Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5538840, 16 pages
https://doi.org/10.1155/2021/5538840

Research Article

OFM-SLAM: A Visual Semantic SLAM for Dynamic Indoor Environments

Xiong Zhao,[1] **Tao Zuo**,[1,2] **and Xinyu Hu**[1]

[1] *College of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China*
[2] *Engineering Research Center for Metallurgical Automation and Detecting Technology of Ministry of Education, Wuhan University of Science and Technology, Wuhan 430081, China*

Correspondence should be addressed to Tao Zuo; zuomu666@163.com

Received 18 January 2021; Revised 28 February 2021; Accepted 19 March 2021; Published 10 April 2021

Academic Editor: Luis Payá

Most of the current visual Simultaneous Localization and Mapping (SLAM) algorithms are designed based on the assumption of a static environment, and their robustness and accuracy in the dynamic environment do not behave well. The reason is that moving objects in the scene will cause the mismatch of features in the pose estimation process, which further affects its positioning and mapping accuracy. In the meantime, the three-dimensional semantic map plays a key role in mobile robot navigation, path planning, and other tasks. In this paper, we present OFM-SLAM: Optical Flow combining MASK-RCNN SLAM, a novel visual SLAM for semantic mapping in dynamic indoor environments. Firstly, we use the Mask-RCNN network to detect potential moving objects which can generate masks of dynamic objects. Secondly, an optical flow method is adopted to detect dynamic feature points. Then, we combine the optical flow method and the MASK-RCNN for full dynamic points' culling, and the SLAM system is able to track without these dynamic points. Finally, the semantic labels obtained from MASK-RCNN are mapped to the point cloud for generating a three-dimensional semantic map that only contains the static parts of the scenes and their semantic information. We evaluate our system in public TUM datasets. The results of our experiments demonstrate that our system is more effective in dynamic scenarios, and the OFM-SLAM can estimate the camera pose more accurately and acquire a more precise localization in the high dynamic environment.

1. Introduction

Simultaneous Localization and Mapping (SLAM) enables the mobile robot to estimate the current position and posture through the sensor and the corresponding motion estimation algorithm without any prior environmental information and establish a three-dimensional map of the environment. RGB-D cameras have become one of the important sensors for the mobile robot assembly due to its cost effectiveness, application occasions, and availability of rich scene information. In the meantime, with the development of computer vision, deep learning, and the improvement of hardware computing capabilities, the research on vision-based Visual SLAM (VSLAM) continues to deepen and is widely used in fields such as autonomous driving, mobile robots, and drones.

Most SLAM algorithms are not robust in dynamic environments; it is easier to calculate their own pose based on static environment information. Dynamic objects in the environment, such as walking people, opening and closing doors, or any other change in the environment, will bring unpredictable abnormal observations to the system, reduce the positioning accuracy of mobile robots, cause dynamic interfering objects to become part of the environment map, and even cause the SLAM system to fail completely. Therefore, the existing algorithms are not well applicable to dynamic environments, and the accuracy and robustness of SLAM systems in dynamic environments need to be improved. At the same time, intelligent mobile robots need to have a higher level of understanding of the scene to perform complex tasks, such as the semantic information of surrounding objects and their location information. The semantic map contains not only the spatial structure information of the surrounding environment but also the semantic information of the environment. Semantic information can be used to reason about objects and environments around the robot, or to provide additional information for navigation and robot tasks. Therefore, the correct analysis of the environment and the establishment of a semantic map are the prerequisites for the interaction between the human and the robot in the intelligent system, and it is also the basis for the mobile robot to perform advanced tasks. With the development of deep learning, some networks can achieve good performance in semantic segmentation, such as Semanticfusion [1], Semantic 3D Mapping [2], MaskFusion [3], and MID-Fusion [4]. This global semantic information obtained through semantic segmentation networks can help robot navigation and path planning, which will significantly improve the intelligence of mobile robots.

In view of the above problems, it is necessary to increase the processing of moving objects in the environment, reduce its impact on the visual SLAM system, and improve the accuracy and robustness of the SLAM system positioning and mapping. At the same time, the semantic segmentation algorithm and the visual SLAM algorithm need to be merged to construct a semantic map of the environment to obtain richer information and a better understanding of the scene. Based on the complex and dynamic indoor dynamic environment, this paper explores the methods of constructing the semantic map in the dynamic environment, combining the visual SLAM system based on the RGB-D camera and the deep learning method. The methods proposed in this paper have a certain value in semantic map construction in the dynamic environment and can help robots achieve more intelligent navigation tasks.

The contributions of this paper can be seen as follows:

(1) A novel OFM-SLAM system is proposed based on ORB-SLAM2 for more accurate positioning and mapping in dynamic scenarios.

(2) A deep convolutional neural network MASK-RCNN framework is adopted to eliminate potential dynamic objects, merge pose information and semantic information, and build a semantic target database to build the high-level semantic map.

(3) We combine the optical flow method with the network of MASK-RCNN to get masks of moving objects, then, we fully remove the dynamic objects in the scenes.

(4) A semantic octree map is constructed using the results of semantic segmentation, and then, we use the log-odds method to remove the residual part of the dynamic target in the map. The dynamic factors in the map are eliminated to generate a three-dimensional octree map of the static environment, which provides reliable environmental information for the navigation of the mobile robot.

The rest of the paper is structured as follows: introduction of related works in Section 2, detailed description of the materials and method we proposed in Section 3, our experimental part in Section 4, and finally, the summary and expectation the future work in Section 5.

2. Related Work

2.1. Visual SLAM. With the continuous improvement of algorithms and computer hardware performance, visual SLAM has developed rapidly in recent years and has been successfully applied to many occasions, such as service robots, sweeping robots, and drones. The basic principle of visual SLAM is to observe the same scene from different perspectives and perform data correlation between different images to calculate the camera movement between different frames. Visual SLAM algorithms are mainly divided into filter methods and optimization methods. In the early SLAM algorithms, filter methods [5] were widely used, such as the Kalman filter (KF) and particle filters. They are only focused on the state estimation at the current moment, but do not make a full use of the previous state. In recent years, nonlinear optimization has gradually become the mainstream solution of the visual SLAM method. With the development of graph optimization theory, many SLAM systems based on optimization methods, such as ORB-SLAM [6] and SVO [5], are proposed to construct a more accurate map. According to the form of map construction, the SLAM algorithm is divided into sparse methods, semidense method, and dense method. According to the different cameras used, it is divided into monocular SLAM, binocular SLAM, and RGB-D SLAM. Visual SLAM algorithms can also be divided into indirect method, semidirect methods, and direct methods according to whether feature points are needed. Mur-Artal and Tardos proposed the ORB-SLAM, which provides a classic visual SLAM algorithm, and it can achieve long-term operation in large scenarios due to the pose map optimization used in the optimization part, which can correct trajectory errors in the system. Subsequently, the author made improvements to the original system, adding support for binocular cameras and RGB-D depth cameras, and developed the ORB-SLAM2 [7] system. The ORB-SLAM2 algorithm is an outstanding algorithm framework among the visual SLAM algorithms based on the feature point method in recent years.

In an ideal environment without dynamic targets, the visual SLAM system can operate normally without interference, but in the actual environment, there are many dynamic targets, such as walking people, running vehicles, and other targets. In the SLAM system, the dynamic feature points extracted on the dynamic target will directly affect the accuracy of the robot's pose estimation, causing errors and drift in the system. To improve the accuracy and robustness of visual SLAM in a dynamic environment, it is necessary to eliminate the influence of dynamic targets in the environment. Therefore, we need to identify and process dynamic targets in the environment.

2.2. SLAM in Dynamic Scenes. In dynamic scenes, scene flow methods are often used to detect dynamic objects in images. Alcantarilla et al. [8] used the scene flow changes of the features to identify the dynamic features in the system, and after removing these dynamic features, the pose estimation was performed. However, the method removed too many dynamic feature points, and the static feature points were prone to be insufficient. Palazzolo [9] presented re-fusion which uses an efficient direct tracking on the truncated signed-distance function (TSDF) and leverage color information encoded in the TSDF to estimate the pose of the sensor. For detecting dynamics, they exploited the residuals obtained after an initial registration, together with the explicit modeling of the free space in the model. Zhang et al. [10] presented FlowFusion using optical flow residuals to highlight the dynamic semantics in the RGB-D point clouds and provided more accurate and efficient dynamic (static) segmentation for camera tracking and background reconstruction, and there are other geometric methods used for large-scale and dynamic environment, such as Lsd-SLAM [11], Static-Fusion [12], EM-Fusion [13], and SOF-SLAM [14] .

In recent years, deep learning-based methods have achieved significant results in the tasks of target recognition and semantic segmentation. Therefore, many researchers believe that applying deep learning technology to visual SLAM is the key to solving dynamic environment problems. The combination of deep learning and visual SLAM enhances the ability of mobile robots to understand and perceive the surrounding environment. In dynamic scenes, Runz et al. [3] presented MaskFusion, a real-time, object-aware, semantic, and dynamic RGB-D SLAM system that goes beyond traditional systems which output a purely geometric map of a static scene. Xu et al. [4] proposed a new multi-instance dynamic RGB-D SLAM system MID-Fusion using an object-level octree-based volumetric representation. It can provide robust camera tracking in dynamic environments and, at the same time, continuously estimate geometric, semantic, and motion properties for arbitrary objects in the scene. Bescos et al. [15] proposed to add dynamic target detection and background repair functions to the ORB-SLAM2 system and used Mask-RCNN [16] for instance segmentation to obtain dynamic target parts, which greatly improves the SLAM performance, but the system cannot achieve real-time operation. The DS-SLAM system proposed by Yu [17] combines the semantic segmentation network SegNet with ORB-SLAM2 to reduce the impact of dynamic targets on the system. Dai et al. [18] proposed a method that utilizes the correlation between map points which could separate points that are part of the static scene and moving objects into different groups. Cui et al. [19] improved the yolov3 algorithm to detect indoor objects, and the real-time semantic segmentation network model based on deep learning is used to segment indoor objects to achieve the classification of objects. Then, they combine the depth information to build the three-dimensional semantic map.

3. Materials and Methods

3.1. System Overview. We proposed the OFM-SLAM system which is based on the state-of-the-art ORB-SLAM2. We add the dynamic objects' processing module and semantic mapping module to the system. The image information stream input to the visual SLAM algorithm often contains various objects, combined with the semantic information extraction of the target detection network MASK-RCNN. The precise geometric information obtained by the RGB-D camera and SLAM algorithm enables the robot to obtain more structured, semantic, and hierarchical map information from the surrounding environment. Figure 1 shows the flow chart of OFM-SLAM.

3.2. Moving Objects' Detection Based on MASK-RCNN. We adopt the network of MASK-RCNN to detect the moving object, with the continuous development of machine learning, more and more semantic segmentation networks have been proposed, and they can achieve pixel-level semantic segmentation. In OFM-SLAM, Mask R-CNN is used to obtain semantic information which was proposed by He et al. [16].

Mask R-CNN can obtain pixel-level semantic segmentation and instance labels at the same time and has high accuracy. For each frame of the input image, Mask-RCNN first obtains the corresponding feature map through the trained ResNet network and sets a fixed number of ROI (Region of Interest) for each feature in the feature map to obtain multiple candidates' ROI through RPN (Region Proposal Network). Then, the ROI Align operation is used to realize the correspondence between the feature map and the original image and adopt a fully connected network for each ROI to classify and establish a bounding box. In another branch, the FCN (Fully Convolution Network) is used to achieve semantic segmentation. The results of semantic segmentation can be used for the construction of semantic maps, and the instance labels can be used to determine potential dynamic targets in the scene.

Figure 2 shows the pipeline of instance segmentation using MASK-RCNN. The Mask-RCNN network mainly consists of two parts. The first part scans the entire input image and generates a candidate area that may contain the target object. The second part classifies the generated candidate area and generates the mask and bounding box through convolution operation. Then, we use the semantic information generated by the MASK-RCNN to construct the semantic map.

The network input is the original RGB image, and the output is a segmented image containing semantic labels. In order to introduce Mask-RCNN into the SLAM framework, on the one hand, it needs to provide semantic information for the SLAM algorithm, and on the other hand, it provides the SLAM algorithm with a priori information that has a high probability of being a dynamic target in the scene. In order to enable the segmentation results of Mask-RCNN to better integrate the SLAM algorithm, the segmentation results of Mask-RCNN are preprocessed, and the segmentation bounding box is removed from the output results of the original Mask-RCNN. While, preserving the semantic labels and segmentation results, we visualize objects that have a high probability of being a dynamic target in the image.

As shown in Figure 3, visualization refers to setting the pixel value of the object detected as a human in the image to

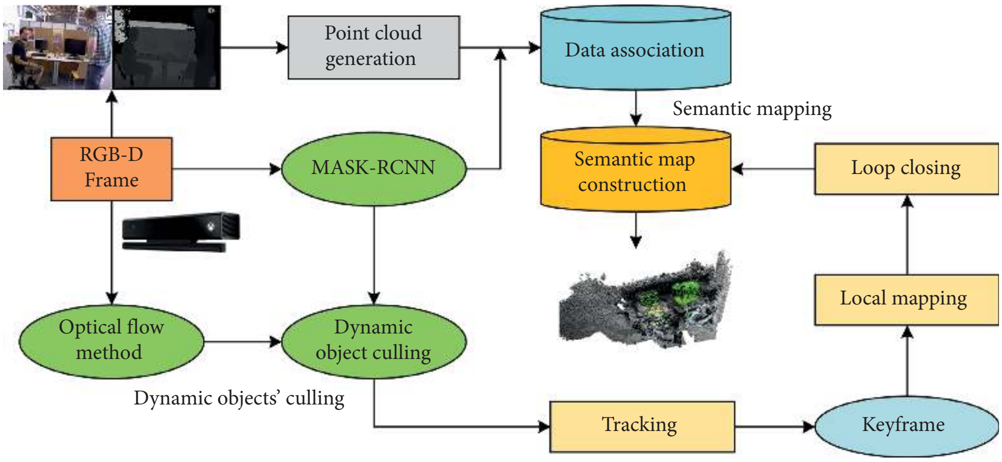


Figure 1: Pipeline of OFM-SLAM which contains five threads as follows: tracking, local mapping, loop closing, dynamic objects' culling, and semantic mapping.

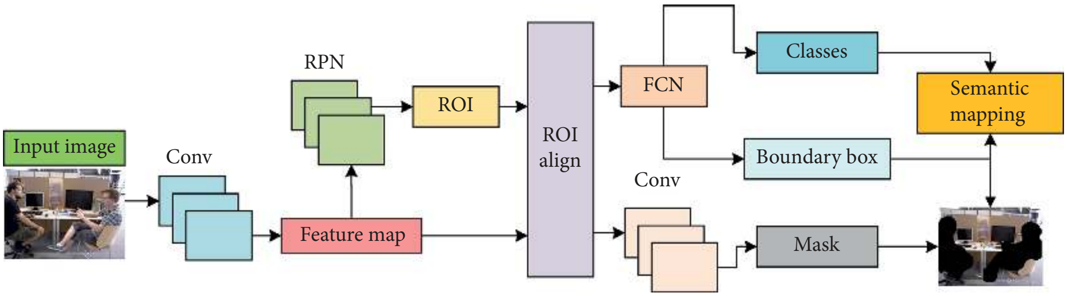


Figure 2: Pipeline of dynamic object detection using MASK-RCNN.

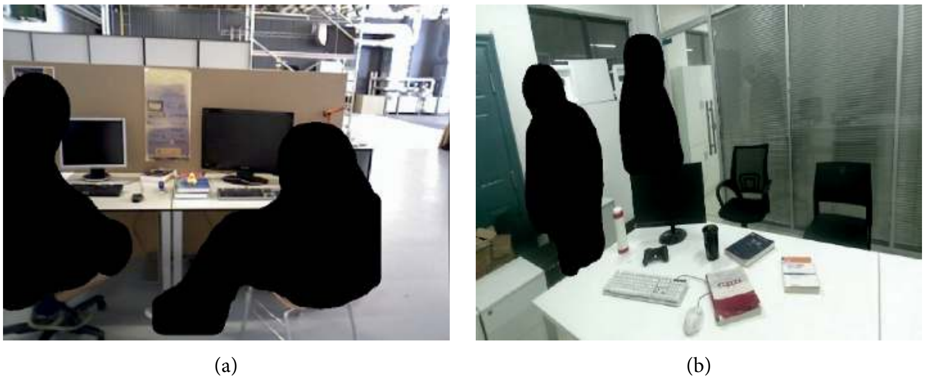


Figure 3: MASK generation using Mask-RCNN in TUM RGB-D datasets and our datasets in our laboratory. (a) Sitting-xyz. (b) Walking-lab.

0, and the pixel value of the remaining objects including the background remains unchanged.

3.3. Dynamic Points' Detection Algorithm Based on the Optical Flow. We use the optical flow method to detect the potential dynamic point in the scene. The optical flow field is the instantaneous velocity field that describes the movement of pixels in the image, that is, the position and velocity changes of the pixels in the image. It uses the temporal changes of the pixel grayscale in the image sequence and the correlation between adjacent frames. The algorithm calculates the motion information of any object between the pixels in the image and then knows the correspondence between the feature points in the current frame image and the previous frame image. According to the velocity vector characteristics of each pixel, the image can be dynamically analyzed. If there is no moving target in the image, the optical flow vector changes continuously throughout the image area. When there are moving objects in the image, there is relative movement between the target and the background. The velocity vector formed by the moving object must be different from the background velocity vector so that the position of the moving object can be calculated.

The dynamic point detection algorithm in this paper is based on the optical flow method. It uses the sparse pyramid Lucas–Kanade (LK) optical flow to track some points in the image to directly obtain the corresponding relationship of the feature points. It does not require descriptor calculation and feature matching process, so it has better real-time performance. Since the consistency of the optical flow is used for moving object detection, the choice of optical flow threshold has a greater impact on the acquisition of dynamic point information. Therefore, the dynamic point detection algorithm in this paper only uses the optical flow for feature point tracking and uses it after obtaining specific corresponding feature points. The epipolar constraint makes dynamic point determination. The pseudocode of the specific dynamic point detection algorithm is shown in Algorithm 1.

By tracking the optical flow of two consecutive frames of images, several pairs of matching points between the images can be obtained so that the positional relationship between the two frames can be restored. The fundamental matrix F describes the relative transformation relationship between the two frames of images. The solution of F can be calculated using the eight pairs of matching feature points.

Consider a pair of matching points, their normalized coordinates are $p_i = (u_i, v_i, 1)$ and $p_i' = (u_i', v_i', 1)$. According to the epipolar geometric constraints, there are

$$(u_i, v_i, 1)\begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & 1 \end{pmatrix}\begin{pmatrix} u_i' \\ v_i' \\ 1 \end{pmatrix} = p_i^T F p_i' = 0. \tag{1}$$

Since the fundamental matrix F itself is equivalent, 8 pairs of matching points can be used to calculate the fundamental matrix F to obtain the correspondence between the two frames of images.

Using the basic matrix F, the key point p_1 in the previous frame can be projected to the current frame to obtain the epipolar line l_2 in the current frame, which is the search domain of the projection point of the spatial point P in the image I_2. By calculating the distance from the key point p_2 to the polar line l_2 obtained by optical flow tracking, it is determined whether it is a dynamic point:

$$l_2 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = F p_1' = F \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \tag{2}$$

where X, Y, and Z represent the line vector of the epipolar line and F is the basic matrix.

The distance D from the matching feature point p_2 tracked by the optical flow to the corresponding polar line l_2 can be calculated by the following equation:

$$D = \frac{\left|p_2'^T F p_1\right|}{\sqrt{\|X\|^2 + \|Y\|^2}}. \tag{3}$$

If D>threshold, then p_1 and p_2 are dynamic points, and the threshold is set to 6 in the algorithm in this paper. Figure 4 shows the result of the optical method.

3.4. Tracking without the Moving Feature Point. There are many feature descriptor methods, which are divided into two categories: gradient histogram-based feature descriptors, such as SIFT and SURF [20], and binary feature descriptors, such as FAST [21], Oriented Fast and Rotated BRIEF (ORB) [22], and Binary Robust Independent Elementary Features (BRIEF) [23]. The ORB algorithm was proposed by Rublee et al. in 2011 [22]. The algorithm extracts feature points through the FAST algorithm and calculates the descriptor through BRIEF, which makes the system more robust to noise. Therefore, our system tracks with the ORB feature points.

Tracking calculation is performed for every new frame of the image stream. The main idea is to find the relative relationship between the current frame and the existing key frames by matching with the key frames in the map to update and calculate the pose of the current frame. We combine the MASK-RCNN and the optical method to remove the dynamic points in the frame; then, the pose of the camera is tracked using the static points in the scene. Figure 5 shows the flow chart of the tracking module of OFM-SLAM. We feed the MASK-RCNN network with the RGB-D image to eliminate the potential dynamic points. Then, we initialize the pose of the camera and perform feature point tracking-combined optical flow method to fully remove the dynamic points. And, the pose of the camera is optimized iteratively by the least square method.

As shown in Figure 6, we compare the tracking result with ORB-SLAM2. It can be seen that our methods do not track the feature points on dynamic objects so that we can get more accurate pose of the camera.

It can be seen in Figure 6(b) that there are still feature points on the chair, due to the chair has not beenmoved or moved slowly by people in the scene. In fact, in our system, we set people as dynamic objects regardless of whether they move. In order to solve the potential dynamic feature points in the scene, such as chairs, books, mouse, and other potential dynamic objects, we combine the system with the optical flow method to remove them.

3.5. Semantic Map Construction. The scene map is the basis for the mobile robot to interact with the environment. At the same time, the established map can help the mobile robot better understand the scene for its positioning, navigation, and obstacle avoidance tasks. On the one hand, most of the mapping links of the previous SLAM algorithm are designed for static environments. The lack of corresponding processing for dynamic objects in the environment causes them to be in different positions on the map at different times, which seriously affects the consistency of the scene map. On

```
Input: Last frame, $I_1$; Last frame's keypoints, $P_1$; Current frame, $I_2$;
Output: Dynamic points, $M$
Current frame's keypoints $P_2 = \text{CalcopticalFlowPyrLK}\ (I_1, I_2, P_1)$
Dynamic points' detection
$F = \text{FindFundamentalMatrix}\ (P_1, P_2)$
for each matched points' pair $p_1$, $p_2$ in $P_1$, $P_2$
    $I = \text{CalcEpipolarLine}\ \ (p_1, F)$
    $D = \text{Distance}\ (I, p_2)$
    if $D > \sigma$ then
        add $p_2$ in $M$
    end if
end for
```

Algorithm 1: Dynamic points' detection algorithm.

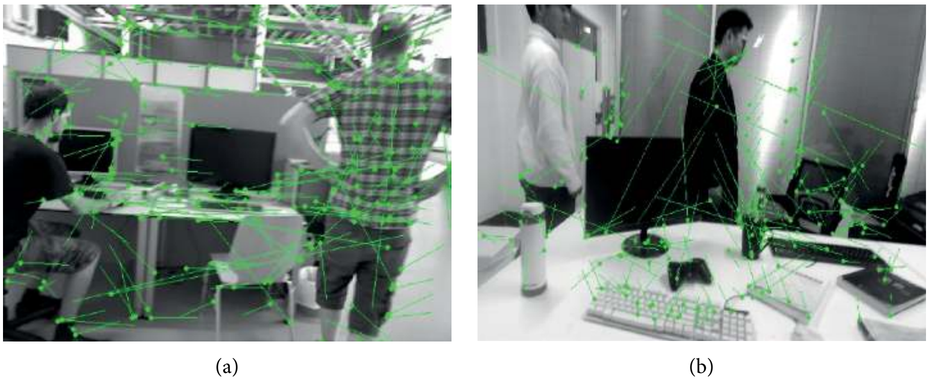

(a) (b)

Figure 4: Results of the optical flow method in various conditions. (a) Tum datasets of walking-xyz. (b) Actual environment in our lab. The green line shows the moving direction of the feature points in frames.

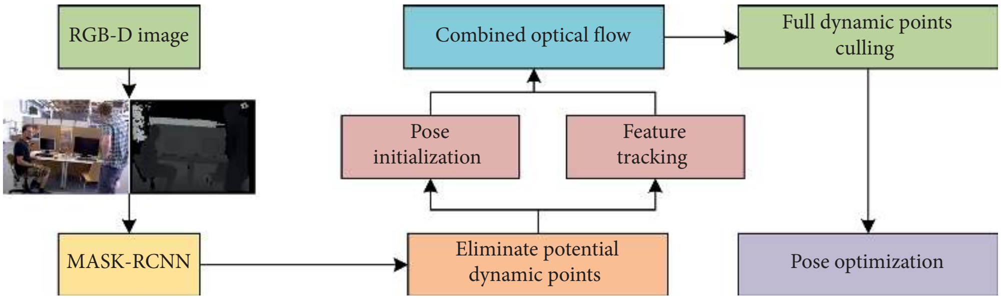
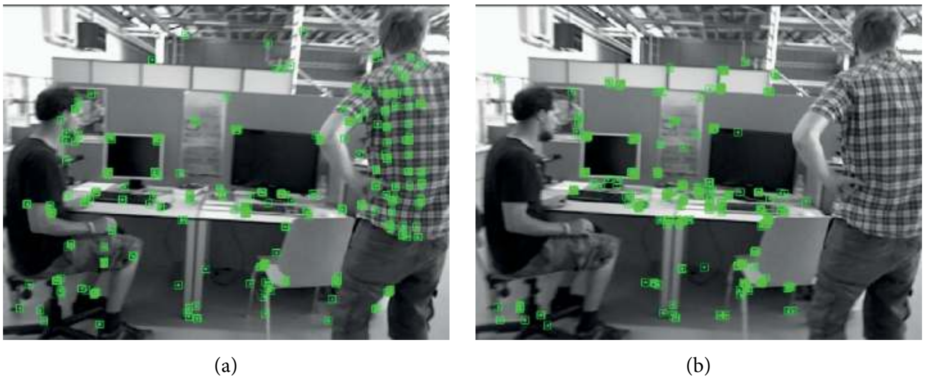


Figure 5: Pipeline of tracking without dynamic points.

the other hand, the established map does not make full use of the semantic information of the environment, and it is usually based on geometric information, such as sparse maps and point cloud maps based on the landmark. The establishment of three-dimensional semantic scenes is mainly divided into two methods at the semantic level: one is to construct a spatial map of the scene first, and then, use deep learning methods to train point clouds or voxels to obtain semantic information. The other is using semantic segmentation which is performed on the two-dimensional image to obtain the semantic information, and then, combine it with the depth map to introduce the semantic information into the three-dimensional space to obtain the semantic map. Since OFM-SLAM has performed the semantic segmentation of the image before the robot pose estimation is performed at the front end, we adopt the latter to construct the semantic map.

For semantic map construction in a dynamic environment, we add a semantic mapping thread. The framework of the algorithm system in the overall dynamic environment is shown in Figure 7.

When we initially get the semantic map, there will also be some undetected dynamic target parts in the map. It can be solved by the log-odds method. The logarithmic value of the


(a)

(b)

Figure 6: (a) Tracking with ORB-SLAM2. (b) Tracking using our methods.

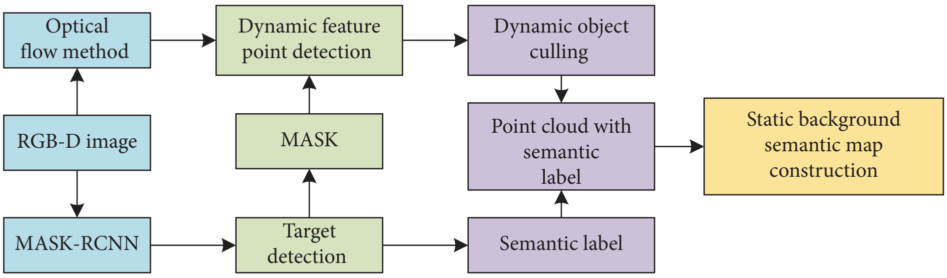

Optical flow method
Dynamic feature point detection
Dynamic object culling
RGB-D image
MASK
Point cloud with semantic label
Static background semantic map construction
MASK-RCNN
Target detection
Semantic label

Figure 7: The pipeline of input and output.

probability is used to describe the probability that a voxel in the octree is occupied. If the probability value of a voxel is greater than a threshold, the voxel is considered to be occupied. The probability of a certain voxel being occupied is expressed as $p \in [0, 1]$, the log odds of p is $l \in R$, logit is the logarithm of odds, and the two can be transformed by the following equation:

$$l = \text{logit}(p) = \log\left(\frac{p}{1-p}\right). \tag{4}$$

The inverse transformation of the two is expressed as follows:

$$p = \log\ \text{it}^{-1}(l) = \frac{\exp(l)}{\exp(l)+1}. \tag{5}$$

In the octree map, the logarithmic value of probability l is used to represent the occupancy of the voxel. When the voxel is observed to be occupied, the logarithmic value of the probability will increase. If it is observed that the voxel is not occupied, reduce the logarithmic value of the probability. Then, the probability that the voxel is occupied is solved using the logarithm of the probability obtained. Suppose a certain voxel is denoted as n, Z_t represents the observation result of voxel n at time t and using $L = (n|Z_{1:t+1})$ represents the logarithm value of the probability of the voxel from the beginning to $t+1$.

The log-odds calculation process is shown in the following equation:

$$L = L(n \mid Z_{1:t+1}) + L(n \mid Z_t). \tag{6}$$

When voxel n is observed to be occupied at time t, then $L(n|Z_t) = \tau$, or $L(n|Z_t) = 0$, where τ is the preset value. When a voxel is repeatedly observed to be occupied or unoccupied, the logarithmic value of the probability increases or decreases accordingly. The occupation probability p of a voxel can be obtained by the inverse transformation of the logarithm of the probability. When the probability p is greater than the set threshold, the voxel is finally considered to be occupied and the voxel is added to the map. The probability logarithm method can remove the residual part of the dynamic target in the map, which is beneficial to the SLAM system to construct a robust octree map. Figure 8(a)) shows the semantic octree map construction using the original TUM RGB-D datasets freiburg3_waking_xyz. Figure 8(b) shows the semantic octree map we constructed that only contains the static part of the whole map.

4. Results and Evaluation

We experimented the OFM-SLAM on eight public TUM datasets to validate our system. And, the tool of the Evo is used to estimate the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE). In the meantime, the accuracy is compared with other systems which are state-of-the-art on two kinds of datasets (including datasets of high-dynamic scenes and low-dynamic scenes).

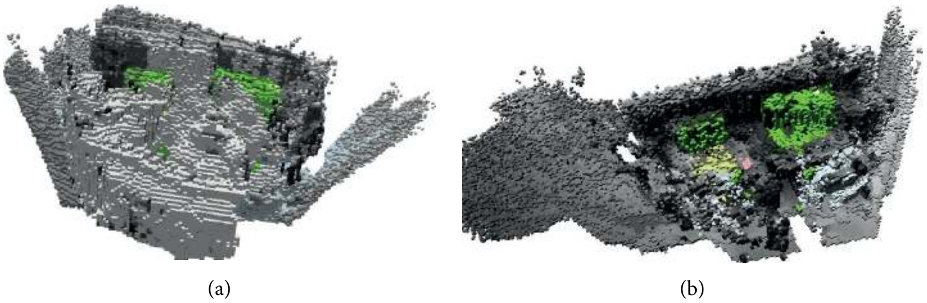


Figure 8: (a) Semantic octree map construction that contains moving objects. (b) Semantic octree map construction removing dynamic objects.

The algorithm was executed on a desktop PC with an Intel Core TM i3-8100 with 8 GB of memory and a NVIDIA GeForce GTX-1070 graphical card.

4.1. ATE in Low-Dynamic Scenes. The absolute trajectory error (ATE) is the direct difference between the estimated pose and the real pose. Figures 9 and 10 show the ATE result in low-dynamic scenes using ORB-SLAM2 and our methods. The black and blue line represents the camera trajectory of the ground truth and the estimated result of the system, and the red line represents the difference between the ground truth and the estimated result which are the same in Figures 11 and 12. We can see that the result of OFM-SLAM is competitive in low-dynamic scenarios compared with ORB-SLAM2.

Since the result of OFM-SLAM is competitive in low-dynamic scenarios compared with ORB-SLAM2, Figure 13 shows more details in x, y, and z views of the trajectory results. In datasets sitting-rpy, the trajectory error of OFM-SLAM is larger compared with the ground truth, and the estimated trajectory by ORB-SLAM2 due to the dynamic objects does not move or moves slowly. However, it can be seen that the trajectory error of OFM-SLAM and ORB-SLAM2 in datasets sitting-halfsphere, sitting-static, and sitting-xyz is almost no deviation.

4.2. ATE in High-Dynamic Scenes. Figures 11 and 12 show the ATE result in high-dynamic scenes using ORB-SLAM2 and our methods. We can see the result of OFM-SLAM is much more accurate in high-dynamic scenarios compared with ORB-SLAM2 due to the moving objects in the environment.

4.3. OFM-SLAM RPE in Low- and High-Dynamic Scenes. The relative pose error (RPE) is used to calculate the difference between the actual pose and estimated pose changes at the same time interval, which is suitable for estimating the drift of the system. Figures 14 and 15 show the RPE result in four datasets (sitting-halfsphere, sitting-xyz, walking-halfsphere, and waking-xyz) using our methods.

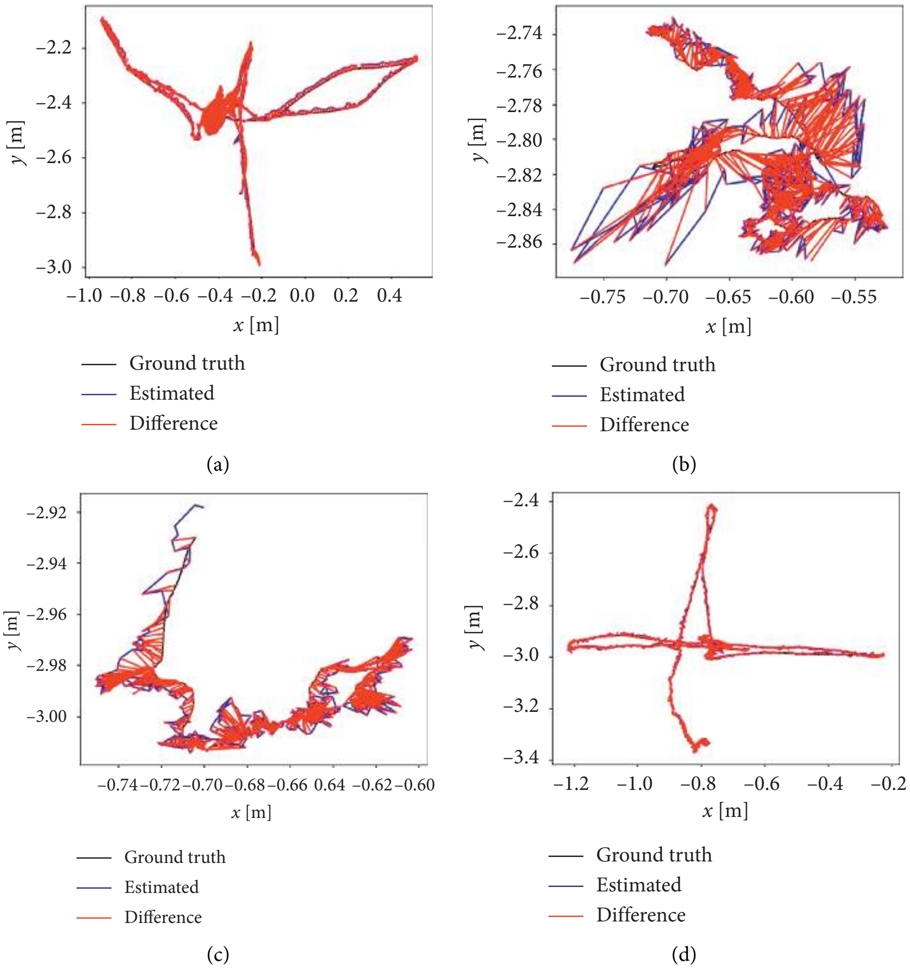
From Figure 14, it can be seen that the relative pose mean error is 1.1833 (cm) and 0.8697 (cm) in datasets sitting-halfsphere and sitting-xyz which are low dynamic. From Figure 15, it can be seen that the relative pose mean error is 1.1850 (cm) and 0.9971 (cm) in datasets walking-halfsphere and walking-xyz which are high dynamic. It shows the relative pose error of our system is little in dynamic scenarios.

4.4. A Comparison with ORB-SLAM2. Figure 16 shows the absolute pose error (APE) result of our system on TUM datasets waking-xyz compared with that of ORB-SLAM2. The blue part refers to the estimated camera trajectory of the ORB-SLAM2, and the green part represents the camera trajectory generated by OFM-SLAM. It can be seen that our system achieved high accuracy in high-dynamic scenes.

4.5. Benchmark. We compare the ATE RMSE (Root Mean Square Error) and RPE RMSE of OFM-SLAM with DS-SLAM [17] and ORB-SLAM2 [7] According to Table 1, we can find that OFM-SLAM is competitive in contrast to the two other system, and our method performs better and more accurate in high-dynamic scenes from the six datasets in TUM that contain moving objects.

4.6. Semantic Map Construction in Dynamic Environment. In Figure 17, we present the result of semantic map construction using eight TUM RGB-D datasets which contains moving objects. They are divided into low dynamics and high dynamics, and we run our system in the two kinds of scenarios. It can be seen that the semantic map is more accurate and robust in high-dynamic scenes in contrast to low-dynamic scenes.

In the left column of the semantic map, due to people moving slowly or even not moving in low-dynamic scenes, we can still see humanoid black shadows in some places. In our system, we will remove dynamic objects and fill static background objects based on the information of the key frame images before and after. If people never move or move too slowly, it will cause our system to fail to obtain static background information blocked by dynamic objects. However, in the right column of the semantic map, people move faster in a high-dynamic environment, so our system can get the occluded background information in time to build the corresponding part of the map.



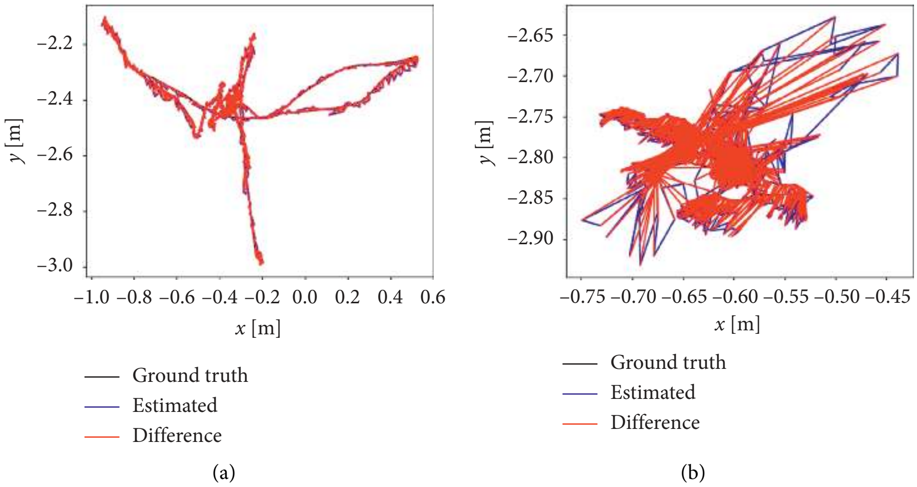
(a)

(b)

(c)

(d)

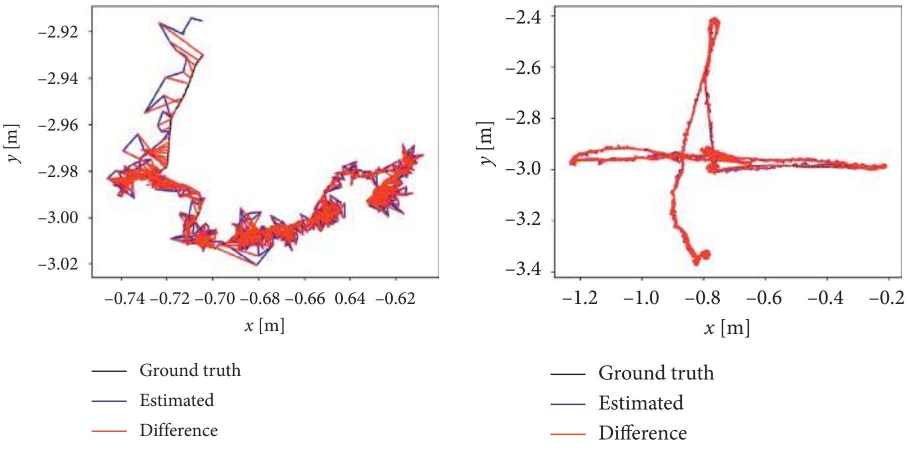
Figure 9: ORB-SLAM2 in low-dynamic scenes. (a) sitting-halfsphere. (b) sitting-rpy. (c) sitting-static. (d) sitting-xyz.



(a)

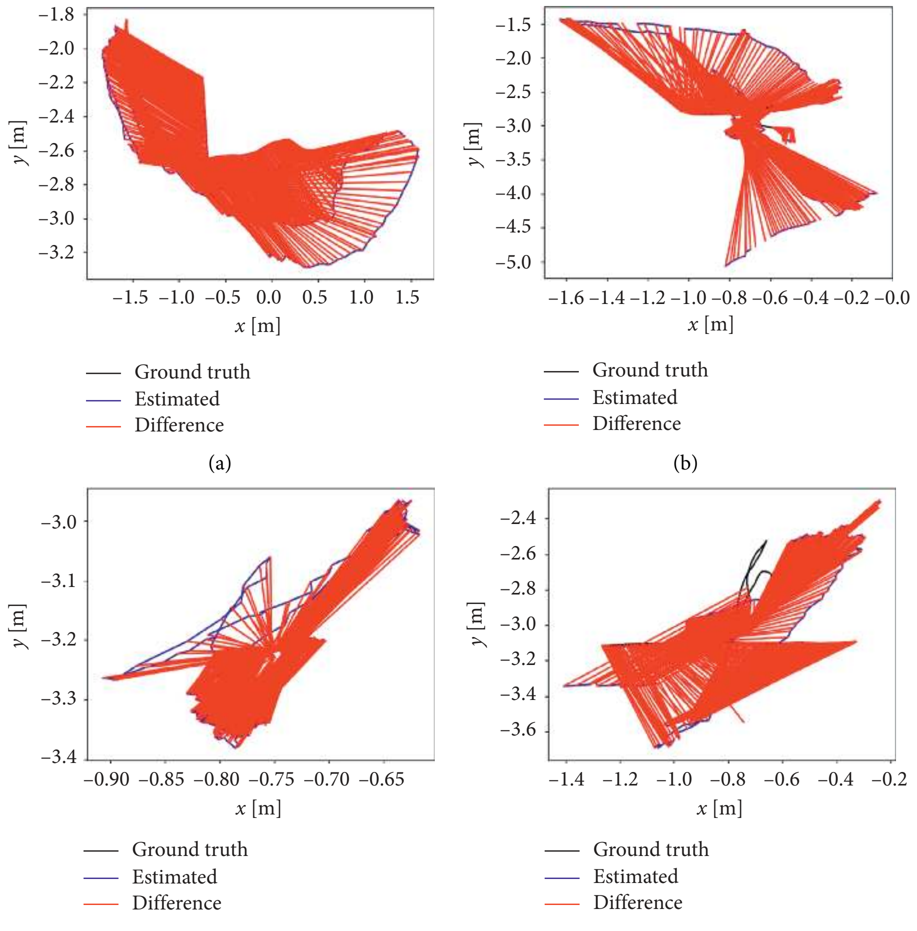
(b)

Figure 10: Continued.



(c) (d)

Figure 10: OFM-SLAM in low-dynamic scenes. (a) sitting-halfsphere. (b) sitting-rpy. (c) sitting-static. (d) sitting-xyz.

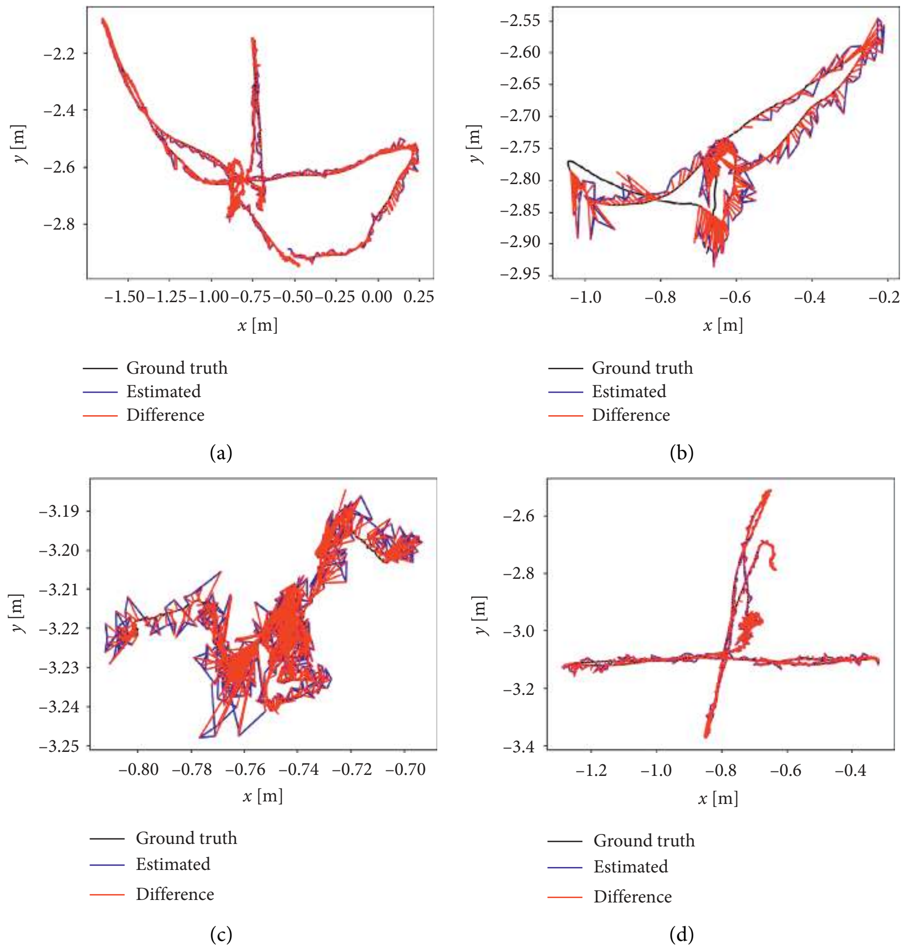


(a) (b)

(c) (d)

Figure 11: ORB-SLAM2 in high-dynamic scenes. (a) walking-halfsphere. (b) walking-rpy. (c) walking-static. (d) walking-xyz.

4.7. Semantic Octree Map Construction in Dynamic Scenes. The semantic octree map is essential for mobile robots to perform advanced and complex tasks, such as obstacle avoidance and navigation or grasping objects. A semantic octree map is generated using the constructed dense point cloud map. The spatial information of the points is stored in the map. Each small square indicates the probability of being occupied. 0 is free, which means passable, and 1 means



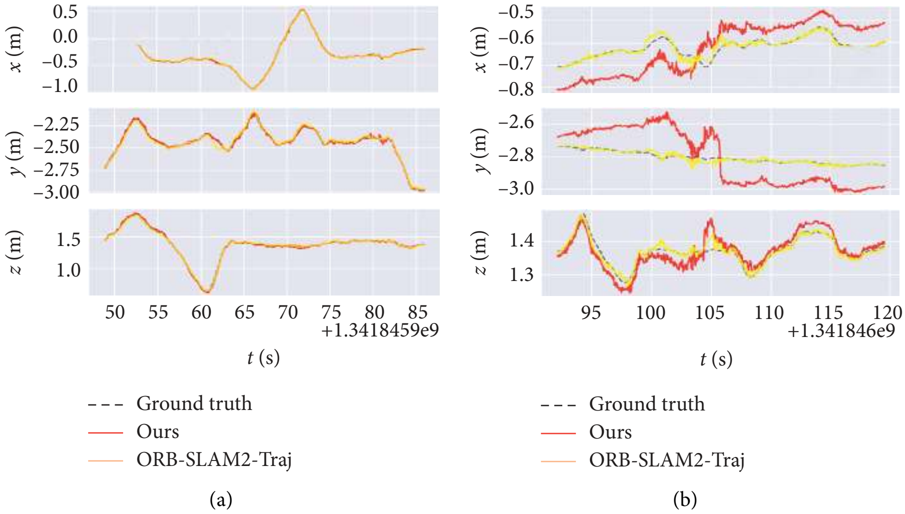
(a)

(b)

(c)

(d)

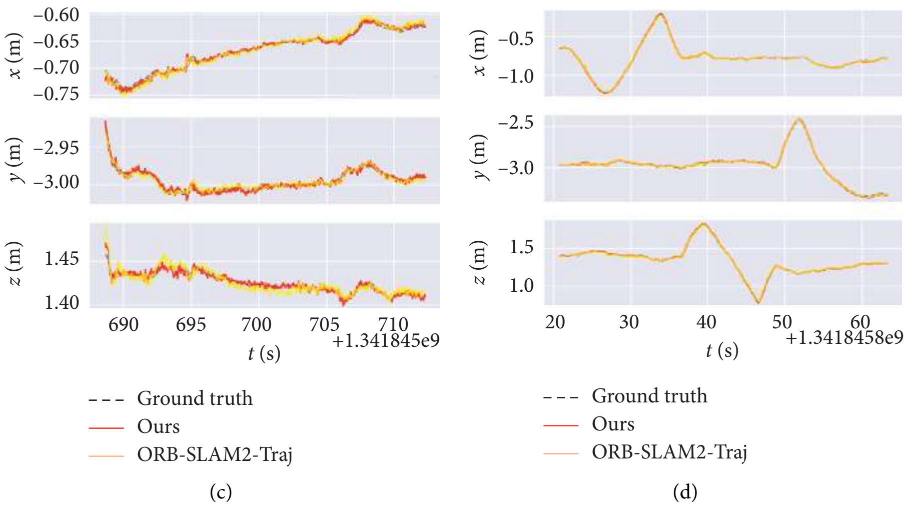
Figure 12: OFM-SLAM in high-dynamic scenes. (a) walking-halfsphere. (b) walking-rpy. (c) walking-static. (d) walking-xyz.



(a)

(b)

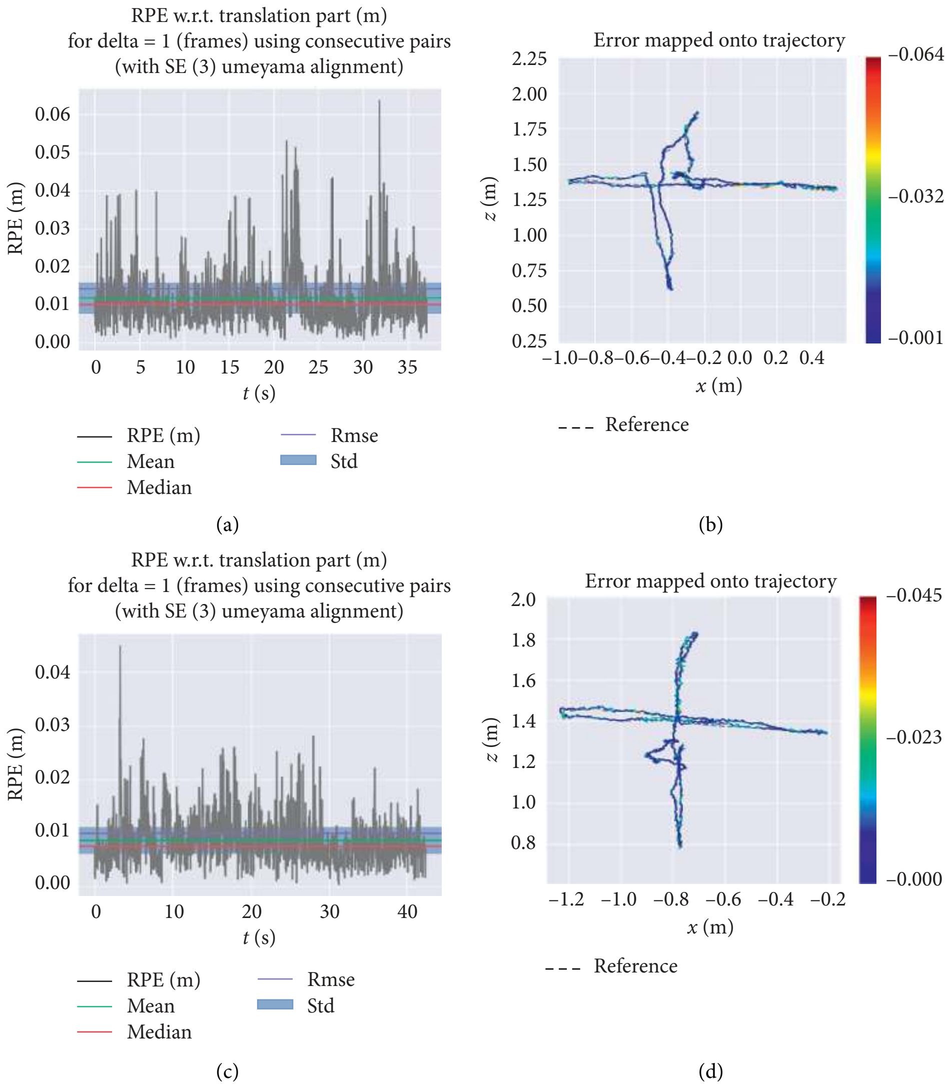
Figure 13: Continued.



(c)

(d)

Figure 13: Trajectory x, y, and z views in low-dynamic scenes. (a) sitting-halfsphere. (b) sitting-rpy. (c) sitting-static. (d) sitting-xyz.



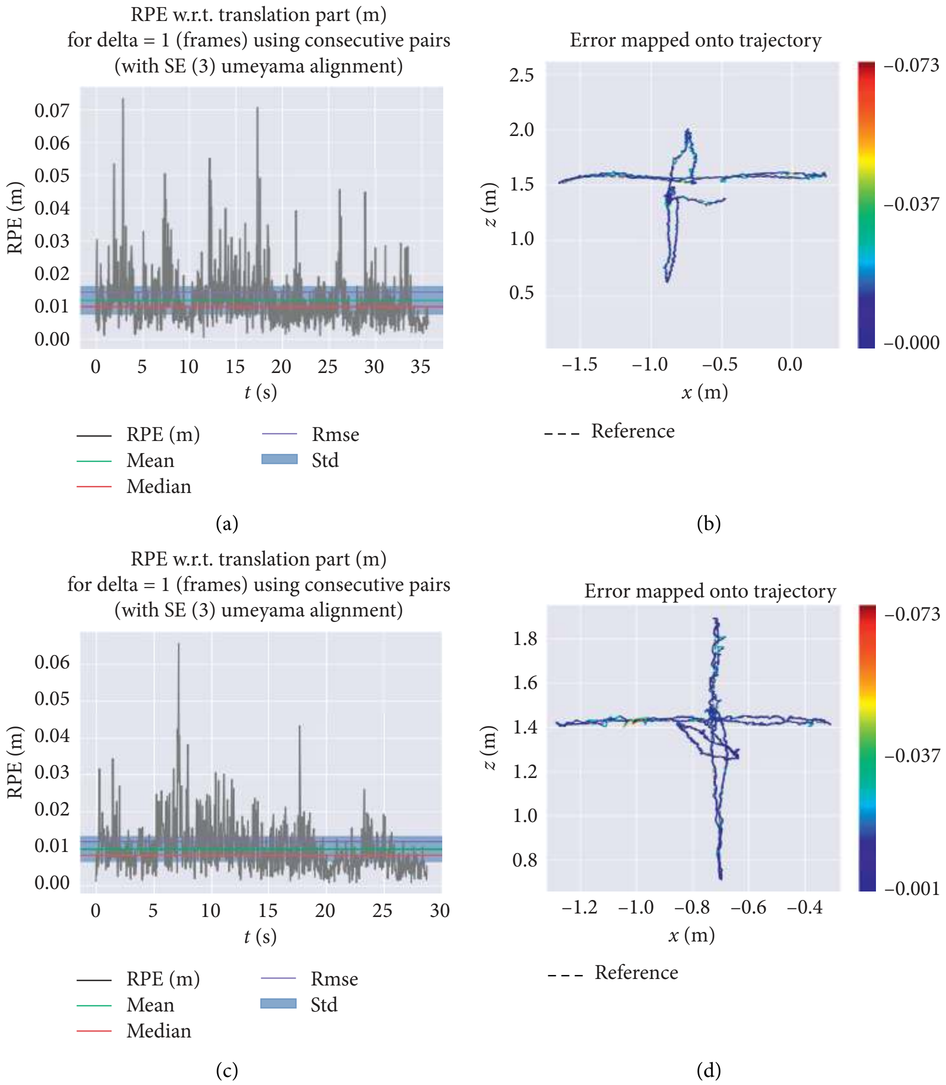
(a)

(b)

(c)

(d)

Figure 14: (a) (b) Relative pose error of the sitting-halfsphere. (c) (d) Relative pose error of the sitting-xyz.



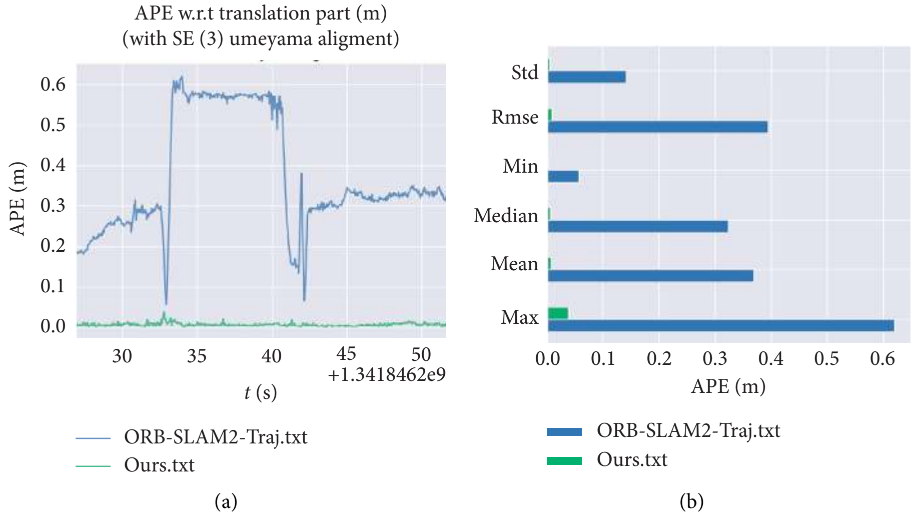
(a)

(b)

(c)

(d)

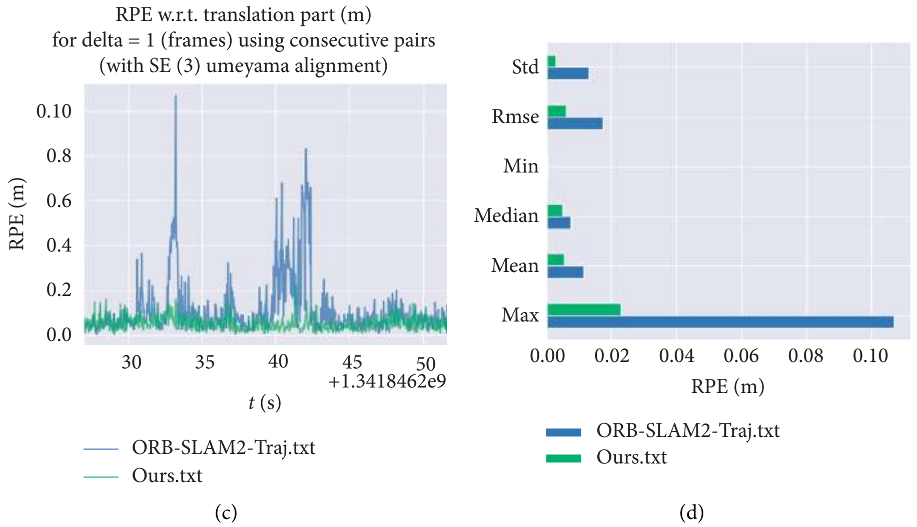
Figure 15: (a) (b) Relative pose error of the walking-halfsphere. (c) (d) Relative pose error of the walking-xyz.



(a)

(b)

Figure 16: Continued.



(c)

(d)

Figure 16: Error estimate in TUM datasets rgbd_dataset_freiburg3_walking_static. (a) (b) Absolute pose error compared with ORB-SLAM2. (c) (d) Relative pose error compared with ORB-SLAM2.

Table 1: ATE and RPE RMSE compared with other systems.

			Error			
Datasets	ATE RMSE (cm)			RPE RMSE (cm)		
	ORB-SLAM2	DS-SLAM	OFM-SLAM	ORB-SLAM2	DS-SLAM	OFM-SLAM
Sitting_static	0.8948	0.650	0.6520	0.5001	0.780	0.5464
Sitting_xyz	0.9033	0.979	1.3789	0.8347	0.863	0.9989
Walking_hs	75.1594	3.030	2.5202	3.7078	2.970	1.4401
Walking_rpy	86.5038	44.420	3.2643	11.2434	15.030	2.3025
Walking_static	39.5077	0.810	0.8085	1.7274	1.020	0.6066
Walking_xyz	59.3589	2.470	1.6777	4.3485	3.330	1.2050

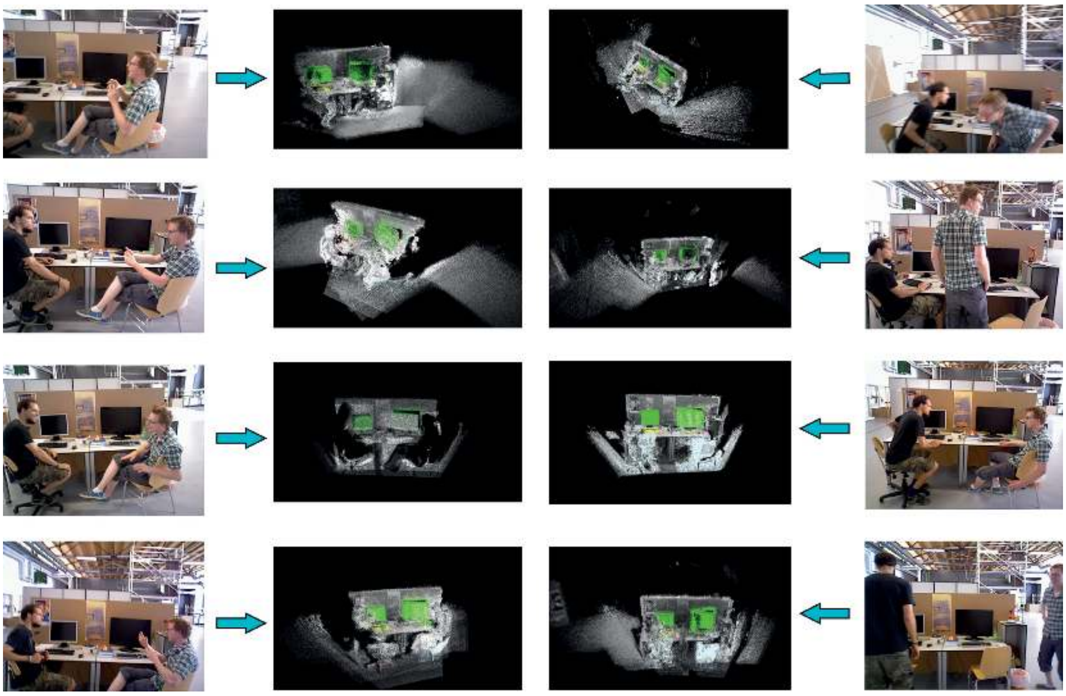


Figure 17: Semantic map construction in low and high dynamic environment.

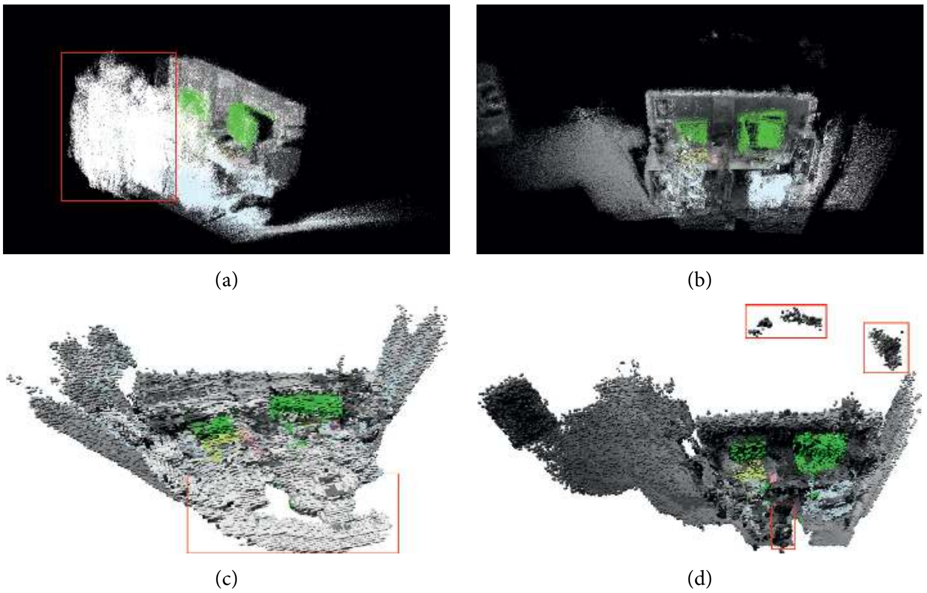


Figure 18: Semantic map construction using TUM datasets (walking-xyz). (a) Semantic map construction contains dynamic objects. (b) Semantic map construction without dynamic objects. (c) Semantic octree map construction contains dynamic objects. (d) Semantic octree map construction without dynamic objects.

occupied, which means impassable. The higher the resolution setting, the smaller the volume of the block. By checking whether each small block is occupied, the mobile robot can navigate with different accuracies. Using the octree to store the map solves the problem that the points in the point cloud map have no volume information and large storage space.

Figure 18 represents semantic map construction in the TUM datasets that have moving objects. We can effectively remove dynamic objects in the scene and reconstruct a more accurate static background semantic map. Due to people not moving or walking slowly, it can be seen that there are still small parts of dynamic objects remaining in the area marked by the red box. However, our system removed most of the dynamic objects successfully in front of the table.

5. Conclusions

To improve the accuracy and robustness of visual SLAM in a dynamic environment and solve the problem that the visual SLAM system generates a large deviation in the pose estimation due to the existence of the moving object in the dynamic scene, we propose a visual SLAM system OFM-SLAM for dynamic indoor environments. OFM-SLAM is able to construct a three-dimensional semantic map that only contains static part in dynamic scenarios. We integrate the optical flow method and the deep learning network MASK-RCNN into OFM-SLAM for dynamic object culling. In high-dynamic indoor environments, OFM-SLAM not only has extremely high positioning accuracy but also has semantic information in dynamic scenes compared to other SLAM systems. However, we cannot generate the sematic map in real time which means it will cost more time to run our proposed system. Therefore, in future works, we can optimize our algorithm to improve real-time performance for application in actual scenarios.

Data Availability

The data used to support the findings of this study are included within this article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The work was supported by Major Project of Hubei Province Technology Innovation, under Grant 2019AAA071, the project of National Natural Science Fund, under Grant nos. 61673304 and 62073249, and the Open Research Foundation of Institute of Robotics and Intelligent Systems (Wuhan University of Science and Technology) under Grant no. F201804.

References

[1] J. McCormac, A. Handa, A. Davison et al., “Semanticfusion: dense 3d semantic mapping with convolutional neural networks,” in *Proceedings of 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4628–4635, IEEE, Marina Bay Sands, Singapore, May 2017.

[2] S. Yang, Y. Huang, and S. Scherer, “Semantic 3D occupancy mapping through efficient high order CRFs,” in *Proceedings of 2017 IEEE/RSJ International Conference on Intelligent Robots*

and Systems (IROS), pp. 590–597, IEEE, Vancouver, Canada, September 2017.

[3] M. Runz, M. Buffier, and L. Agapito, "MaskFusion: real-time recognition, tracking and reconstruction of multiple moving objects," in *Proceedings oof 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 10–20, Munich, Germany, October 2018.

[4] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "MID-fusion: octree-based object-level multi-instance dynamic SLAM," in *Proceedings of 2019 International Conference on Robotics and Automation (ICRA)*, pp. 5231–5237, Montreal, Canada, May 2019.

[5] T. S. Ho, Y. C. Fai, and E. S. L. Ming, "Simultaneous localization and mapping survey based on filtering techniques," in *Proceedings of 2015 10th Asian Control Conference (ASCC)*, pp. 1–6, Kota Kinabalu, Malaysia, May 2015.

[6] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[7] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[8] P. F. Alcantarilla, J. J. Yebes, J. Almazán, and L. M. Bergasa, "On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *Proceedings of 2012 IEEE International Conference on Robotics and Automation*, pp. 1290–1297, Saint Paul, MN, USA, May 2012.

[9] E. Palazzolo, "ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals," in *Proceedings of 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7855–7862, Macao, China, May 2019.

[10] T. Zhang, H. Zhang, Y. Li, Y. Nakamura, and L. Zhang, "FlowFusion: dynamic dense RGB-D SLAM based on optical flow," in *Proceedings of 2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7322–7328, Paris, France, May 2020.

[11] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: large-scale direct monocular slam," in *European Conference on Computer Vision*, pp. 834–849, Springer, Berlin, Germany, 2014.

[12] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "StaticFusion: background reconstruction for dense RGB-D SLAM in dynamic environments," in *Proceedings of 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3849–3856, Brisbane, UK, May 2018.

[13] M. Strecke and J. Stueckler, *EM-fusion: Dynamic Object-Level SLAM with Probabilistic Data Association*, in *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5864–5873, Seoul, Korea (South), October 2019.

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

[15] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Venice, Italy, October 2017.

[17] C. Yu, "DS-SLAM: a semantic visual SLAM towards dynamic environments," in *Proceedings of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1168–1174, Madrid, Spain, October 2018.

[18] W. Dai, Y. Zhang, P. Li, Z. Fang, and S. Scherer, "RGB-D SLAM in dynamic environments using point correlations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, inprint, 2020.

[19] X. Cui, C. Lu, and J. Wang, "3D semantic map construction using improved ORB-SLAM2 for mobile robot in edge computing environment," *IEEE Access*, vol. 8, pp. 67179–67191, 2020.

[20] D. S. Agarwal and K. K. Gupta, "A comparative study of SIFT and SURF algorithms under different object and background conditions," in *Proceedings of 2017 International Conference on Information Technology (ICIT)*, pp. 42–45, Singapore, Singapore, December 2017.

[21] Y. Biadgie and K. Sohn, "Feature Detector Using Adaptive Accelerated Segment Test," in *Proceedings of 2014 International Conference on Information Science & Applications (ICISA)*, pp. 1–4, Seoul, Korea, May 2014.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceedings of 2011 International Conference on Computer Vision*, pp. 2564–2571, Barcelona, Spain, November 2011.

[23] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," *Lecture Notes in Computer Science*, Vol. 6314, Springer, Berlin, Heidelberg, 2010.