints to deal with dynamic scenes. The semantic segmentation module pixel-wisely segments the objects, and the category of objects is detailed in Table 1. According to the categories of common objects in our daily life, we generally divide them into two categories: static and potential dynamic.

The static objects include dining tables, sofa, TV, and so on, while potential dynamic ones are further subdivided into rigid and non-rigid objects. For non-rigid objects (such as people, cats, dogs), that cannot keep absolute static, the feature points located on them need to be completely excluded. It is necessary to completely exclude the feature points on the object (e.g. human, cat, dog) that cannot be absolutely static. For those rigid objects that may be static or dynamic (such as bicycles or cars), the system needs to judge further whether

**TABLE 1. Dynamic categories of common objects in daily life.**

| Static          | Non-rigid     | Rigid          |
|-----------------|---------------|----------------|
| background      | bird          | aeroplane      |
| diningable      | cat           | bicycle        |
| pottedplant     | cow           | boat           |
| sofa            | dog           | bottle         |
| train           | person        | bus            |
| tv              | sheep         | car            |
| chair           |               |                |

![FIGURE 5](image1.png)

This figure shows the process of excluding dynamic feature points and extracting new feature points. The shown scenario is that some customers hang out in a mall. Different kinds of feature points are displayed in different colors: the regular feature points are green, feature points on dynamic semantic objects are yellow, feature points that do not conform the fundamental matrix are purple, and newly extracted feature points are light blue. Firstly, two consecutive RGB images are used as input for optical flow tracking. Secondly, semantic segmentation and epipolar geometric constraints are used to exclude dynamic feature points. Finally, the new feature points are extracted based on the mask, which combines the remaining static feature points and semantic segmentation. The figure is best viewed in color.

![FIGURE 6](image2.png)

Left figure represents the static scenes, the transformation of feature points from \( X_i \) and \( x_2 \) is defined by epipolar constraint:

\[
(x_i^{k-1})^T F x_i^k = 0
\]

Right figure shows the dynamic scenes, the feature point \( x_i^{k} \) in the second frame lies too far from the epipolar line \( l_i' \).

where \( (u_1, v_1) \), \( (u_2, v_2) \) are the pixel coordinates of \( x_i^{k-1} \) and \( x_i^k \) respectively. The epipolar line \( l_i' \) is the image in the second

\[
\lambda_i^{k-1} = \arg \min_{\pi} \sum_i r(\lambda_i^{k-1}, \pi_k^{k-1})
\]

\[
\begin{align*}
X_i^{k-1} &= \begin{bmatrix} u_1, v_1 \end{bmatrix}, & x_i^{k-1} &= \begin{bmatrix} u_2, v_2 \end{bmatrix} \\
X_i^k &= \begin{bmatrix} u_1, v_1, 1 \end{bmatrix}, & x_i^k &= \begin{bmatrix} u_2, v_2, 1 \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\lambda_i^{k-1} &= \arg \min_{\pi} \sum_i r(\lambda_i^{k-1}, \pi_k^{k-1})
\end{align*}
\]
view of the ray back-projected from $X_i$ in the condition of $X$ is static, and it can be computed by the following equation:

$$l'_k = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = F_{k-1}^{-1}X_{k-1} = F_{k-1}^{-1} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}$$  \hspace{1cm} (8)$$

where $a,b$ and $c$ are the vector of epipolar line $l'_k$ and $F_{k-1}$ is the fundamental matrix of the two consecutive image $c_{k-1}$ and $c_k$. If the 3D point $X$ is static, its projection point lying on the second image should be on epipolar line $l'$. Therefore, the distance $D$ from a feature point to its corresponding epipolar line can reflect whether the point is dynamic or not, which could be determined as follows:

$$\varrho_k^i = \frac{||X_i^T F_{k-1}^{-1}X_{k-1}||}{\sqrt{||a||^2 + ||b||^2}}$$  \hspace{1cm} (9)$$

We donate $\delta$ as the threshold to determine whether the feature points are dynamic or not, and the way to judge it is to calculate the distance $\varrho_k^i$ of each feature point by using formula(9). If $\varrho_k^i > \delta$, the feature point will be labeled as dynamic. Otherwise, it will be labeled as static.

### IV. EXPERIMENT AND RESULTS

We perform challenging datasets to verify our approach’s robustness and accuracy in dynamic scenes, including indoor and outdoor evaluation using the ADVIO dataset [43] and initialization test in dynamic scenes.

The experiments are performed on a laptop with Intel Core i7-4710MQ CPU(4-core 2.5GHz), 16GB of RAM, NVIDIA GTX860M GPU with 2GB of graphic memory for semantic segmentation module.

#### A. ACCURACY TEST ON ADVIO DATASET

Real indoor environments, like malls, museums, are often dynamic and complicated. However, most open-source SLAM algorithms are designed on the premise of static environments and do not have sound localization and mapping performance in dynamic scenes. To fully evaluate the performance of our system, the ADVIO DATASET, which is designed for pinpoint differences in published methods, is used in this section. ADVIO DATASET develops a set of versatile and challenging real-world computer vision benchmark sets for visual-inertial odometry. This dataset comprises 23 sequences recorded with an iPhone, a Google Pixel Android phone, and a Google Tango device in different indoor and outdoor scenes. Both the RGB and the IMU data are available, together with the ground-truth trajectory.
FIGURE 8. The experimental comparison results between VINS-Mono and the proposed system for each sequence. The left column in each subgraph depicts the trajectories of VINS-Mono, our proposed system, and the ground truth. The right column shows the comparison absolute pose error (APE) of translation between the two methods.

For a comprehensive comparison, we use the Root Mean Square Error (RMSE), Mean Error and Standard deviation (Std.) of Absolute Pose Error (APE) [44] of the keyframe trajectories to evaluate the performance of the proposed method. Since the Visual-inertial system can estimate the scale information, we consider using SE(3) Umeyama alignment to align the estimated trajectory with GroundTruth.
In Table 2, Table 3, and Fig. 8, we compare our absolute pose error (APE) accuracy for seven sequences of ADVIO dataset to the following state-of-the-art methods: VIORB-SLAM [45], MAPLAB [46], VINS-Mono [10].

1) INDOOR ACCURACY TEST
As shown in Fig.8, the first five sequences of the dataset recorded indoor scenes. The camera recorded different routes passing through large shopping malls and subway stations in the sequences of 02, 03, 05, 06, and 12. The dynamics of the sequence were also divided into low, moderate, and high according to the number of dynamic objects [43].

As can be seen clearly from sequences 02 to 12 in Table 2 and Table 3, our method has better performance in most sequences. Due to the influence of dynamic objects in the dataset, VIORB-SLAM cannot even achieve the initialization of the system. Although Maplab can sometimes complete initialization in the middle of the dataset, it often re-initializes due to the severe drift of pose estimation caused by the bad initialization parameters or rapid movement. Unlike previous methods, VINS-Mono, with the assistance of IMU, has better accuracy and robustness in fast motion and can get complete motion trajectories even in some challenging sequences of ADVIO. Our method, based on VINS-Mono, is equipped with the semantic segmentation capability, which could effectively exclude the feature points on dynamic objects like pedestrians, moving vehicles, has higher localization accuracy in dynamic environments. Fig. 7 shows the real-time distribution of feature points on the image of VINS-Mono and Our method in the indoor scenes. Our method can remove the feature points of dynamic objects (pedestrians) more effectively with the aid of semantic segmentation and multi-view geometry and extract as many feature points as possible in the static area. Compared with VINS-Mono, the accuracy of RMSE is improved by 1% ~ 31%

2) OUTDOOR ACCURACY TEST
To test the performance of our method in large-scale outdoor environments, we use the outdoor sequences 20 and 23, which contain urban outdoor scenes and suburban(campus) outdoor scenes in this experiment. Table 2 and Table 3 show the APE of our results for sequence 20 and sequence 23, comparing to VIORB-SLAM, MAPLAB, and VINS-Mono.

The results are similar to those in indoor cases. It can be seen from Table 2 and Table 3 that the accuracy of our method is 2% ~ 8% better than VINS-Mono in the aspect of absolute position error for both translation part and angle part.

Fig. 9 shows the real-time images of VINS-Mono and our approach. As we can see clearly, there are a certain number of feature points on dynamic objects like pedestrians or vehicles when using VINS-Mono, while our method can detect dynamic objects, effectively exclude dynamic feature points, and extract new feature points on the static background.

B. INITIALIZATION TEST IN DYNAMIC SCENE
The initialization procedure is vital for the Visual-inertial SLAM system, which could provide initial information, including gravity vector, scale factor, gyroscope bias, and the speed of each frame, for subsequent nonlinear state estimation. However, in the initialization process, the dynamic objects in the scene will bring huge errors to the initial parameter estimation, which will affect the subsequent pose estimation.

To test the performance of our system’s initialization in dynamic scenes, sequence 02 and 06 of the ADVIO dataset, which have abundant dynamic scenes, are used in this section. As is shown in Fig.10, VINS-Mono cannot distinguish whether the feature points are located on dynamic objects, and it roughly uses all the feature points for pose estimation. Unlike the former method, our method combines semantic segmentation and multi-view geometric constraints, which can effectively exclude dynamic feature points and perform state estimation only using static feature points, to have better initialization results. Fig.11 shows that the trajectory estimated by our method after initialization in a dynamic environment is more consistent with GroundTruth than VINS-Mono.

C. TIMING ANALYSIS
To evaluate the real-time performance of our method, we test the time consumption of several major modules. Table 4 shows the average time consumption of tracking thread and semantic thread. It can be seen that compared with other modules, the average time consumption of the semantic segmentation module is the longest, reaching 216.68ms.

To reduce the system delay caused by semantic segmentation, we add an independent thread to execute semantic segmentation. Simultaneously, the feature points processing thread could make full use of the time waiting for semantic segmentation, and perform feature point tracking and exclude
V. CONCLUSION AND DISCUSSION

In this article, we propose a visual-inertial SLAM system that, building on VINS-Mono, utilizes the deep-learning method to have better performance in dynamic environments. Our system combines the DeepLab V3+ semantic segmentation method with the geometric constraint-based approach, which can not only effectively identify the feature points on predefined dynamic objects, but also undefined dynamic objects. While waiting for the completion of the semantic segmentation thread, the optical flow method is adopted to track the feature points on the real-time images from the camera. In this way, the system can track the feature points stably even in the rapid movement. At the same time, a high-performance GPU is no longer indispensable for this system. A low-performance GPU can also achieve the same effect, which can significantly reduce the hardware cost of the system.

To test the performance of our method in dynamic environments, we carry out experiments in the public ADVIO dataset. The comparison against the state-of-the-art SLAM methods shows that our method achieves the highest accuracy in most sequences. In the test of ADVIO indoor dynamic environment sequences, our system shows 3.51% ~ 31.08% accuracy improvement against the original VINS-Mono and better than Maplab and VIORB-SLAM, which cannot even achieve pose estimation. In the test of outdoor sequences of ADVIO, our system does not perform as well as indoor scenes. Although in sequences 20 and 23, the localization accuracy is higher than VINS-Mono, in sequence 22, the APE of our system is lower than that of the original VINS-Mono. Because in sequence 22, there is a large-scale scene with static, low texture but high-speed motion. Compared with VINS-Mono, our system is more vulnerable in high-speed scenarios due to the expensive computation of the Convolution Neural Network. To further verify the initialization performance in dynamic scenes, we take the dynamic crowd scene as the start point of the system’s initialization. Our system completed the system’s initialization in the dynamic environment and got a high-precision trajectory, while VINS-Mono failed. The experiment results show that our approach is suitable for mobile robots or AR applications in indoor dynamic environments.

In further research, we will extend our research in two aspects. Firstly, we will use other features such as line features, planar features, and even semantic features as the supplement of point features to improve the robustness of the system in low texture or high dynamic environments. Secondly, we will combine semantic segmentation results with exploring the method of dense semantic mapping further to meet the navigation requirement.

REFERENCES

[1] F. Bonin-Font, A. Ortiz, and G. Oliver, “Visual navigation for mobile robots: A survey,” J. Intell. Robotic Syst., vol. 53, no. 3, pp. 263–296, Nov. 2008.
[2] D. Yang, S. Bi, W. Wang, C. Yuan, W. Wang, X. Qi, and Y. Cai, “DRE-SLAM: Dynamic RGB-D encoder SLAM for a differential-drive robot,” Remote Sens., vol. 11, no. 4, p. 380, Feb. 2019.

[3] G. Sibley, C. Mei, I. Reid, and P. Newman, “Vast-scale outdoor navigation using adaptive relative bundle adjustment,” Int. J. Robot. Res., vol. 29, no. 10, pp. 958–980, Oct. 2010. [Online]. Available: http://journals.ega.pub/doi/10.1177/0278364910369268

[4] S. Yang, S. A. Scherer, X. Yi, and A. Zell, “Multi-camera visual SLAM for autonomous navigation of micro aerial vehicles,” Robot. Auto. Syst., vol. 93, pp. 116–134, Jul. 2017.

[5] Q. H. Gao, T. R. Wan, W. Tang, L. Chen, and K. B. Zhang, “An improved augmented reality registration method based on visual SLAM,” in (Including Subsec: Lecture Notes Artificial Lecture Notes Bioinformatics), Lecture Notes in Computer Science) vol. 10345. Bournemouth, U.K.: Springer-Verlag, 2017, pp. 11–19. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-65849-0_2

[6] N. Mahmoud, O. G. Grasa, S. A. Nicolau, C. Doignon, L. Soler, J. Marescaux, and J. M. M. Montiel, “On-patient see-through augmented reality based on visual SLAM,” Int. J. Comput. Assist. Radiol. Surgery, vol. 12, no. 1, pp. 1–11, Jan. 2017. https://link.springer.com.libproxy1.nus.edu.sg/article/10.1007/s11548-016-1444-x

[7] W. Tan, H. Lui, Z. Dong, G. Zhang, and H. Bao, “Robust monocular SLAM in dynamic environments,” in Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR), vol. 12, no. 1, pp. 1–11, Jan. 2017. https://link.springer.com/chapter/10.1007/978-3-319-65849-0_2

[8] A. J. Davison, “Real-time simultaneous localisation and mapping with a single camera,” Ph.D. dissertation, Cambridge Univ., Cambridge, U.K., 2003.

[9] H. S. Maghdid, I. A. Lami, K. Z. Ghafoor, and J. Lloret, “ORB-SLAM: A versatile and accurate monocular SLAM system,” IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[10] H. S. Maghdid, I. A. Lami, K. Z. Ghafoor, and J. Lloret, “Dynamic scene semantics SLAM based on object detection and improved octomap,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Venice, Italy, Oct. 2017, pp. 2961–2969.

[11] X. Zhao, K. Yu, Z. Rong, S. Leutenegger, and T. Darrell, “Real-time visual-inertial localization using semantic segmentation towards dynamic environments,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Taipei, Taiwan, Oct. 2010, pp. 3975–3980.

[12] S. A. Maghdid, H. S. Maghdid, S. R. HmaSalah, K. Z. Ghafoor, A. S. Sadiq, and S. Khan, “Indoor human tracking mechanism using integrated onboard smartphones Wi-Fi device and inertial sensors,” Telecommun. Syst., vol. 71, no. 3, pp. 447–458, Jul. 2019. doi: 10.1007/s11235-018-0517-2.

[13] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-manifold preintegration for real-time visual-inertial odometry,” IEEE Trans. Robot., vol. 33, no. 1, pp. 1–19, Feb. 2017.
XINYANG ZHAO received the bachelor’s degree in measurement, control technology, and instrumentation from Harbin Engineering University, Harbin, China, in 2015, and the master’s degree in control engineering from the Harbin Institute of Technology, Harbin, in 2017. His current research interests include computer vision, simultaneous localization and mapping, robotics, and deep learning.

CHANGHONG WANG (Senior Member, IEEE) received the B.S. degree in automation, the M.E. degree in control science and engineering, and the Ph.D. degree in navigation guidance and control from the Harbin Institute of Technology (HIT), Harbin, China, in 1983, 1986, and 1991, respectively.

From 1993 to 1994, he was a Postdoctoral Fellow with the Department of Automation, Faculté Polytechnique de Mons, Mons, Belgium. In 1986, he joined the Department of Control Science and Control Engineering, HIT, where he is currently a Full Professor. He was the Director of control theory and application for five years and the Chairman of the Department of Control Science and Control Engineering for three years. He is also the Director of the Space Control and Inertial Technology Research Center and the Deputy Dean of the HIT (Anshan) Institute of Industrial Technology, Harbin Institute of Technology. He has authored more than 120 refereed papers published in technical journals, books, and conference proceedings. His current research interests include intelligent control and intelligent systems, inertial technology and its testing equipment, robotics, precision servo systems, and network control.

Dr. Wang was named as the Thomson Reuters Highly Cited Researcher, in 2014 and 2015.

MARCELO H. ANG, JR. (Senior Member, IEEE) received the B.Sc. degree (cum laude) in mechanical engineering and industrial management engineering from the De La Salle University Manila, Manila, Philippines, in 1981, the M.Sc. degree in mechanical engineering from the University of Hawai‘i at Mānoa, Honolulu, HI, USA, in 1985, and the second M.Sc. and Ph.D. degrees in electrical engineering from the University of Rochester, Rochester, NY, USA, in 1986 and 1988, respectively. His work experience includes heading the Technical Training Division of Intel’s Assembly and Test Facility, Philippines, research positions at the East West Center in Hawaii and at the Massachusetts Institute of Technology, and a faculty position as an assistant professor of electrical engineering at the University of Rochester. In 1989, he joined the Department of Mechanical Engineering, National University of Singapore, where he is currently an Associate Professor and an Acting Director of the Advanced Robotics Centre. His research interests include robotics, mechatronics, and applications of intelligent systems methodologies. He teaches the graduate and undergraduate levels in the following areas robotics, creativity and innovation, applied electronics and instrumentation, advanced computing, and product design and realization. He is also active in consulting work in these areas. In addition to academic and research activities, he is actively involved in the Singapore Robotic Games as its Founding Chairman and the World Robot Olympiad as a member of the Advisory Council.