Robotica (2024), **42**, pp. 3570–3591
doi:10.1017/S0263574724001553

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

A semantic visual SLAM based on improved mask R-CNN in dynamic environment

Kang Zhang[1,2], Chaoyi Dong[1,2,3], Hongfei Guo[4], Qifan Ye[1,2], Liangliang Gao[1,2], Shuai Xiang[1,2], Xiaoyan Chen[1,2,3] and Yi Wu[4]

[1]College of Electric Power, Inner Mongolia University of Technology, Hohhot, China
[2]Intelligent Energy Technology and Equipment Engineering Research Centre of Colleges and Universities in Inner Mongolia Autonomous Region, Hohhot, China
[3]Engineering Research Center of Large Energy Storage Technology, Ministry of Education, Hohhot, China
[4]Inner Mongolia Academy of Science and Technology, Hohhot, China
Corresponding authors: Chaoyi Dong; Email: dongchaoyi@imut.edu.cn; Hongfei Guo; Email: ghf-2005@163.com

Received: 5 January 2024; **Revised:** 9 June 2024; **Accepted:** 12 July 2024; **First published online:** 3 October 2024

Keywords: visual SLAM; dynamic environment; semantic segmentation; ORB-SLAM2; motion consistency detection

Abstract

To address the issues of low positioning accuracy and weak robustness of prior visual simultaneous localization and mapping (VSLAM) systems in dynamic environments, a semantic VSLAM (Sem-VSLAM) approach based on deep learning is proposed in this article. The proposed Sem-VSLAM algorithm adds semantic segmentation threads in parallel based on the open-source ORB-SLAM2's visual odometry. First, while extracting the ORB features from an RGB-D image, the frame image is semantically segmented, and the segmented results are detected and repaired. Then, the feature points of dynamic objects are eliminated by using semantic information and motion consistency detection, and the poses are estimated by using the remaining feature points after the dynamic feature elimination. Finally, a 3D point cloud map is constructed by using tracking information and semantic information. The experiment uses Technical University of Munich public data to show the usefulness of the Sem-VSLAM algorithm. The experimental results show that the Sem-VSLAM algorithm can reduce the absolute trajectory error and relative attitude error of attitude estimation by about 95% compared to the ORB-SLAM2 algorithm and by about 14% compared to the VO-YOLOv5s in a highly dynamic environment and the average time consumption of tracking each frame image reaches 61 ms. It is verified that the Sem-VSLAM algorithm effectively improves the robustness and positioning accuracy in high dynamic environment and owning a satisfying real-time performance. Therefore, the Sem-VSLAM has a better mapping effect in a highly dynamic environment.

1. Introduction

Simultaneous localization and mapping (SLAM) refers to a sort of technology, which a mobile robot can employ to estimate its own pose, and to build its surrounding environment map [1, 2]. The mobile robot obtains the surrounding environment information only through its own sensors and without any prior information. For decades, SLAM technology has been developed as a prerequisite for the navigation, guidance, and control of intelligent mobile robots [3, 4]. SLAM technology is currently extensively utilized in the areas of smart homes, autonomous mobile robots, and unmanned driving. Because of its low cost and capacity to gather extensive environmental information, camera-based visual SLAM (VSLAM) has emerged as an important area of robotic technology [5, 6].

At present, VSLAM-based algorithms have made great progress and many novel algorithms have emerged, such as ORB-SLAM2 (Orient FAST and Rotated BRIEF SLAM3) [7], ORB-SLAM3 [8], and large-scale direct monocular SLAM (LSD-SLAM) [9]. However, the existing algorithms often operate

under the strict presumption that the current environment is static or just low dynamic. Such requirements for the environment are very strict and do not meet the situation of ordinary dynamic scenes. However, in the real world, human walking, object moving, and repeatedly switching doors and windows are inevitable, so this assumption of static working environment seriously affects the practicability of those VSLAM systems in the real environment. This hypothesis will generate cumulative errors in the VSLAM system, which will seriously affect the robustness and accuracy of mobile robot localization. Therefore, the accuracy and robustness of VSLAM in a dynamic environment have grown to be extremely difficult and urgently need to be solved [10].

With deep learning's rapid growth and widespread application in recent years, the accuracy and robustness of VSLAM have been attracted many interests and widely concerned by scholars [11]. To increase the accuracy and robustness of the VSLAMs in dynamic environments, numerous deep learning-based VSLAM algorithms have been proposed. These schemes are mainly divided into two groups: the methods based on object detection and semantic segmentation or instance segmentation [12].

In the method of object detection, Riazuelo et al. [13] used object detection to identify people with high dynamic characteristics, and the dynamic feature points in the object detection frame are eliminated. Zhong et al. [14] used single shot multibox detector to identify moving objects and remove all feature points in surrounding regions. Ye et al. [15] used You Only Look Once version5 (YOLOv5) to eliminate the feature points in the dynamic object detection box and retain the static feature points for pose estimation. Since the object bounding box obtained by the object detection method cannot reach the pixel-level accuracy, Eliminating the dynamic feature points raises the danger of removing the static feature points from the bounding box. The mis-elimination may decrease the accuracy of posture estimation. Fang et al. [16] used another object detection (YOLOv4) to obtain the bounding box and used a maximum inter-class variance algorithm for depth images to segment the foreground in the bounding box. Liu et al. proposed a KMOP-SLAM algorithm based on unsupervised learning mode and manual detection in order to reduce tracking errors in dynamic environments [17]. Zheng et al. combined object detection with Bayesian filtering to propose a lightweight RLD-SLAM algorithm and used semantic and motion information to track dynamic objects [18]. Zhong et al. proposed a YOLO-SLAM algorithm to reduce the impact of dynamic targets on SLAM system by combining target detection with geometric constraints [19]. This method uses depth image and object detection to achieve pixel-level accuracy. However, the segmentation accuracy will decline as the dynamic object gets closer to its depth image's depth value.

For the method based on semantic segmentation or instance segmentation, DS-SLAM was proposed by Yu et al. [20], where a consistency check is performed using the optical flow tracking of interframe images. The feature points of dynamic were eliminated by combining SegNet to improve the accuracy of pose estimation and octrees were used to establish a map with semantic information [21]. In the DynaSLAM method put out by Bescos et al. [22], a multiview geometry and a Mask R-CNN [23] instance segmentation network were merged to detect dynamic objects, and a corresponding static map was used to complete the background repair. Fan et al. proposed a SLAM-PCD that can build high-precision point cloud maps by studying the introduced noise module [24]. Liu et al. proposed a real-time RDS-SLAM algorithm based on ORB-SLAM3, which improved tracking accuracy by adding semantics and semantically optimized threads and removing dynamic outliers [25]. Wu et al. proposed a DynaTM-SLAM algorithm, which can jointly optimize information such as camera attitude and map points [26]. These aforementioned semantic segmentation or instance segmentation methods do not use semantic information, but simply combine the two.

To solve the problem of low accuracy and poor robustness of VSLAM in dynamic environment, this article has improved ORB-SLAM2 and proposed a semantic VSLAM (Sem-VSLAM) algorithm based on deep learning, the major work is as follows:

To lessen the error of pose estimation, the Sem-VSLAM based on semantic segmentation and the motion consistency detection algorithm is proposed. The method eliminates the dynamic feature points of dynamic objects using geometric information and semantic segmentation information.

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

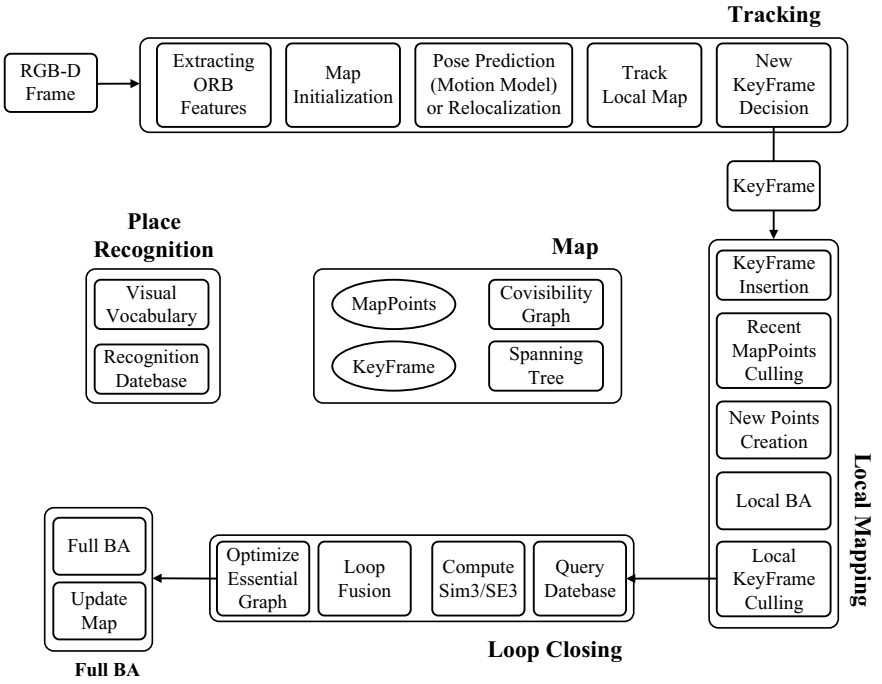

Figure 1. *ORB-SLAM2 system framework.*

By using semantic information in dynamic environment, dynamic object occlusion is detected and restored.

The experiment uses the Technical University of Munich (TUM) indoor dataset to test and analyze the Sem-VSLAM algorithm and compare it with ORB-SLAM2 and Visual Odometry You Only Look Once v5 small (VO-YOLOv5s).

2. Methods

2.1. ORB-SLAM2

The ORB-SLAM2 system was first proposed by Mur-Artal et al. [7], which is also a representative work using the front-end visual odometry feature point method. The system constructs three multi-thread frameworks that can run in real time. The framework has also been employed and improved by many scholars on the application of VSLAM systems. Figure 1 depicts the ORB-SLAM2 system framework.

The tracking performs ORB feature extraction on each input frame image and performs interframe feature point matching. The local mapping is responsible for managing the keyframe obtained. The common view relationship with other keyframe is determined according to the map point information contained in the keyframe. Thus, the unqualified map points and redundant keyframe can be eliminated. Finally, a series of keyframe groups with common view relationship and their observed map points in a sliding window are optimized by a local BA. The loop closing starts to work when it detects that the process of loop closing is about to occur. It uses loop closing constraints to fuse keyframe and optimize graphs to eliminate the cumulative error generated by the system during operation.

2.2. The framework of Sem-VSLAM

Based on the framework of ORB-SLAM2, the Sem-VSLAM improves the front-end visual odometry part by adding parallel threads of semantic segmentation, detection and restoration, adding motion

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

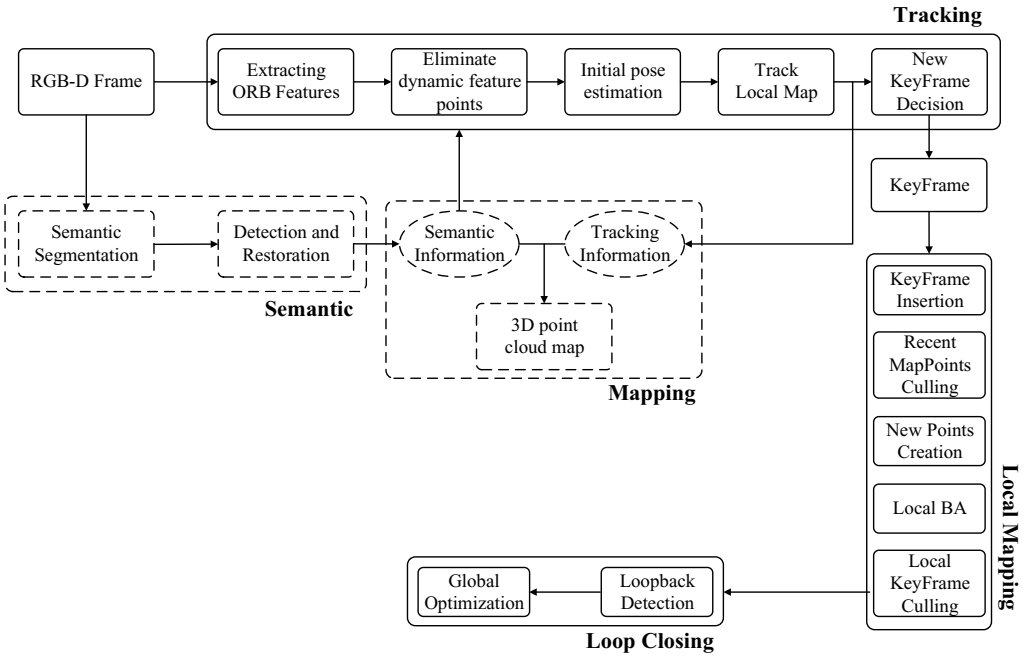

Figure 2. *The system framework of sem-VSLAM.*

consistency detection algorithm in the tracking thread, adding a 3D point cloud map thread. The purpose of such improvement is to detect the dynamic object to reduce the interference of dynamic feature points to the visual odometry. In Figure 2, the Sem-VSLAM system framework is displayed.

The improved system mainly includes four threads that can run in parallel in real time. When each RGB-D frame is input, it is transmitted to the tracking and the semantic, and the RGB-D image is processed in parallel. The Mask R-CNN [23] network is utilized by the semantic, which divides objects into static objects and dynamic objects. The function of detection and restoration starts to repair if a lack of segmentation is detected. Then, the pixel-level semantic labels of dynamic objects are provided to the tracking thread, and the geometric constraint of motion consistency is used to further detect outliers (abnormal values) of potential dynamic feature points. In this way, the feature points on the dynamic object are eliminated, and the remaining relatively stable static feature points are used to estimate the poses. Finally, a 3D point cloud map can be constructed by combining semantic information and tracking information.

2.3. Mask R-CNN

To solve the problem of low accuracy and poor robustness of VSLAM in dynamic environment, this article combines a deep learning method to detect dynamic objects. The dynamic objects are identified using a semantic segmentation network and pixel-level semantic segmentation of dynamic objects are obtained as semantic prior knowledge. The article uses the Mask R-CNN [23] network to perform the semantic segmentation. The network can not only obtain pixel-level semantic labels but also obtain instance labels. This article mainly uses pixel-level semantic tags to detect dynamic objects, while instance tags can also be useful in the future tracking different dynamic objects.

The Mask R-CNN is a framework that extends the Faster R-CNN [27] object detection framework to instance segmentation. The prior information of its motion properties can be further gained using the pixel-level semantic labels that the Mask R-CNN network has obtained. For example, if the pixel-level label is “human,” then it is assumed that the pixels are a dynamic object with high confidence, because according to people’s common sense, people tend to move; if the pixel-level label is “desk,” then it is

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

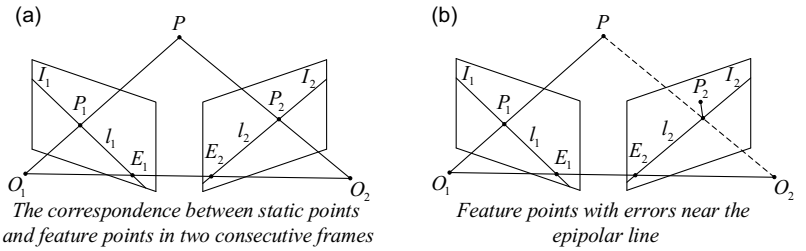

Figure 3. *An illustration of multi-view epipolar geometric constraints.*

assumed that the pixels are a static object; if the pixel-level label is a "chair," then it cannot be assumed that the pixel is a static object with high confidence. Because the chair itself cannot move, but there may be moving behavior under human activity, the pixel of this object is defined as a potentially dynamic object.

To perform semantic segmentation in the indoor environment, this article uses the Mask R-CNN under the TensorFlow framework. The pretraining model of the network is fully trained through the MS COCO dataset [28], which has a high recognition and segmentation effect. The MS COCO dataset has a total of more than 80 different categories of objects. One is moving objects with high dynamic confidence, and 18 object categories in the MS COCO dataset are selected (e.g., people, bicycles, cars, motorcycles, aircraft cats, birds, and dogs); the other is the potential dynamic objects that move as people move, mainly selecting three object categories in the MS COCO dataset (e.g., chairs, books, and cups). If there are special requirements for the new categories, the network can also be retrained. The trained neural network is trained using the TUM in Germany, which can avoid spending a lot of time of collecting image data, and the plenty of data can also guarantee to obtain a robust result.

2.4. Motion consistency detection

Most dynamic objects can be segmented using the Mask R-CNN network utilized in this article, but the segmentation effect for potential dynamic objects that may move is not satisfying for the realistic applications. For example, books carried by people and chairs moved with people. To overcome this difficulty, using multi-view epipolar geometric constraints to further detect whether the features of objects are dynamic features. If the features of objects are dynamic features, they cannot meet the multi-view epipolar geometric constraints. If they are static features, they meet the multi-view epipolar geometric constraints.

Figure 3(a) depicts the relationship between two successive frame image points, where $\boldsymbol{P}$ is a point in space, $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are obtained from two consecutive frame images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, respectively. The baseline is defined as $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$. The epipolar plane is a plane π that is determined by the space point $\boldsymbol{P}$ and the baseline. The intersection lines $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$ of plane $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ with plane π are called polar lines. The intersection points $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ of the image plane $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ with the baseline are called the poles. The homogeneous coordinates of feature points $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ can be expressed as

$$\begin{aligned} \boldsymbol{P}_1 &= [\boldsymbol{u}_1, \boldsymbol{v}_1, 1]^{\mathrm{T}}, \\ \boldsymbol{P}_2 &= [\boldsymbol{u}_2, \boldsymbol{v}_2, 1]^{\mathrm{T}}, \end{aligned} \tag{1}$$

where $\boldsymbol{u}_i, \boldsymbol{v}_i (\boldsymbol{i} = 1, 2)$ are coordinates of $\boldsymbol{P}_i (\boldsymbol{i} = 1, 2)$, respectively. The polar line $\boldsymbol{l}_1$ is

$$\boldsymbol{l}_1 = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{bmatrix} = \boldsymbol{F}\boldsymbol{P}_1 = \boldsymbol{F} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{v}_1 \\ 1 \end{bmatrix}, \tag{2}$$

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

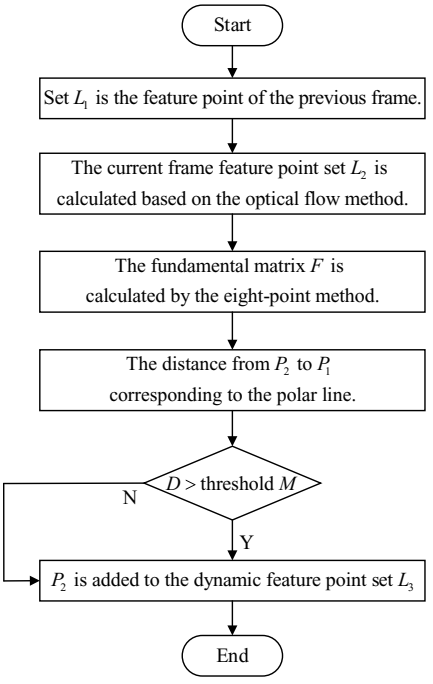

Figure 4. *The flowchart of the motion consistency detection algorithm.*

where $\boldsymbol{F}$ is a fundamental matrix, which transform the features point to the polar line. The mapping relationship can be expressed as

$$\boldsymbol{P}_2^{\mathrm{T}}\boldsymbol{F}\boldsymbol{P}_1 = 0. \tag{3}$$

Under the premise that point $\boldsymbol{P}_1$ in $\boldsymbol{I}_1$ and the fundamental matrix $\boldsymbol{F}$ are known, if $\boldsymbol{P}$ is a static point, it must satisfy the constraint of (3). However, due to the uncertainty of feature extraction and fundamental matrix $\boldsymbol{F}$ estimation, the feature points located near the epipolar line usually have errors, resulting in two image points of spatial point mapping not satisfying the constraints of (3). Feature point $\boldsymbol{P}_2$ is quite close to epipolar line $\boldsymbol{l}_2$, as shown in Figure 3(b). Therefore, judgment criteria need to be added to evaluate the degree of the mismatch cases. A distance $\boldsymbol{D}$ from $\boldsymbol{P}_2$ to $\boldsymbol{l}_2$ can be determined as

$$\boldsymbol{D} = \frac{\left|\boldsymbol{P}_2^{\mathrm{T}}\boldsymbol{F}\boldsymbol{P}_1\right|}{\sqrt{\|\boldsymbol{X}\|^2 + \|\boldsymbol{Y}\|^2}}. \tag{4}$$

By calculating (4), if $\boldsymbol{D}$ is less than a preset threshold, point $\boldsymbol{P}$ is viewed to be static; otherwise, point $\boldsymbol{P}$ is viewed to be dynamic.

Figure 4 depicts the flow chart for the method used to detect motion consistency. First, according to the previous frame feature point set $\boldsymbol{L}_1$, the feature point set $\boldsymbol{L}_2$in the current frame is determined using the optical flow method. Then the fundamental matrix $\boldsymbol{F}$ is estimated using at least 5 pairs of feature matching pairings, where the classical eight-point method is usually used. Finally, the relationship between the distance of the corresponding epipolar line from $\boldsymbol{P}_2$ to $\boldsymbol{P}_1$ and the preset threshold is evaluated to ascertain whether the feature point is moving or not. If it moves, it is a dynamic feature, otherwise it is a static feature.

2.5. Detection and restoration

In this article, besides introducing Mask R-CNN to ORB-SLAM2, and the network is also lightweighted to ensure meeting real-time requirements. However, in the lightweight processing of this network, the

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

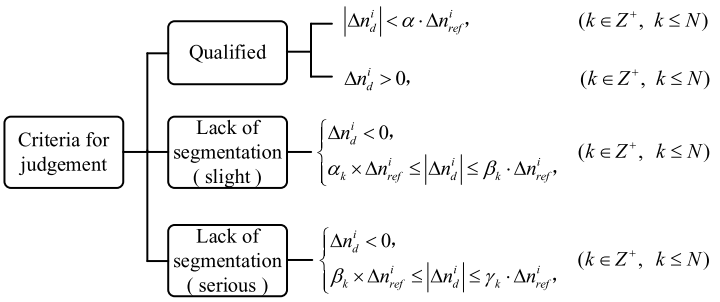

Figure 5. *Judgment criteria of segmentation result states.*

network emerges unstable segmentation quality, especially when the human limbs in the image are not displayed completely or the image is blurred, the segmentation quality is seriously reduced, which can result in a decline in the accuracy of pose estimation and mapping in the follow-up process. This article provides a corresponding measure to repair the segmentation results to address this issue. Each segmentation result is qualified based on the assumption that the camera has a high frame rate. If it is judged to be qualified, it is used directly; if it is judged to be unqualified, corresponding measures are taken to repair it and then use it again.

In the indoor environment, people are usually considered as dynamic objects with high dynamic characteristics. Therefore, this article mainly detects the segmentation results with lackness of segmentation for human. Let $\boldsymbol{W_i}$ be the current segmentation result and $\boldsymbol{n_d}(\boldsymbol{W_i})$ be the number of points in the current segmentation result. To better reflect the $\boldsymbol{n_d}(\boldsymbol{W_i})$ close to the real change in a short time, the images that meet the segmentation qualification requirement are selected, and the serial number of the segmented image is close to the current segmented image as much as possible. The change of the people number $\boldsymbol{W_{i-k}}$ can be denoted as $\Delta \boldsymbol{n_d^i} = \boldsymbol{n_d}(\boldsymbol{W_i}) - \boldsymbol{n_d}(\boldsymbol{W_{i-k}})$. If $\Delta \boldsymbol{n_d^i} \ll 0$, it may be due to the lack of segmentation of $\boldsymbol{W_i}$, which is probably caused by the excessive speed of human movement or the excessive angular velocity of human from the front to the side. In both of the two cases (the excessive speed and the excessive angular velocity), the probability of $\boldsymbol{W_i}$ missing segmentation is even higher than the single case. Thus, to increase the accuracy with which the absence of segmentation is detected, $\Delta \boldsymbol{n_d^i} \ll 0$ is judged as a sign of lack of segmentation.

Considering that the size of $\Delta \boldsymbol{n_d^i}$ requires a reference value to measure, this reference value requires to be able to change with the speed of the camera's movement and also have the characteristics of being less affected by noise such as over-segmentation and lack of segmentation. Therefore, a new concept dynamic reference variation $\Delta \boldsymbol{n_{ref}^i}$ is introduced to reflect the size of $\Delta \boldsymbol{n_d^i}$, and an array $\boldsymbol{Q}$ is used to store a specific $|\Delta \boldsymbol{n_d^i}|$ to calculate $\Delta \boldsymbol{n_{ref}^{i+1}}$. The initial value of $\Delta \boldsymbol{n_{ref}^{i+1}}$ is set to 500.

According to the size of $\Delta \boldsymbol{n_d^i}$ relative to $\Delta \boldsymbol{n_{ref}^i}$, the states of the segmentation result are divided into three categories: qualified, lack of segmentation (slight), and lack of segmentation (serious). As shown in Figure 5, three criteria are listed for judging the states of the segmentation results. Here, $\Delta \boldsymbol{n_{ref}^i}$ is the reference standard for the current segmentation results, $\boldsymbol{N}$ is a hyperparameter (set to 5), and α_k, β_k, and γ_k are preset thresholds and satisfy the relation: $\alpha_k < \beta_k < \gamma_k$.

After obtaining the state of $\boldsymbol{W_i}$, if $\boldsymbol{W_i}$ is qualified and satisfies $0.05 \times \Delta \boldsymbol{n_{ref}^i} \le |\Delta \boldsymbol{n_d^i}| \le \alpha \cdot \Delta \boldsymbol{n_{ref}^i}$, the reference variation is updated, otherwise not updated and at the same time $\Delta \boldsymbol{n_{ref}^{i+1}} = \Delta \boldsymbol{n_{ref}^i}$. The update of the reference variation is as follows:

$$\Delta \boldsymbol{n_{ref}^{i+1}} = \begin{cases} \dfrac{\boldsymbol{L_i} \cdot \Delta \boldsymbol{n_{ref}^i} + \left|\Delta \boldsymbol{n_d^i}\right|}{\boldsymbol{L_i} + 1}, & \boldsymbol{L_i} < \boldsymbol{L}_{\max} \\ \dfrac{\boldsymbol{L}_{\max} \cdot \Delta \boldsymbol{n_{ref}^i} - \boldsymbol{q_f} + \left|\Delta \boldsymbol{n_d^i}\right|}{\boldsymbol{L}_{\max}}, & \boldsymbol{L_i} = \boldsymbol{L}_{\max} \end{cases}, \tag{5}$$

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

Figure 6. *TUM datasets in different environments.*

where $\boldsymbol{q_f}$ is the starting element of array $\boldsymbol{Q}$ when generating $\boldsymbol{W_i}, \boldsymbol{L_i}$ is the $\boldsymbol{i}$th element of array $\boldsymbol{Q}$ when generating $\boldsymbol{W_i}, \boldsymbol{L}_{\max}$ is a hyperparameter representing the maximum length of $\boldsymbol{Q}$, and $\boldsymbol{L}_{\max}$ is set to 20. After updating the reference variable, $|\Delta \boldsymbol{n}_{\boldsymbol{d}}^{\boldsymbol{i}}|$ is inserted into array $\boldsymbol{Q}$; if $\boldsymbol{Q} > \boldsymbol{L}_{\max}$, the starting element need to be discarded.

After the calculation of $\Delta \boldsymbol{n}_{\boldsymbol{ref}}^{\boldsymbol{i+1}}$, the results lack of segmentation (slight and serious) can be repaired. Because the difference of dynamic objects in an image does not change much in a short time, the repair method adopted is to superimpose the occlusion and mask of the dynamic object by the segmentation result, which is the largest number of human points in the first $\boldsymbol{N}$ segmentation results of the current frame, on the repaired object.

3. Experiments

This article mainly uses the TUM dataset to evaluate the positioning accuracy and robustness of the algorithm. Because ORB-SLAM2 is excellent in a static environment, this article compares the Sem-VSLAM with the ORB-SLAM2 to evaluate the improvement effect of the Sem-VSLAM in a dynamic environment. The dataset was collected using a Kinect camera at a rate of 30 Hz, and the collected data image resolution was 640 × 480. This dataset has been also adopted by most scholars who study VSLAM.

In this article, the accuracies of VSLAM algorithms are evaluated by the evaluation indices proposed in reference [29], which are absolute trajectory error (ATE) and relative pose error (RPE). ATE represents the difference between the estimated pose and the true pose, which directly reflects the accuracies of estimated trajectories. RPE represents the difference between the estimated pose and the true pose in a fixed time interval. The parameter used in this article for evaluating these two indices is called root mean squared error (RMSE). The calculation formula is

$$\boldsymbol{RMSE}\left(\boldsymbol{E}_{1,n}, \Delta\right) = \left[\frac{1}{\boldsymbol{m}} \sum_{i=1}^{m} \|\boldsymbol{trans}\left(\boldsymbol{E}_i\right)\|^2\right]^{\frac{1}{2}} \tag{6}$$

where $\boldsymbol{trans}(\boldsymbol{E}_i)$ represents the pose estimation error of absolute and relative at the $\boldsymbol{i}$th moment, $\boldsymbol{E}_{1,n}$ represents the camera pose from the first time point to the $\boldsymbol{n}$th, and Δ is the fixed time interval.

The walking sequence of the TUM dataset expresses the information that two people sit on a chair and gradually get up to walk around the desk and then sit down. At the same time, the camera also moves in a preset motion mode. At part time nodes, most of the image area is taken up by people strolling, which is very challenging for the performance evaluation of VSLAM algorithm in dynamic environment. Therefore, this type of dataset can be used as a high dynamic environment dataset. Furthermore, the sitting sequence refers to the small swing of people sitting on the chair. The desk sequence refers to the movement of the camera around the stationary desk. The TUM datasets in the three different environments are shown in Figure 6. Besides in the high dynamic environment, the effectiveness of the Sem-VSLAM was also evaluated in this article in low dynamic environment and static environment.

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

Table I. *Training and experimental computer parameters.*

Name	Model number
Operating system	Ubuntu18.04
CPU	Intel(R) Core (TM) i9-10920X CPU @ 3.50GHz
GPU	NVIDIA GeForce RTX 3080, 10GB
Computer memory	32GB
CUDA version	CUDA 11.2
Python version	Python 3.8

Table II. *The detection performance of the backbone-changed mask R-CNN networks and YOLOv5s network.*

Different network	FLOPs/G	Accuracy /%	Speed /FPS
Mask R-CNN-ResNet-50	3.8	76.24	145
Mask R-CNN-ResNet-101	7.6	78.53	137
Mask R-CNN- MobileNetV3	2.3	75.62	147
Mask R-CNN- GhostNet	8.3	78.23	134
Mask R-CNN-SegNet	7.5	77.36	136
Mask R-CNN-PSPNet	5.6	76.81	140
YOLOv5s	2.3	76.94	145
Improved mask R-CNN	2.1	77.61	151

The fr3_walking_xyz, fr3_walking_rpy, fr3_walking_halfsphere, and fr3_walking_static datasets are used as high dynamic environment datasets, and the Sem-VSLAM algorithm is used to test the accuracy. In order to compare the two algorithms' performance in the low dynamic environment and the static environment, this article also uses the dataset fr3_sitting_xyz, fr3_sitting_rpy, fr3_sitting_halfsphere, and fr3_sitting_static as the low dynamic dataset and the dataset fr2_desk as the static dataset. This article also compares the Sem-VSLAM algorithm with other VSLAM algorithms, for example, ORB-SLAM3 [7], VO-YOLOv5s [15], KMOP-SLAM [17], RLD-SLAM [18], YOLO-SLAM [19], DS-SLAM [20], DynaSLAM [22], SLAM-PCD [24], RDS-SLAM [25], DynaTM-SLAM [26], Static Fusion [30], DRSO-SLAM [31], RDMO-SLAM [32],CRF-SLAM [33], Amos-SLAM [34], DN-SLAM [35], DOR-SLAM [36], and PR-SLAM [37], for dynamic environments, and analyzes its advantages and disadvantages in accuracy and real-time performance.

3.1. Semantic segmentation mask experiment test

In this article, the improved Mask R-CNN semantic segmentation network is trained, and the semantic segmentation mask experiment is carried out. Table I lists the parameters of the computer used for training and experiments.

In this article, ResNet50/101 model in the traditional Mask R-CNN network is replaced with five different network models, that is, Backbone network is changed, and Backbone's network is tested separately. The following seven types of networks were tested and evaluated in terms of FLOPs/G, accuracy, and speed. The Mask R-CNN network performance test combined with different network models is shown in Table II. The improved Mask R-CNN network refers to Mask R-CNN-ModlieNetV2. The improved Mask R-CNN network performance has the smallest FLOPs, the fastest detection speed, and good accuracy, which can basically meet the requirements of the subsequent combination with SLAM system in this paper.

The improved Mask R-CNN network is trained with two datasets. The first training is conducted with MS COCO dataset, and then the second training is conducted with TUM dataset. The purpose of

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

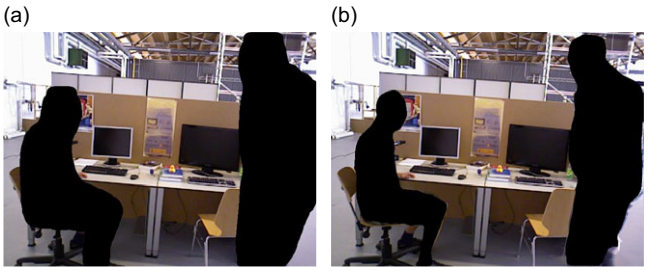

Figure 7. *Dynamic feature point elimination process.*

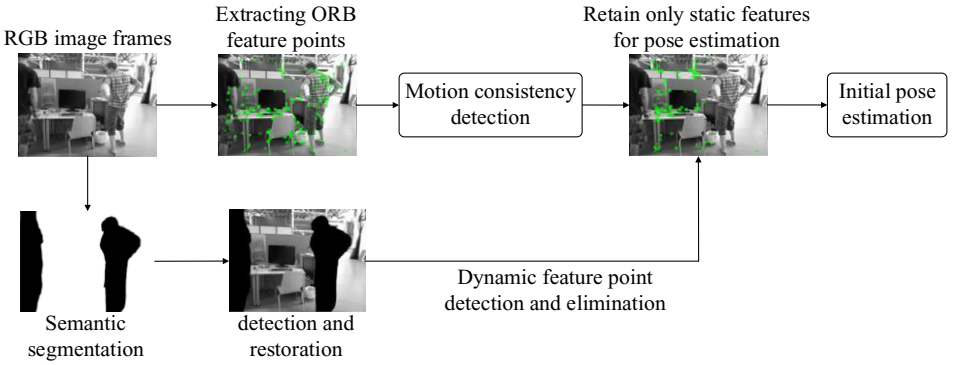

Figure 8. *Dynamic feature point elimination process.*

conducting two trainings is to achieve more accurate segmentation accuracy in the future. In order to meet the real-time needs, only the segmented dynamic objects are set as moving people. The training time has been increased from the original 212 ms per frame image processing to 51 ms, an improvement of 75.94%. The comparison of Mask experimental effects between Mask R-CNN network and the improved Mask R-CNN network is shown in Figure 7. It can be clearly seen from the figure that the segmentation effect of the improved Mask R-CNN network is better and the segmentation accuracy is higher than that of the mask R-CNN network.

3.2. Dynamic feature point elimination process

In the dynamic environments, this article uses the Mask R-CNN network to semantically segment the dynamic object and detect and repair the dynamic object that lacks segmentation. Combined with the motion consistency detection, the accuracy of eliminating dynamic object feature points is further verified to avoid erroneously eliminating static feature points. Figure 8 depicts the dynamic feature point elimination process. In the subsequent local map construction and camera pose estimation, only the feature points retained after removing the dynamic object feature points are used to enhance the accuracy and robustness of the Sem-VSLAM algorithm in the dynamic environment.

Figure 9(a) is the classic ORB-SLAM2 effect diagram without eliminating dynamic feature points. Obviously, many feature points stay on the human body. Therefore, the feature points distributed on static objects will be reduced accordingly, resulting in insufficient extraction of feature points on static objects. Figure 9(b) is to add semantic segmentation threads and combine the effect comparison diagram of motion consistency detection and dynamic object detection and restoration methods. It is obvious that a variety of feature points that were scattered throughout the human body have been eliminated and replaced with static feature points. This increases the quantity of feature points retrieved from

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

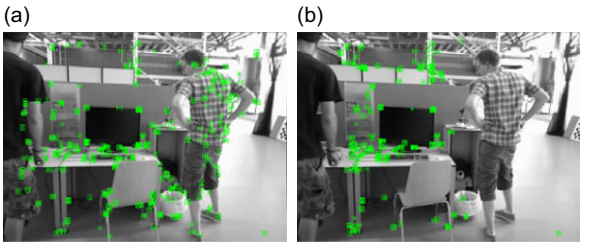

Figure 9. *Comparison of dynamic feature point elimination effects (TUM dataset).*

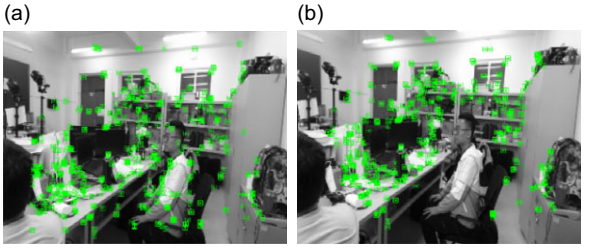

Figure 10. *Comparison of dynamic feature point elimination effects (laboratory environment dataset).*

static objects, enhances the accuracy of the entire system, and lays a foundation for the accuracy of the final map construction. The dynamic feature point elimination experiment is also carried out in the dynamic environment of the laboratory in this article. Figure 10 illustrates a comparison of the impact of eliminating dynamic feature points.

3.3. The experiment on the TUM dataset

In this article, the proposed Sem-VSLAM algorithm and ORB-SLAM2 algorithm are used to the experiments on 9 sub-datasets from the TUM dataset. Tables III, IV, and V display quantitative comparisons. In this experiment, the discrepancy between the estimated value and the true value of the camera pose is assessed using ATE and RPE, and RMSE and standard deviation (SD) of these differences are calculated. The RMSE describes the error between the estimated value and the true value, which is easily affected by the maximum, minimum, and accidental errors. The SD describes the degree of dispersion between the estimated value and the true value, which can better reflect the stability of the system. An improvement value is used in the Table III-V to evaluate the quality between the proposed Sem-VSLAM algorithm and the ORB-SLAM2 algorithm. The improvement value can be calculated as

$$\gamma = \frac{\alpha - \beta}{\alpha} \times 100\%, \tag{7}$$

where γ is the degree of improvement of the Sem-VSLAM algorithm, β is the camera trajectory error of the Sem-VSLAM algorithm, and α is the camera trajectory error of the classic ORB-SLAM2 algorithm.

As can be seen from Table I, in a high dynamic environment, the RMSE and SD of the absolute trajectory error of the Sem-VSLAM algorithm compared with the ORB-SLAM2 algorithm decreased by 96.39% and 95.03% on average, which fully illustrates that in a high dynamic environment. Under the environment, the Sem-VSLAM algorithm can significantly reduce the error of pose estimation. However, in a low dynamic environment, compared with ORB-SLAM2, the RMSE and SD of the absolute trajectory error of the Sem-VSLAM algorithm decreased by 33.43% and 33.22% on average. In a static environment, compared with ORB-SLAM2, the RMSE and SD of the absolute trajectory error of the

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

Table III. *RMSE and SD comparison of ATE.*

		ORB-SLAM2/m		VO-YOLOv5s /m		Sem-VSLAM algorithm/m		Improvement (1)/%		Improvement (2)/%	
Environment	**Dataset**	**RMSE**	**SD**	**RMSE**	**SD**	**RMSE**	**SD**	**RMSE**	**SD**	**RMSE**	**SD**
High dynamic	fr3_w_xyz	0.7985	0.4267	0.0167	0.0086	0.0152	0.0073	98.10	98.29	8.98	15.12
	fr3_w_rpy	0.5965	0.2103	0.0455	0.0321	0.0405	0.0262	93.21	87.54	10.99	18.38
	fr3_w_half	0.6596	0.3241	0.0317	0.0169	0.0296	0.0143	95.51	95.59	6.62	15.38
	fr3_w_static	0.5200	0.2281	0.0086	0.0045	0.0066	0.0030	98.73	98.68	23.26	33.33
Low dynamic	fr3_s_xyz	0.0157	0.0078	0.0121	0.0058	0.0108	0.0052	31.21	33.33	10.74	10.34
	fr3_s_rpy	0.0335	0.0194	0.0245	0.0153	0.0208	0.0124	37.91	36.08	15.10	18.95
	fr3_s_half	0.0291	0.0151	0.0264	0.0139	0.0188	0.0092	35.40	39.07	28.79	33.81
	fr3_s_static	0.0089	0.0041	0.0066	0.0033	0.0063	0.0031	29.21	24.39	4.55	6.06
Static	fr1_xyz	0.0083	0.0030	0.0096	0.0053	0.0079	0.0028	4.82	6.67	6.25	7.55

Table IV. *RMSE and SD comparison of RPE.*

		ORB-SLAM2/m		VO-YOLOv5s /m		Sem-VSLAM algorithm/m		Improvement (1)/%		Improvement (2)/%	
Environment	**Dataset**	**RMSE**	**SD**	**RMSE**	**SD**	**RMSE**	**SD**	**RMSE**	**SD**	**RMSE**	**SD**
High dynamic	fr3_w_xyz	0.4185	0.2803	0.0221	0.0113	0.0190	0.0091	95.46	96.75	14.03	19.47
	fr3_w_rpy	0.4038	0.2608	0.0668	0.0479	0.0573	0.0379	85.81	85.47	14.22	20.88
	fr3_w_half	0.4113	0.3122	0.0355	0.0194	0.0263	0.0124	93.61	96.03	25.92	36.08
	fr3_w_static	0.2249	0.1934	0.0109	0.0062	0.0088	0.0041	96.09	97.88	19.27	33.87
Low dynamic	fr3_s_xyz	0.0179	0.0097	0.0136	0.0066	0.0121	0.0062	32.40	36.08	11.03	6.06
	fr3_s_rpy	0.0435	0.0327	0.0315	0.0149	0.0283	0.0132	34.94	25.69	10.16	11.41
	fr3_s_half	0.0364	0.0218	0.0253	0.0182	0.0230	0.0141	36.81	35.32	9.09	22.52
	fr3_s_static	0.0094	0.0048	0.0081	0.0037	0.0075	0.0036	20.21	25.00	7.41	2.70
Static	fr1_xyz	0.0113	0.0047	0.0158	0.0073	0.0109	0.0046	3.54	2.13	2.51	2.74

Table V. *RMSE and SD comparison of relative rotation errors.*

		ORB-SLAM2/m		VO-YOLOv5s /m		Sem-VSLAM algorithm/m		Improvement (1)/%		Improvement (2)/%	
Environment	**Dataset**	**RMSE**	**SD**	**RMSE**	**SD**	**RMSE**	**SD**	**RMSE**	**SD**	**RMSE**	**SD**
High dynamic	fr3_w_xyz	8.0639	5.4594	0.6279	0.3799	0.6060	0.3750	92.49	93.13	3.5	1.3
	fr3_w_rpy	6.7831	5.0043	1.3437	0.9202	1.2023	0.7816	82.28	84.38	10.52	15.06
	fr3_w_half	8.8143	6.5846	0.8118	0.4177	0.7360	0.3575	91.65	94.57	9.34	13.16
	fr3_w_static	3.9432	3.3472	0.2711	0.1255	0.2428	0.1059	93.84	96.84	10.44	15.62
Low dynamic	fr3_s_xyz	0.5577	0.3211	0.5106	0.2785	0.4839	0.2605	13.23	18.87	5.23	6.46
	fr3_s_rpy	0.9142	0.5257	0.8462	0.4671	0.8053	0.4359	10.70	17.09	4.83	6.68
	fr3_s_half	0.7029	0.3347	0.6561	0.3028	0.5874	0.2801	16.43	16.31	10.47	7.50
	fr3_s_static	0.2822	0.1289	0.2728	0.1185	0.2757	0.1234	2.30	4.27	2.30	4.27
Static	fr1_xyz	0.4639	0.2214	0.9961	0.4825	0.9943	0.4819	0.39	1.22	0.03	0.12

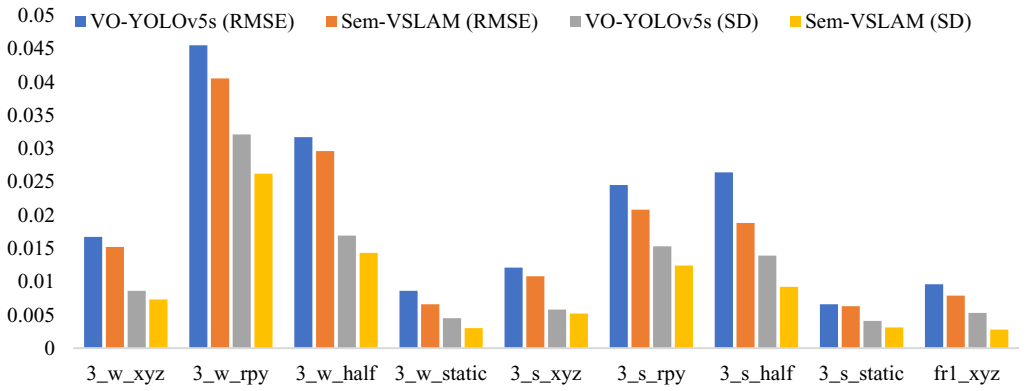

Figure 11. *Comparison of RMSE and SD data in ATE (VO-YOLOv5s).*

Sem-VSLAM algorithm decreased by 4.82% and 6.67% on average. The reason may be that the classical ORB-SLAM2 system uses the RANdom Sample Consensus algorithm. The algorithm itself can identify the dynamic feature points of small movements and eliminate them as noise. To a certain extent, ORB-SLAM2 is able to reduce the impact of small moving objects on the accuracy of the algorithm.

Table II and Table III list the RMSE and SD data of the relative trajectory error in a highly dynamic environment and calculate the improvement of the Sem-VSLAM algorithm. In the high dynamic environment, the RMSE and SD in the relative displacement error decreased by 96.39% and 94.03% on average, and the RMSE and SD in the relative rotation error decreased by an average of 90.07% and 92.23% in the high dynamic environment; in the low dynamic environment, the RMSE and SD in the relative displacement error decreased by 31.09% and 30.52% on average, and the RMSE and SD in the relative rotation error decreased by 10.67% and 14.14% on average; in a static environment, the RMSE and SD in the relative displacement error decreased on average by 9.42% and 6.23%, the RMSE and SD in the relative rotation error decreased by 5.71% and 6.23% on average. These two have the same effect as the ATE data in Table I. Therefore, from Table II and Table III, it can also be concluded that in a highly dynamic environment, the accuracy and performance of the Sem-VSLAM algorithm are greatly improved and more robust conclusions.

This article also compares Sem-VSLAM algorithm with VO-YOLOv5s algorithm improved by combining YOLOv5s in dynamic environment, as shown in Table I, II and III. As can be seen from Table I, RMSE and SD in absolute error decreased by 12.46% and 20.55% on average under high dynamic environment. In the low dynamic average decreased by 14.80% and 17.29%, the average decline in static conditions was 6.25% and 7.55%. As can be seen from Table II, RMSE and SD in relative pose errors in high dynamic environment decrease by 19.36% and 27.58% on average. In the low dynamic, the average decreased by 9.42% and 6.23%; And the average decrease in static environment is less than 2.51% and 2.74%. As can be seen from Table III, RMSE and SD in relative rotation errors decreased by 8.45% and 11.29% on average under high dynamic environment. In the low dynamic average decreased by 5.71% and 6.23%, the average decrease in static environment is less than 0.03% and 0.12%. By comparing VO-YOLOv5s algorithm in dynamic environment, it can be seen that Sem-VSLAM algorithm also verifies the conclusion that localization accuracy and robustness are better.

In order to compare and display the data in Tables III, IV and V more intuitively, absolute trajectory error data comparison histograms, relative position error data comparison histograms, and relative rotation error data comparison histograms are also plotted, as shown in Figures 11, 12 and 13, respectively. From the figures, it is more intuitive to show that the Sem-VSLAM algorithm is smaller in RMSE and SD data, that is, the various errors of its own positional estimation are smaller, which can better improve the positioning accuracy and construct more accurate 3D point cloud maps in dynamic environments.

In this article, the trajectory comparison of Sem-VSLAM algorithm with ORB-SLAM2 and VO-YOLOv5s algorithm in dynamic environments is also plotted based on three datasets in highly dynamic

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

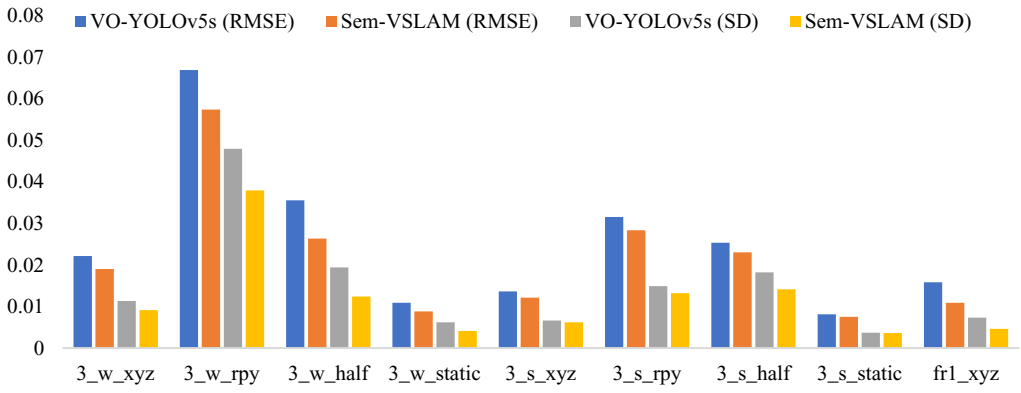

Figure 12. *Comparison of RMSE and SD data in RPE (VO-YOLOv5s).*

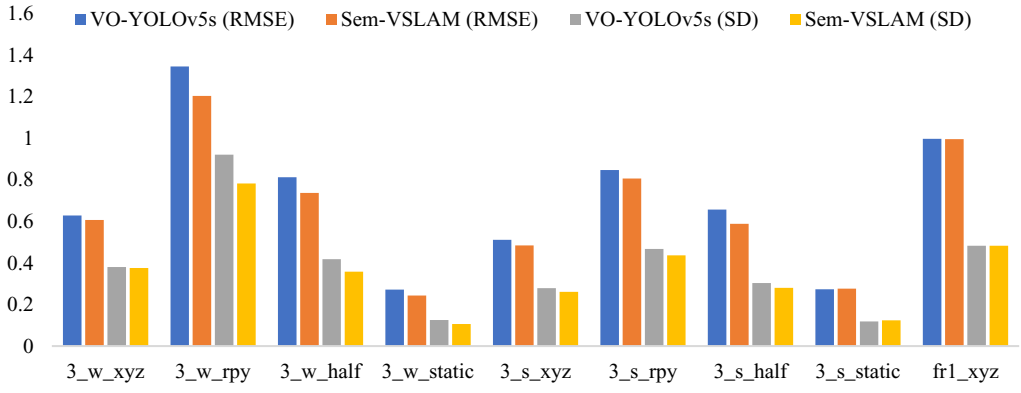

Figure 13. *Comparison of RMSE and SD data in relative rotation error (VO-YOLOv5s).*

environments, respectively. The left image is the ATE trajectory map, ground truth represents the real trajectory of the camera, estimated represents the estimated trajectory of the camera, and difference represents the error between the two. As shown in Figures 14, 15, 16 and 17, it can be more visualized that the algorithm in this article compared to ORB-SLAM2 and VO-YOLOv5s algorithm in dynamic environments, the absolute trajectory error and relative trajectory error of the proposed Sem-VSLAM algorithm in the high dynamic environment can be greatly reduced, which makes the pose estimation more accurate, the robustness of the system better, and the mapping effect better.

This article also compares other visual SLAM algorithms based on dynamic environment, which can further verify the effectiveness of the proposed algorithm. The experiment mainly compares the RMSE data of each algorithm ATE, and the experimental data are all from the original papers of each algorithm. As shown in Table VI, the positioning accuracy of Sem-VSLAM is much higher than that of ORB-SLAM3 and other algorithms and can achieve similar positioning accuracy of visual SLAM algorithms in other dynamic environments. Moreover, under the fr3_walking_static and fr3_sitting_static datasets, the Sem-VSLAM algorithm proposed in this article has higher positioning accuracy and better performance.

In order to more intuitively compare and display the data with similar accuracy to Sem-VSLAM algorithm in Table VI, RMSE comparison diagram of partial visual SLAM algorithm ATE in dynamic environment was also drawn, as shown in Figure 18. From the figure, it can be seen more intuitively that the Sem-VSLAM algorithm has better performance than VO-YOLOv5s, RLD-SLAM, Dyna SLAM, SLAM-PCD, fr3_walking_xyz, fr3_walking_static and fr3_sitting_static in the datasets

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

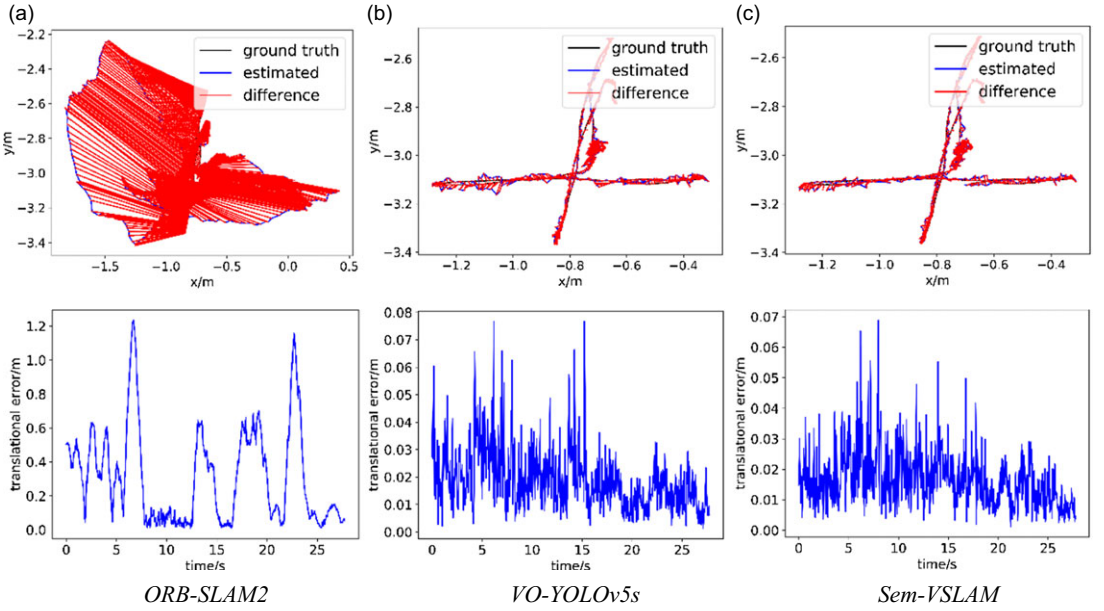

Figure 14. *ATE and RPE of the three algorithms on fr3_ walking _xyz dataset.*

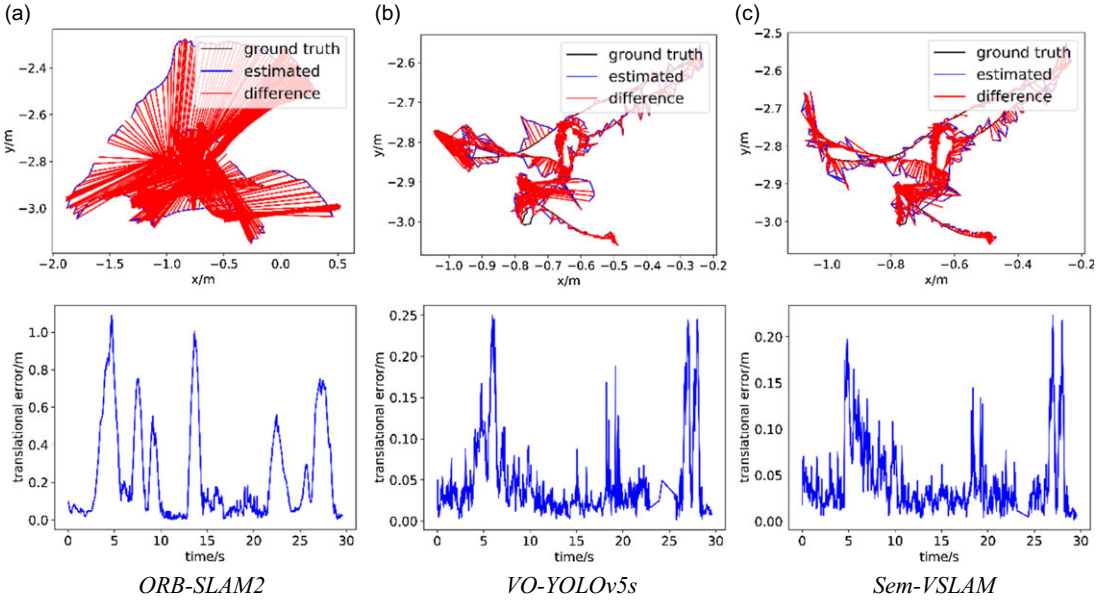

Figure 15. *ATE and RPE of the three algorithms on fr3_ walking _rpy dataset.*

FR3_walking_static. CRF-SLAM and PR-SLAM algorithms have smaller RMSE and similar positioning accuracy to YOLO-SLAM, DynaTM-SLAM, Amos-SLAM, and DN-SLAM algorithms. Therefore, Sem-VSLAM algorithm has smaller self-position estimation errors on datasets fr3_walking_xyz, fr3_walking_static, and fr3_sitting_static, which can better improve positioning accuracy and build more accurate three-dimensional point cloud images in dynamic environments.

In practical applications, the real-time performance of the algorithm is also one of the important indicators for evaluating a SLAM system. Such an experiment mainly compares the average time required

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

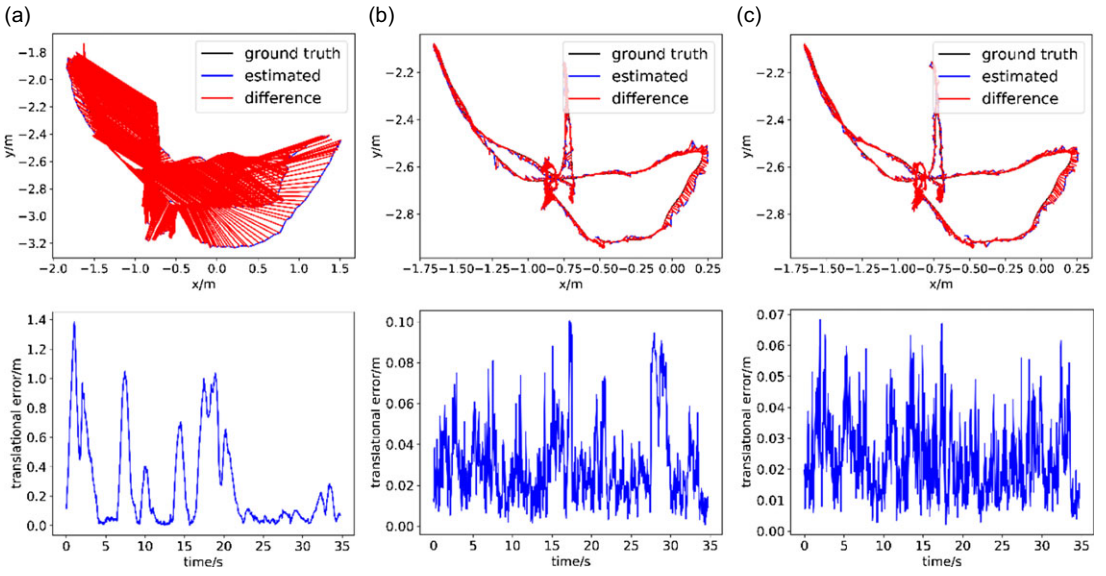

Figure 16. *ATE and RPE of the three algorithms on fr3_ walking _halfsphere dataset.*

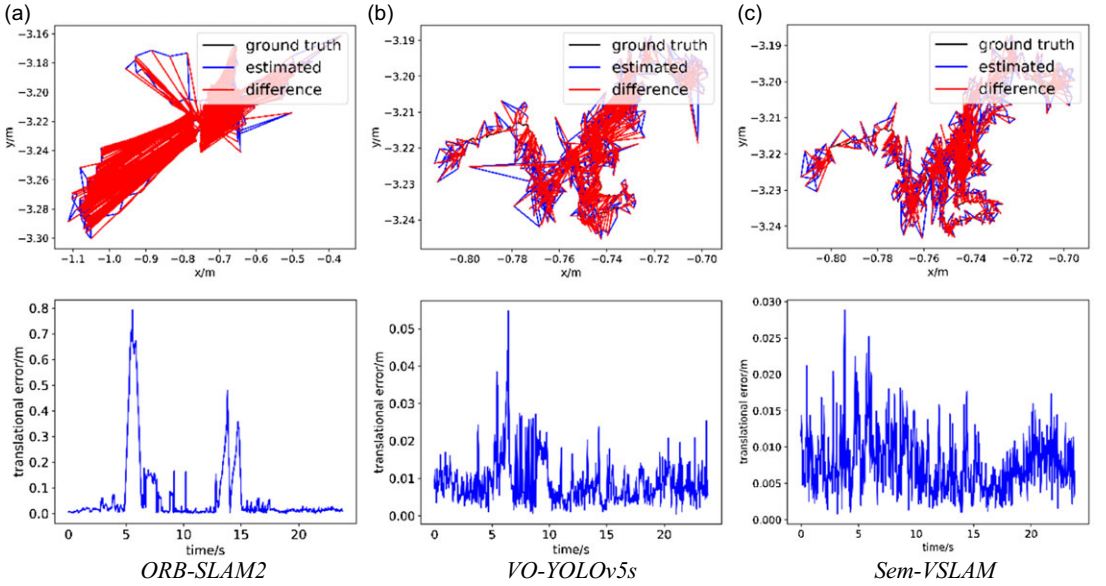

Figure 17. *ATE and RPE of the three algorithms on fr3_walking _static dataset.*

by the tracking thread to process each frame image. The tracking time performance of ORB-SLAM2, ORB-SLAM3, VO-YOLOv5s, KMOP-SLAM, and RLD-SLAM algorithms based on different hardware platforms was compared. The comparisons are shown in Table VII. In this article, the Sem-VSLAM algorithm has an average time of about 60.73 ms per frame of images, which is equivalent to the speed of about 17 frames per second. The Sem-VSLAM algorithm can basically achieve the excellent accuracy and real time and obtain comprehensive optimal effectiveness.

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

Table VI. *RMSE comparison of VSLAM algorithm ATE in dynamic environment.*

Dataset	ORB-SLAM3	VO-YOLOv5s	KMOP-SLAM	RLD-SLAM	YOLO-SLAM	DS-SLAM	Dyna SLAM	SLAM-PCD	RDS-SLAM
fr3_w_xyz	0.9178	0.0167	0.0190	0.0160	0.0146	0.0247	0.0164	0.0157	0.0213
fr3_w_rpy	1.0197	0.0455	0.0490	0.0318	**0.0216**	0.4442	0.0354	0.0453	0.1468
fr3_w_half	0.6572	0.0317	0.1760	0.0263	0.0283	0.0303	0.0296	0.0241	0.0259
fr3_w_static	0.3614	0.0086	0.0320	0.0075	0.0073	0.0081	0.0068	0.0077	0.0815
fr3_s_static	0.0090	0.0066	–	–	0.0066	0.0065	0.0108	0.0080	0.0088

DynaTM-SLAM	Static Fusion	DRSO-SLAM	RDMO-SLAM	CRF-SLAM	Amos-SLAM	DN-SLAM	DOR-SLAM	PR-SLAM	Ours
0.0150	0.1270	0.0158	0.0226	0.0160	**0.0140**	0.0150	0.0183	0.0165	0.0152
0.0288	–	0.0752	0.1283	0.0460	0.0270	0.0320	0.1168	0.0335	0.0405
0.0291	0.3910	0.0268	0.0304	0.0280	0.0250	0.0260	0.0246	**0.0230**	0.0296
0.0068	0.0140	0.0111	0.0126	0.0110	0.0070	0.0080	0.0076	0.0072	**0.0066**
0.0064	0.0130	0.0064	0.0066	–	–	–	0.0097	–	**0.0063**

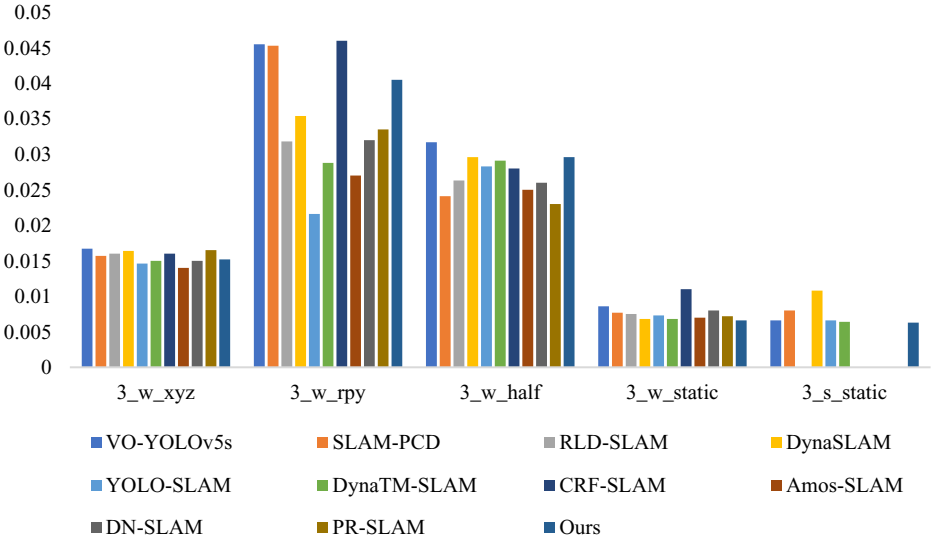

Figure 18. *Comparison of RMSE of VSLAM algorithm ATE in dynamic environment.*

3.4. The experiment on the TUM dataset

Figure 19(a) represents how ORB-SLAM2 algorithm constructs 3D point cloud maps containing dynamic feature points in the public TUM dataset, and Figure 19(b) represents how Sem-VSLAM algorithm constructs 3D point cloud maps without dynamic feature points in the public TUM dataset. The two figures verify that the proposed algorithm in this article is able to efficiently eliminate the dynamic feature points and construct 3D point cloud maps using only the remaining static feature points under the official public simulation data.

Figure 20(a) represents that ORB-SLAM2 algorithm constructs 3D point cloud maps containing dynamic feature points in the laboratory environment, and Figure 20(b) represents that the Sem-VSLAM algorithm constructs 3D point cloud maps without dynamic feature points in the laboratory environment. These two figures demonstrate that this article's algorithm is able to efficiently eliminate the dynamic feature points and construct 3D point cloud maps using only the remaining static feature points in the real laboratory environment.

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

Table VII. *Real-time comparison of VSLAM algorithm in dynamic environment (ms).*

Algorithms	Hardware platform	Tracking
ORB-SLAM2	Intel i9 CPU, GeForce RTX 3080 GPU	37.51
ORB-SLAM3	Intel i9 CPU, GeForce RTX 3080 GPU	36.42
VO-YOLOv5s	Intel i9 CPU, GeForce RTX 3080 GPU	68.24
KMOP-SLAM	Intel i7 CPU, GeForce GTX 1080Ti GPU	257.82
RLD-SLAM	Intel i7 CPU, GeForce GTX 1050Ti GPU	29.5
YOLO-SLAM	Intel i5 CPU, GeForce GTX1660Ti GPU	696.09
DS-SLAM	Intel i7 CPU, P4000 GPU	103.42
DynaSLAM	Nvidia Tesla M40 GPU	260.91
RDS-SLAM	GeForce GTX 2080 Ti GPU	205.42
DynaTM-SLAM	Intel i5 CPU, RTX 3070 GPU	147.00
DRSO-SLAM	Intel i7 CPU, GTX 960 M GPU	–
RDMO-SLAM	GeForce GTX 2080 Ti GPU	22-35
CRF-SLAM	Intel Core i9 CPU	90.21
Amos-SLAM	Intel i5 CPU, GeForce GTX 3060 GPU	92.88
DN-SLAM	Intel i9 CPU, RTX4090 GPU	–
DOR-SLAM	Intel i7 CPU, GeForce RTX 3070 GPU	170-200
PR-SLAM	AMD R5 CPU, RTX 2070Ti GPU	59.73
Ours	Intel i9 CPU, GeForce RTX 3080 GPU	60.73

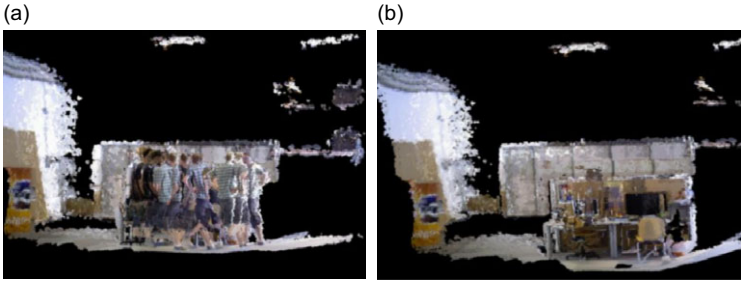

Figure 19. *3D point cloud map under official public data.*

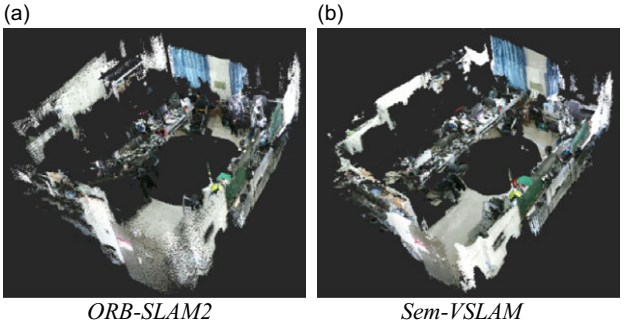

Figure 20. *3D point cloud map constructed in the laboratory environment.*

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

4. Conclusion

To solve the problems of low accuracy, poor robustness, and low real-time performance of VSLAMs in dynamic environments, this article improves the classical ORB-SLAM2 algorithm to propose a novel Sem-VSLAM algorithm. While extracting ORB features, a semantic segmentation thread is added in parallel to obtain semantic information, and then the segmentation effect is evaluated and the missing parts are repaired. The motion consistency detection algorithm is used to detect dynamic and static feature points and eliminate dynamic feature points. Finally, the processed feature points are used for interframe matching and pose estimation.

The experimental results on the public TUM dataset show that the Sem-VSLAM algorithm can effectively improve the accuracy and robustness of the VSLAM system in a dynamic environment and ensure its real-time performance. However, the Sem-VSLAM algorithm still has some points, which can be further improved: the dynamic object only takes people as the research object; when the camera rotates greatly, the image blur leads to the loss of tracking, and finally the positioning mapping will fail; for some potentially moving objects, there may also be problems that cannot be effectively detected and eliminated. Based on this limitation, we will consider combining the inertial measurement unit so that the system can reach satisfying applications.

Author contributions. Kang Zhang: Data curation, writing original draft, software, and validation. Chaoyi Dong: Conceptualization, methodology, and supervision. Hongfei Guo: Data curation and supervision. Qifan Ye: Software. Liangliang Gao: Validation. Shuai Xiang: Validation. Xiaoyan Chen: Software and validation. Yi Wu: Validation.

Financial support. This work was supported by the National Natural Science Foundation of China (61364018 and 61863029), Inner Mongolia Natural Science Foundation (2016JQ07, 2020MS06020, and 2021MS06017), Inner Mongolia Scientific and Technological Achievements Transformation Project (CGZH2018129), Industrial Technology Innovation Program of IMAST (2023JSYD01006), Science and Technology Plan Project of Inner Mongolia Autonomous Region (2021GG0264 and 2020GG0268), and Basic Research Funds for universities directly under the Inner Mongolia Autonomous Region (ZTY2023071).

Competing interests. The authors declare no Competing interests exist.

Ethical approval. None.

References

[1] R. Li, X. Zhang, S. Zhang, J. Yuan, H. Liu and S. Wu, "BA-LIOM: Tightly coupled laser-inertial odometry and mapping with bundle adjustment," *Robotica* **42**(3), 684–700 (2024).
[2] K. Zhang, C. Dong, L. Gao, Q. Ye, X. Chen, J. Zhao, F. Hao and S. Xiang, "A Graph-Optimized SLAM with Improved Levenberg-Marquardt Algorithm," **In:** *9th International Conference on Control, Decision and Information Technologies*, (IEEE, 2023) pp. 1821–1825.
[3] Y. Cai, Y. Ou and T. Qin, "Improving SLAM techniques with integrated multi-sensor fusion for 3D reconstruction," *Sensors* **24**(7), 2033 (2024).
[4] L. Gao, C. Dong, X. Liu, Q. Ye, K. Zhang and X. Chen, "A 2D laser SLAM graph optimization based on a position and angle partition and cholesky decomposition," *J Appl Sci Eng* **26**(9), 1255–1262 (2022).
[5] H. Kuang, Y. Li, Y. Zhang, Y. Wan and G. Ge, "Research on rapid location method of mobile robot based on semantic grid map in large scene similar environment," *Robotica* **40**(11), 4011–4030 (2022).
[6] X. Liu, C. Dong, Q. Wang, L. Gao, Q. Ye and X. Chen, "Research on a Visual SLAM Loop Closure Detection Algorithm Based on a VGG-19 Network," **In:** *2021 China Automation Congress*, (IEEE, 2021) pp. 4781–4785.
[7] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans Robot* **33**(5), 1255–1262 (2017).
[8] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel and J. D. Tardos, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans Robot* **37**(6), 1874–1890 (2021).
[9] L. Messina, S. Mazzaro, A. E. Fiorilla, A. Massa and W. Matta, "Industrial Implementation and Performance Evaluation of LSD-SLAM and Map Filtering Algorithms for Obstacles Avoidance in a Cooperative Fleet of Unmanned Aerial Vehicles," **In:** *3rd International Conference on Intelligent Robotic and Control Engineering (IRCE)*, (IEEE, 2020) pp. 117–122.
[10] J. Cheng, Y. Sun and M. Q.-H. Meng, "Robust semantic mapping in challenging environments," *Robotica* **38**(2), 256–270 (2020).
[11] A. Li, J. Wang, M. Xu and Z. Chen, "DP-SLAM: A visual SLAM with moving probability towards dynamic environments," *Inform Sci* **556**, 128–142 (2021).

https://doi.org/10.1017/S0263574724001553 Published online by Cambridge University Press

[12] L. Kenye and R. Kala, "Improving RGB-D SLAM in dynamic environments using semantic aided segmentation," *Robotica* **40**(6), 2065–2090 (2022).
[13] L. Riazuelo, L. Montano and J. Montiel, "Semantic Visual SLAM in Populated Environments," **In:** *2017 European Conference on Mobile Robots (ECMR)*, (IEEE, 2017) pp. 1–7.
[14] F. Zhong, S. Wang, Z. Zhang and Y. Wang, "Detect-SLAM: Making Object Detection and SLAM Mutually Beneficial," **In:** *IEEE Winter Conference on Applications of Computer Vision*, (IEEE, 2018) pp. 1001–1010.
[15] Q. Ye, C. Dong, X. Liu, L. Gao, K. Zhang and X. Chen, "A Visual Odometry Algorithm in Dynamic Scenes Based on Object Detection," **In:** *5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, (IEEE, 2022) pp. 465–470.
[16] J. Fang and Z. Fang, "Visual SLAM optimization in dynamic scenes based on object detection network," *J Beijing Univ Technol* **48**(05), 466–475 (2022).
[17] Y. Liu and J. Miura, "KMOP-vSLAM: Dynamic Visual SLAM for RGB-D Cameras Using K-Means and OpenPose," **In:** *IEEE/SICE International Symposium on System Integration (SII)*, (IEEE, 2021) pp. 415–420.
[18] Z. Zheng, S. Lin and C. Yang, "RLD-SLAM: A robust lightweight VI-SLAM for dynamic environments leveraging semantics and motion information," *IEEE Trans Ind Electron* **71**(11), 14328–14338 (2024).
[19] W. Wu, L. Guo, H. Gao, Z. You, Y. Liu and Z. Chen, "YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint," *Neur Comput Appl* **34**(8), 1–16 (2022).
[20] C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang, Q. Wei and Q. Fei, "DS-SLAM: A Semantic Visual SLAM Towards Dynamic Environments," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (IEEE, 2018) pp. 1168–1174.
[21] V. Badrinarayanan, A. Kendall and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans Pattern Anal Mach Intell* **39**(12), 2481–2495 (2017).
[22] B. Bescos, J. Fácil, J. Civera and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robot Autom Lett* **3**(4), 4076–4083 (2018).
[23] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," **In:** *Proceedings of the IEEE international Conference on Computer Vision*, (IEEE, 2017) pp. 2961–2969.
[24] Y. Fan, Q. Zhang, S. Liu, Y. Tang, X. Jing, J. Yao and H. Han, "Semantic SLAM with more accurate point cloud map in dynamic environments," *IEEE Access* **8**, 112237–112252 (2020).
[25] Y. Liu and J. Miura, "RDS-SLAM: Real-time dynamic SLAM using semantic segmentation methods," *IEEE Access* **9**, 23772–23785 (2021).
[26] M. Zhong, C. Hong, Z. Jia, C. Wang and Z. Wang, "DynaTM-SLAM: Fast filtering of dynamic feature points and object-based localization in dynamic indoor environments," *Robot Auton Syst* **174**, 104634 (2024).
[27] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks," **In:** *Advances in Neural Information Processing Systems*, vol. **28** (2015).
[28] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan and C. Zitnick. "Microsoft Coco: Common Objects in Context," **In:** *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, (IEEE, 2014) pp. 740–755.
[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (IEEE, 2012) pp. 573–580.
[30] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon and D. Cremers, "Staticfusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments," **In:** *IEEE International Conference on Robotics and Automation (ICRA)*, (IEEE, 2018) pp. 3849–3856.
[31] N. Yu, M. Gan, H. Yu and K. Yang, "DRSO-SLAM: A Dynamic RGB-D SLAM Algorithm for Indoor Dynamic Scenes," **In:** *33rd Chinese Control and Decision Conference (CCDC)*, (IEEE, 2021) pp. 1052–1058.
[32] Y. Liu and J. Miura, "RDMO-SLAM: Real-time visual SLAM for dynamic environments using semantic label prediction with optical flow," *IEEE Access* **9**, 106981–106997 (2021).
[33] Z.-J. Du, S.-S. Huang, T.-J. Mu, Q. Zhao, R. R. Martin and K. Xu, "Accurate dynamic SLAM using CRF-based long-term consistency," *IEEE Trans Vis Comput Graph* **28**(4), 1745–1757 (2022).
[34] Y. Zhuang, P. Jia, Z. Liu, L. Li, C. Wu, X. Lu, W. Cui and Z. Liu, "Amos-SLAM: An anti-dynamics two-stage RGB-D SLAM approach," *IEEE Trans Instrum Meas* **73**, 1–10 (2023).
[35] C. Ruan, Q. Zang, K. Zhang and K. Huang, "DN-SLAM: A visual SLAM with ORB features and neRF mapping in dynamic environments," *IEEE Sens J* **24**(4), 5279–5287 (2023).
[36] G. Liao and F. Yin, "DOR-SLAM: A Visual SLAM Based on Dynamic Object Removal for Dynamic Environments," **In:** *China Automation Congress (CAC)*, (IEEE, 2023) pp. 1777–1782.
[37] H. Zhang, J. Peng and Q. Yang, "PR-SLAM: Parallel real-time dynamic SLAM method based on semantic segmentation," *IEEE Access* **12**, 36498–36514 (2024).

Cite this article: K. Zhang, C. Dong, H. Guo, Q. Ye, L. Gao, S. Xiang, X. Chen and Y. Wu (2024). "A semantic visual SLAM based on improved mask R-CNN in dynamic environment", Robotica **42**, 3570–3591. https://doi.org/10.1017/S0263574724001553