Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7600669, 13 pages
https://doi.org/10.1155/2022/7600669

Research Article

MISD-SLAM: Multimodal Semantic SLAM for Dynamic Environments

Yingxuan You,[1] **Peng Wei**,[1] **Jialun Cai**,[1] **Weibo Huang**,[1] **Risheng Kang**,[2] **and Hong Liu**[1]

[1]*Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, Beijing 100871, China*
[2]*Department of Mechanical Engineering, KU Leuven, Leuven 3000, Belgium*

Correspondence should be addressed to Hong Liu; hongliu@pku.edu.cn

Received 19 October 2021; Revised 26 February 2022; Accepted 14 March 2022; Published 5 April 2022

Academic Editor: Chi-Hua Chen

Simultaneous localization and mapping (SLAM) is one of the most essential technologies for mobile robots. Although great progress has been made in the field of SLAM in recent years, there are a number of challenges for SLAM in dynamic environments and high-level semantic scenes. In this paper, we propose a novel multimodal semantic SLAM system (MISD-SLAM), which removes the dynamic objects in the environments and reconstructs the static background with semantic information. MISD-SLAM builds three main processes: instance segmentation, dynamic pixels removal, and semantic 3D map construction. An instance segmentation network is used to provide semantic knowledge of surrounding environments in instance level. The ORB features located on the predefined dynamic objects are removed directly. In this way, MISD-SLAM effectively reduces the impact of dynamic objects to provide precise pose estimation. Then, combining multiview geometry constraint with *K*-means clustering algorithm, our system removes the undefined but moving pixels. Meanwhile, a 3D dense point cloud map with semantic information is reconstructed, which recovers the static background without the corruptions of dynamic objects. Finally, we evaluate MISD-SLAM by comparing to ORB-SLAM3 and the state-of-the-art dynamic SLAM systems in TUM RGB-D datasets and real-world dynamic indoor environments. The results indicate that our method significantly improves the localization accuracy and system robustness, especially in high-dynamic environments.

1. Introduction

Recently, robot technology has been developed rapidly with the wide range applications of the Internet of Things (IoT). Simultaneous localization and mapping (SLAM) is an essential technology for most mobile robots. SLAM system, using the data of its on-board sensors, constructs a map of unknown environment and simultaneously estimates its pose within the map. The on-board sensors carried by the robot to perceive surrounding environments can be divided into two categories, camera and lidar. Visual SLAM, whose main sensor is the camera, has received considerable attention and research efforts in the last few decades. An increasing number of excellent visual SLAM systems have been proposed, such as MonoSLAM [1], PTAM [2], LSD-SLAM [3], and ORB-SLAM1-3 [4–6]. Most of the visual SLAM systems can build 3D geometric map and estimate pose precisely [7, 8] and serve as the baseline for both indoor and outdoor SLAM systems [9]. Moreover, with the development of deep neural networks (DNN) in recent years, many people have begun to integrate visual SLAM with DNN to achieve object detection and semantic segmentation [10–16], which makes the systems able to understand the surrounding environments in semantic level.

Despite the progress of visual SLAM systems, the robustness of SLAM system in dynamic scenes is still a challenge. "Dynamic" means there are dynamic objects in the scenes. According to the motion state, objects can be divided into five cases: (1) immovable objects, such as the wall. (2) Objects with motion properties are moving, such as a moving car or a walking person. (3) Objects with motion properties are in the stationary state, such as a parking car on the

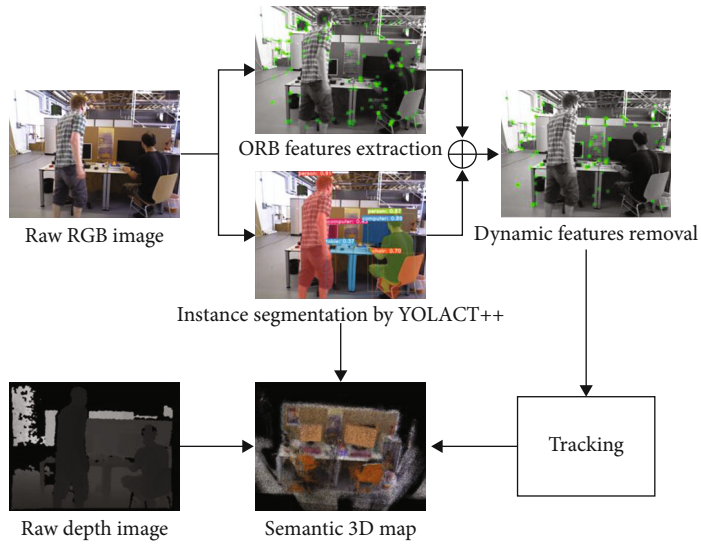


Figure 1: The overview of MISD-SLAM. The raw RGB image is input to perform ORB feature extraction and instance segmentation for semantic information simultaneously. Then, the system removes dynamic features and continues tracking thread to calculate camera pose. Finally, semantic 3D map is built using the data of depth image, camera pose, and semantic information in an independent thread.

side of the road. (4) Objects without motion properties are in the stationary state, such as a static desk. (5) Objects without motion properties are moved, such as a door being opened or closed and tables and chairs being moved. Among these cases, objects with motion properties and objects without motion properties but being moved are defined as dynamic objects in our work. The dynamic objects may lead to the pose estimation inaccurate or failed and make the map corrupted. However, many SLAM works are based on the common assumption that the environments are static. If there are dynamic objects, the motion of objects will be computed into the motion of the camera. Therefore, any dynamic objects in the frame may reduce the accuracy of camera pose estimation or even lead to localization and mapping failure. In this paper, we are focusing on semantic understanding and dynamic robustness in visual SLAM. We propose a semantic visual SLAM system with high performance of accuracy and robustness in dynamic indoor environments, which removes the dynamic objects and reconstructs the static background with semantic information. The overview of the proposed system is shown in Figure 1.

The main contributions of this paper are as follows:

(1) A multimodal semantic visual SLAM system for indoor dynamic environments (MISD-SLAM) is proposed, which significantly increases the accuracy of pose estimation and works more robust in indoor dynamic scenes

(2) A real-time instance segmentation module is proposed to provide semantic knowledge for dynamic objects detection and semantic map reconstruction

(3) A robust tracking strategy is proposed by detecting and removing dynamic features based on the semantic information, which not only reduces the impact of dynamic objects to improve the accuracy of pose evaluation but also remains static features as many as possible to improve the robustness in dynamic environments. Then, the method of multiview geometry constraint removes other dynamic pixels and provides static pixels for map reconstruction without the corruptions of dynamic objects

(4) The high performance of MISD-SLAM in accuracy and robustness is evaluated by the comparison with the state-of-the-art visual SLAM systems on TUM RGB-D datasets [17] and real-world dynamic environments

The rest of this paper is structured as follows: Section 2 discusses an overview of various related work in the fields of visual SLAM with semantic mapping in dynamic environments. Section 3 demonstrates the method of our system in detail. In Section 4, MISD-SLAM is evaluated and compared with the state-of-the-art SLAM systems, DS-SLAM [18], DynaSLAM [19], DetectSLAM [20], SOF-SLAM [21], and SaD-SLAM [22]. And an experiment in real-world environments is carried out to evaluate the performance of the system in real scenes. Finally, Section 5 concludes with a brief conclusion.

2. Related Work

2.1. Semantic Visual SLAM. Traditional visual SLAM mainly focus on geometric information without semantic knowledge of the surrounding environments, which limits the capabilities of robots for high-level tasks. In last few years, with the significant development of deep neural networks (DNN), integrating DNN into visual SLAM to build both geometric and semantic maps has become an important research direction. There are many DNN frameworks. SSD [10] and YOLO [11] can detect objects in boxes. PSPNet [23], SegNet [16], and DeepLab [24–27] are capable to segment objects in pixel level. Moreover, Mask-RCNN [13]

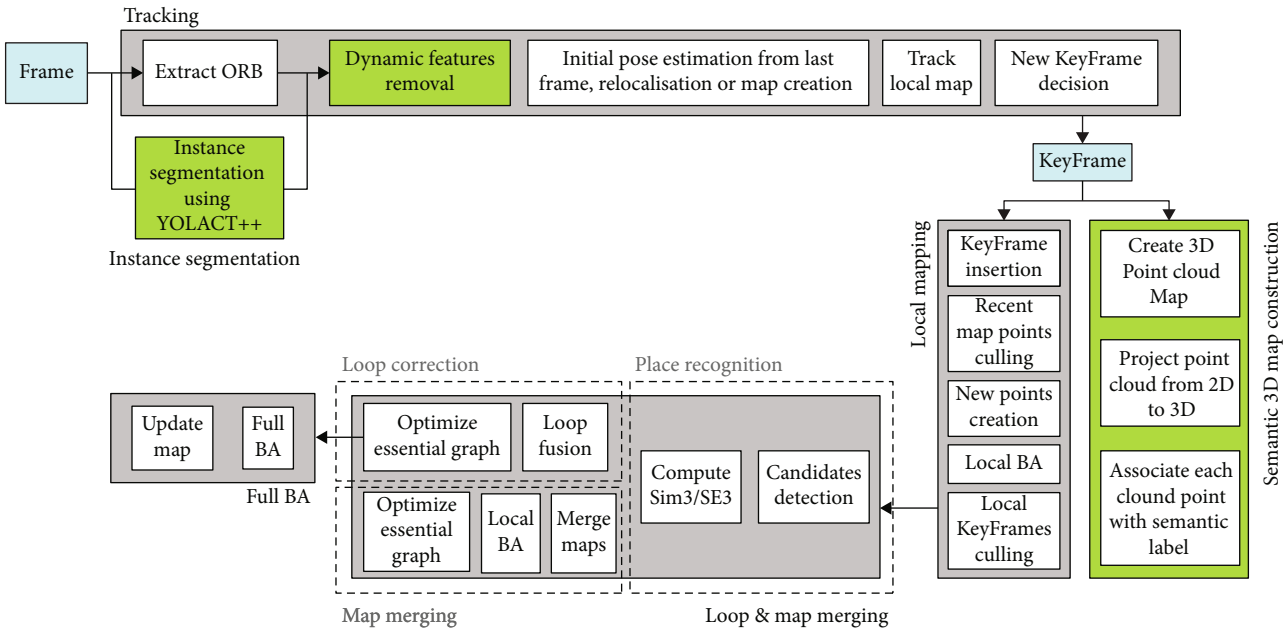


Figure 2: The main architecture of MISD-SLAM. Our system is built on ORB-SLAM3. Modules in green color are new created including instance segmentation, dynamic pixels removal, and semantic 3D map construction, by which MISD-SLAM system could obtain semantic knowledge, reduce the impact of dynamic objects, and build a semantic map.

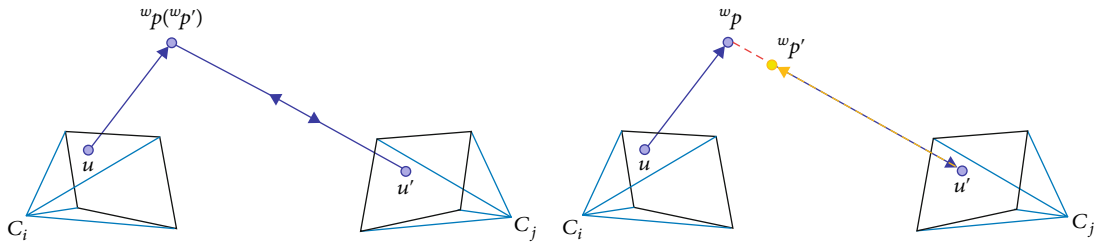
and YOLACT [14, 15] can further distinguish different instances of the same object in pixel-level. The main usages of semantic knowledge obtained from DNN can be divided into two categories, moving dynamic objects and building semantic maps.

2.2. Visual SLAM in Dynamic Scenes. The majority of visual SLAM systems are based on the common assumption that the environments are static, while the real world is changeable and dynamic. In recent years, several dynamic SLAM methods have been proposed. DS-SLAM [18] combines dynamic object detection and moving consistency check to remove the feature points located on dynamic objects. But the categories it can detect is only 20, which limits its application in complex scenarios. DynaSLAM [19] integrates Mask-RCNN [13] and multiview geometry for motion segmentation, which performs well in dynamic environments. However, it removes all potentially moving objects, such as cars parked on the side of the road, which may lead to too few feature points and impact pose estimation. VDO-SLAM [28] maximizes the number of feature points on dynamic objects using the method of dense optical flow and gets impressive results. But it is complex for real-time operation. SaD-SLAM [22] proposes a RGB-D SLAM system based on ORB-SLAM2 [5], which uses epipolar constraint of feature points in two adjacent frames to detect the static feature points in dynamic environments. But the semantic segmentation has to be processed offline that limits its application in real world. PoseFusion [29] combines human detection method, OpenPose [30], and the dense RGB-D SLAM framework, ElasticFusion [31]. However, it is limited to human detection and may not work well if the human is incomplete in the input image. StaticFusion [32] proposes a method of static and dynamic segmentation to reconstruct the background structure and applies K-means clustering algorithm to reduce the computational complexity. But it will fail if the initial images have more than 30% moving objects. Co-Fusion [33] is a model-based method, which combines object segmentation method and the dense reconstruction framework of ElasticFusion [31]. However, the map of static environment is required to be reconstructed as the precondition for tracking, segmentation, and fusion of dynamic objects, which limits its application. If two or more objects move together, they are represented by the same model until they separate. FlowFusion [34] is a flow-based method, which proposes an optical flow residual base dynamic segmentation and dense RGB-D SLAM method. It can distinguish dynamic and static clusters by setting the thresholds for high and low residuals. But it is not sensitive to the slight motions and may fail in very fast motions.

In this paper, we propose a multimodal semantic visual SLAM system for dynamic environments (MISD-SLAM) based on ORB-SLAM3 [6], which can reduce the impact of dynamic objects to evaluate accurate poses and reconstruct a semantic 3D dense map of static background. Different from the prior works, MISD-SLAM combines multiview geometry constraint method and K-means clustering algorithm to reduce the impact of dynamic pixels for map reconstruction of static background. The experiments in both public datasets and real-world environments demonstrate that our method has high performance of accuracy and robustness in dynamic indoor scenes.

3. System Overview

In this section, we will present the technical details about MISD-SLAM. Figure 2 presents the architecture of the system. We build MISD-SLAM on ORB-SLAM3 [6], which is one of the most novel feature-based visual SLAM systems



(a) The situation of static point (b) The situation of dynamic point

Figure 3: Multiview geometry constraint. C_i and C_j are two consecutive frames, u and u' are corresponding image pixels, and ^wp and $^wp'$ are their back-projection points.

that performs impressively in many datasets as well as real-world scenarios. MISD-SLAM builds three new processes:

(i) Instance segmentation, based on a pretrained network, detects and segments different instances to provide semantic knowledge of surrounding environments

(ii) Dynamic pixel removal removes the ORB features located in predefined dynamic objects which are detected by instance segmentation network then combines multiview geometry constraint with K -means clustering algorithm to remove the undefined but moving pixels to improve the accuracy and robustness in changing environments

(iii) Semantic 3D map construction combines semantic knowledge obtained from instance segmentation network with geometric structure to construct a semantic 3D dense point cloud map in global

3.1. Instance Segmentation. Under the demand of dynamic pixel detection and semantic mapping, we adopt a deep learning-based network to provide instance segmentation and semantic labels in pixel-level. MISD-SLAM utilizes the network of YOLACT++ [15] that is pretrained on MS COCO datasets [35] and can segment 80 classes. The semantic knowledge of the surrounding environments has two purposes. On the one hand, it serves as a prior information for dynamic features removal. We predefined person as a dynamic object in indoor environments. System then removes the ORB features located on the predefined objects, which improves the accuracy of pose evaluation in tracking thread and remains static ORB features as many as possible to improve the robustness in dynamic environments. On the other hand, the semantic knowledge of the pixel is integrated into corresponding 3D point to reconstruct semantic dense point cloud map in the thread of map reconstruction.

3.2. Dynamic Pixels Removal. Although prior semantic information can filter out the predefined dynamic objects in images, there may be some missing detections due to image blurring, incomplete observation, and the moving of not predefined objects. Therefore, the method of multiview geometry constraint is applied to detect the real motion of the remaining image pixels.

As shown in Figure 3, a method of multiview geometry constraint [36], which is based on the relationship of corresponding points in two consecutive frames, can be used to detect whether a pixel is static or dynamic. For current frame i and the last frame j, firstly, the pixel u in frame i is back-projected to 3D world coordinate as a point $^w\mathbf{p}$ using the information of camera pose $^w_i\mathbf{T}$ of current frame and its depth value z from the depth image:

$$^w\mathbf{p} = {}^w_i\mathbf{T}\pi^{-1}(\mathbf{u}, z), \tag{1}$$

where π^{-1} denotes the function of back-projection which depends on the camera types.

Then, the 3D point in world coordinate $^w\mathbf{p}$ is projected to the image pixel u' of the last frame j:

$$\mathbf{u}' = \pi\left(\left({}^w_j\mathbf{T}\right)^{-1} {}^w\mathbf{p}\right), \tag{2}$$

where π denotes the function of perspective projection and $^w_j\mathbf{T}$ is the camera pose of frame j estimated in tracking thread.

Furthermore, the 3D point $^w\mathbf{p}'$ in the world coordinate of the pixel u' in the last frame j can be rebuilt as:

$$^w\mathbf{p}' = {}^w_j\mathbf{T}\pi^{-1}\left(\mathbf{u}', z'\right), \tag{3}$$

where z' denotes the depth value of pixel u' in frame j.

If the point is static in both current frame i and the last frame j, as shown in Figure 3(a), the points in 3D world coordinate $^w\mathbf{p}$ and $^w\mathbf{p}'$ is pretty close to each other or even overlap. Otherwise, if the point is dynamic, the distance d between the two points $^w\mathbf{p}$ and $^w\mathbf{p}'$ in 3D world coordinate is large, as shown in Figure 3(b). Therefore, a threshold d_{th} is set to judge the dynamic points and static points. Due to the depth error increases with distance, the threshold d_{th} is set to linearly grow with the depth z:

$$d_{\text{th}} = d_{\text{base}} + kz, \tag{4}$$

where d_{base} is the base value of distance and k is the scaling factor of depth z. We set $d_{\text{base}} = 0.2$ and $k = 0.025$.

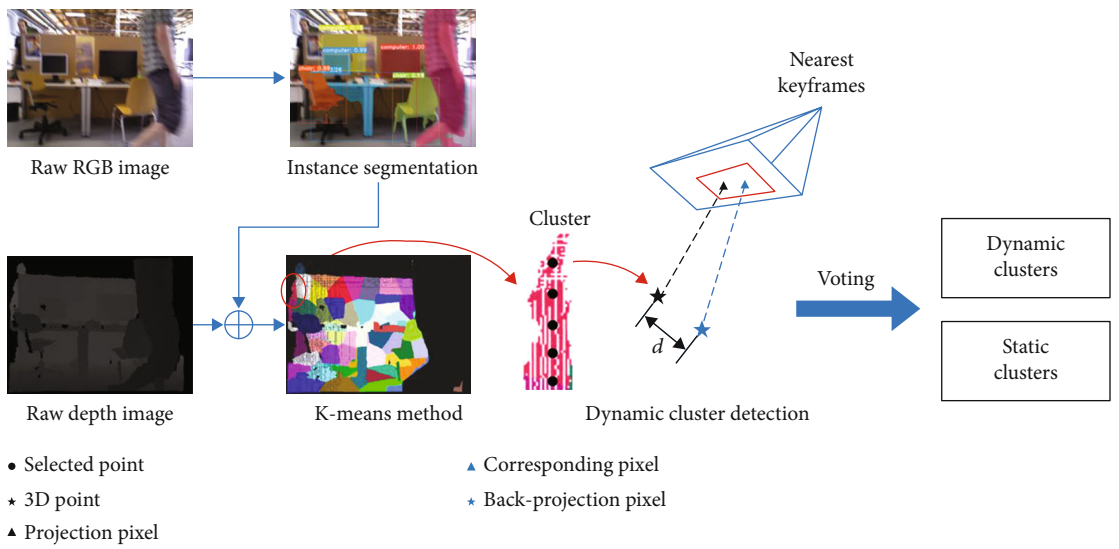
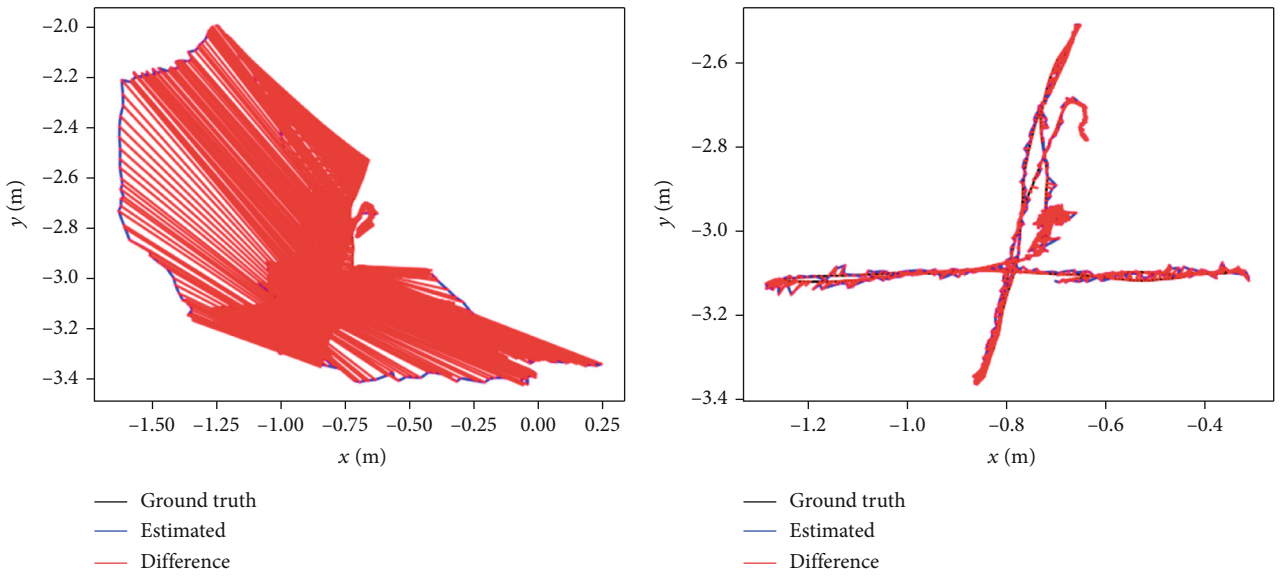


Figure 4: The flowchart of dynamic pixel removal. The raw RGB image is firstly utilized to instance segmentation to remove the pixels located on dynamic objects. Then, the remaining pixels would be combined with the depth image to generate clusters using K-means method. Under multiview geometry constraint, points can be divided into dynamic and static. And the motion state of each cluster is determined by voting method.



(a) Trajectory from ORB_SLAM3

(b) Trajectory from MISD_SLAM

Figure 5: The trajectory difference between two systems and ground truth in high-dynamic sequence *walking_xyz*. The blue line represents the trajectory estimated by the respective SLAM system, and the red line is the difference between estimated trajectory and the ground truth.

If the distance d between ${}^{w}\mathbf{p}$ and ${}^{w}\mathbf{p}'$ is larger than the threshold d_{th}, then the image pixel of current frame is determined as dynamic. Otherwise, the pixel is static.

Furthermore, in order to reduce the calculation and running time, K-means clustering algorithm and voting method are proposed to detect dynamic pixels, as shown in Figure 4. Combing with the depth image and the camera pose estimated from the tacking thread, the remaining pixels of the RGB image are back-projected to 3D points in the world coordinate to create a point cloud. The 3D point cloud are divided into k clusters by K-means clustering algorithm, where k is calculated by the number of remaining pixels/2000. In each cluster, 100 points are randomly selected. If there are not 100 points in one cluster, then all of the points are selected. The motion property of each selected point is determined by the method of multiview geometry constraint method. Then, voting method is used to determine the motion property of each cluster according to the majority motion properties of its selected points:

$$\text{Motion}_i = \begin{cases} \text{dynamic}, & \text{num}_{\text{dynamic}} > \text{num}_{\text{static}}, \\ \text{static}, & \text{num}_{\text{dynamic}} <= \text{num}_{\text{static}}, \end{cases} \tag{5}$$

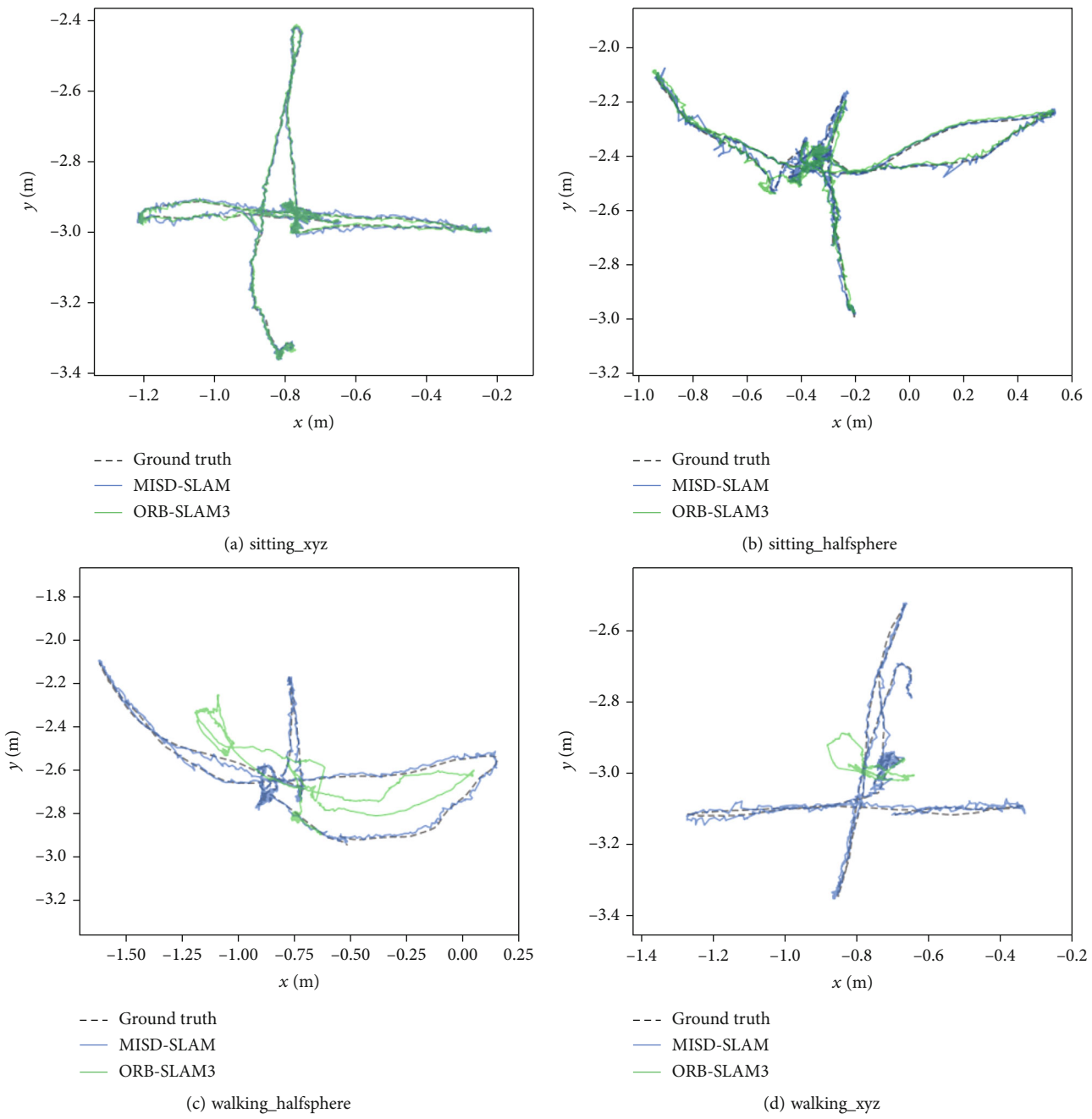


Figure 6: Trajectory comparison in dynamic sequences. Sitting series is in low-dynamic, and walking series is in high-dynamic. The green line is the trajectory of ORB-SLAM3, the blue line is the trajectory of MISD-SLAM, and the black dotted line is the ground truth.

where Motion_i denotes the motion property of the i_{th} cluster, $\text{num}_{\text{dynamic}}/\text{num}_{\text{static}}$ denotes the number of dynamic/static points among the selected points in i_{th} cluster. If the number of dynamic points is larger than the number of static points, then the cluster is dynamic; otherwise, the cluster is static. Finally, the corresponding image pixels of the points in dynamic clusters are determined as dynamic.

Combining the predefined dynamic objects in instance segmentation and the dynamic pixels in motion detection, the total dynamic pixels are obtained. In the following process of map reconstruction, the dynamic pixels are removed, and the static pixels are used to reconstruct the map of static background.

3.3. Semantic 3D Map Construction. The thread focuses on reconstruct the static background of keyframes. The map type is 3D point cloud with semantic labels. As shown in Figure 1, the inputs of map reconstruction thread are the RGB-D image pair, semantic image, and camera pose estimated in the tracking thread. After the process of dynamic pixels removal, the total dynamic regions in RGB image are obtained, including the predefined dynamic objects

Table 1: Results of absolute trajectory error (ATE) and improvements of MISD-SLAM compared to ORB-SLAM3 in dynamic sequences.

Sequence	ORB-SLAM3 [6]			MISD-SLAM (ours)			Improvements		
	RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.
Sitting_static	0.0067	0.0055	0.0037	0.0059	0.0052	0.0028	11.94%	5.45%	24.32%
Walking_static	0.0248	0.0199	0.0148	0.0091	0.0079	0.0036	63.31%	60.30%	68.92%
Walking_xyz	0.2895	0.2620	0.1232	0.0129	0.0112	0.0063	95.54%	95.73%	94.89%
Walking_rpy	0.1655	0.1396	0.0889	0.1252	0.0904	0.0866	24.35%	35.24%	2.59%
Walking_halfsphere	0.3305	0.2977	0.1434	0.0168	0.0149	0.0077	94.92%	94.99%	94.63%

Table 2: Results of absolute trajectory error (ATE) and improvements of processed PL-SVO compared to original PL-SVO in dynamic sequences.

Sequence	PL-SVO [37]			Processed PL-SVO			Improvements		
	RMSE	Mean	S.D.	RMSE	Mean	S.D.	RMSE	Mean	S.D.
Sitting_static	0.0084	0.0077	0.0033	0.0064	0.0056	0.0031	23.81%	27.27%	6.06%
Walking_static	0.0075	0.0065	0.0037	0.0057	0.0048	0.0030	24.00%	26.15%	18.92%
Walking_xyz	0.2811	0.2409	0.1363	0.0230	0.0199	0.0116	90.96%	90.74%	91.49%
Walking_rpy	0.1770	0.1541	0.0872	0.0357	0.0275	0.0228	79.83%	82.15%	73.85%
Walking_halfsphere	0.1149	0.1112	0.0289	0.0401	0.0320	0.0242	65.10%	71.22%	16.26%

detected in instance segmentation and the dynamic pixels determined by their motion properties. To reduce the impact of the region edges, the dynamic region edges in the image are expanded 20 pixels. The depth values of the expanded dynamic regions in the depth image are set to zero. Except the image pixels located in the expanded dynamic regions, the other image pixels are static pixels. However, the static pixels are not all reconstructed to the map, because the points of some static pixels may have existed in the map, which are called redundant points. To avoid redundant points, before back-projecting the static image pixels into 3D points in the local point cloud, we perform an operation of pixel-point association. As for each point in the global point cloud, it is projected to a pixel in current image based on the camera pose and camera internal parameters, so that we can get the position (x, y) and depth value z of the projected pixel. Then, according to the position (x, y) of the projected pixel, we obtain four pixels around it in current image. If the minimum depth difference between the four image pixels and the projected pixel is smaller than the threshold (0.02 in our work), which indicates that the image pixel has been reconstructed in the map, then the image pixel of the minimum depth difference will not be used to be back-projected into the 3D space. We set the depth value of this pixel to zero. Then, the image pixels with depth value in the range of $z_{\min}$ to $z_{\max}$ are back-projected into 3D points to generate the local point cloud of current keyframe, based on the camera internal parameters and the camera pose that is estimated in the tracking thread. We set the threshold values $z_{\min} = 0.2$ and $z_{\max} = 8.0$. In this way, the dynamic pixels and the redundant pixels in the image are not reconstructed in the point cloud because their depth values were set to zero. For semantic labels, a color attribute of point is applied to represent its category according to the semantic image. For example, the point classified to chair is labeled in orange color, and the keyboard is purple color. As for the point without semantic information, its color attribute is set the corresponding pixel value in RGB image. Then, the local static point cloud with semantic labels of current keyframe is fused into the global point cloud in the world coordinate, which reconstructs the map of static background incrementally.

4. Experiments

In this section, we demonstrate our MISD-SLAM system in public TUM RGB-D datasets [17] and real-world scenes to evaluate its performance of accuracy and robustness in dynamic environments. First, MISD-SLAM system is compared with original ORB-SLAM3 [6] to verify the improvement of performance. Then, we replace ORB-SLAM3 to another backbone, PL-SVO [37] to validate the effectiveness of the proposed method. In addition, our MISD-SLAM system is compared with the state-of-the-art SLAM systems in dynamic environments. Besides, the semantic 3D dense point cloud maps and the time performance are presented. Finally, an experiment in real-world environments is carried out to evaluate the performance of the system in real scenes. All the experiments run on a computer with Intel E5-2683 CPU and Nvidia GTX 1080 GPU. The GPU is only used for instance segmentation.

4.1. Experiments in TUM RGB-D Datasets. The TUM RGB-D datasets [17] provide video sequences of indoor scenes recorded by Microsoft Kinect at the frame rate of 30 Hz. The datasets include RGB images and depth images with 640×480 resolution, as well as ground truth trajectories. We select the sequences of dynamic scenes to evaluate our MISD-SLAM system. In the sequences of *sitting* series, there are two people sitting on chairs in front of a desk and talking with each other. These sequences represent low-dynamic environments. In the sequences of *walking* series, people

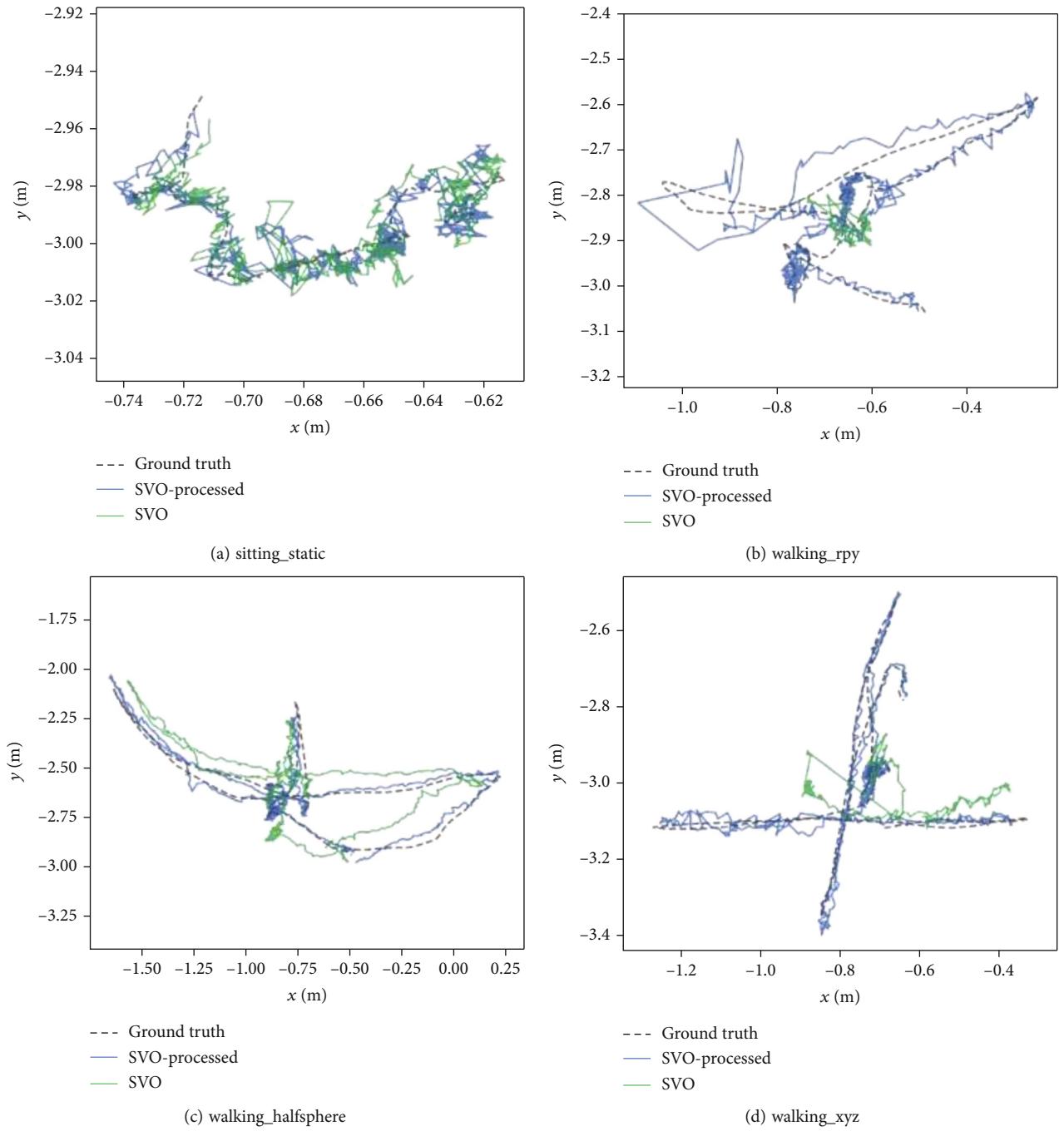


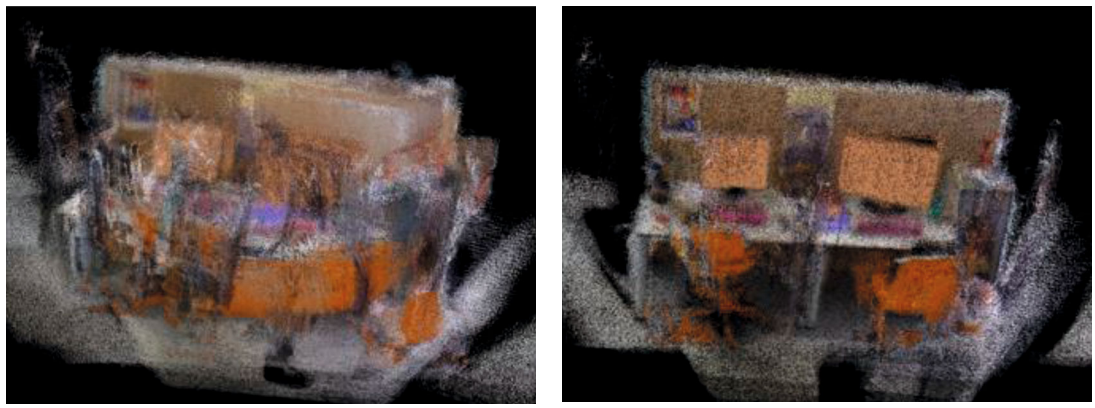
Figure 7: Trajectory comparison of original PL-SVO [37] and processed PL-SVO in dynamic sequences. Sitting series is in low-dynamic, and walking series is in high-dynamic. The green line is the trajectory of original PL-SVO, the blue line is the trajectory of processed PL-SVO, and the black dotted line is the ground truth.

walk most of the time. These sequences are in high-dynamic, which would seriously impact the accuracy and robustness of ordinary SLAM systems. ORB-SLAM3 [6] is a state-of-the-art visual system and serves as the backbone of our MISD-SLAM, so we firstly compare these two systems and make a quantitative evaluation.

We compare ORB-SLAM3 [6] and MISD-SLAM in dynamic sequences of TUM RGB-D datasets [17], which are composed of four patterns of camera ego-motions including keeping still in one place (static), moving along three directions (*xyz*), rotating along the principle axes (*rpy*) and moving on a small half sphere (halfsphere). The comparison results are presented in Figure 5, where the two camera trajectories obtained from these systems in sequence *walking_xyz* are, respectively, plotted with the ground truth trajectory. The difference between trajectories of ORB-SLAM3 and ground truth is apparent in Figure 5(a), while in Figure 5(b), the two trajectories of MISD-SLAM and ground truth are very close, which shows the robustness and accuracy of our system. The reason is

Table 3: Comparison results of absolute trajectory RMSE (m) against the state-of-the-art dynamic SLAM systems.

Sequence	DS-SLAM [18]	DynaSLAM [19]	Detect-SLAM [20]	SOF-SLAM [21]	SaD-SLAM [22]	MISD-SLAM (ours)
Sitting_static	0.0065	—	—	0.010	0.0060	0.0059
Walking_static	0.0081	0.006	—	0.007	0.0166	0.0091
Walking_xyz	0.0247	0.015	0.0241	0.018	0.0167	0.0129
Walking_rpy	0.4442	0.035	0.2959	0.027	0.0318	0.1252
Walking_halfsphere	0.0303	0.025	0.0514	0.029	0.0257	0.0168



(a) map without dynamic points removal (b) map with dynamic points removal

Figure 8: The semantic 3D dense point cloud maps built by MISD-SLAM. (a) is the map without dynamic point removal, and (b) is the map with dynamic point removal.

Table 4: Time evaluation of MISD-SLAM in dynamic sequences of TUM RGB-D datasets (ms). The results of each row are the running time of corresponding modules in each sequence. The last row is the average time of above sequences in corresponding modules.

Sequence	ORB feature extraction (ms)	Instance segmentation (ms)	Multiview geometry (ms)	Semantic map construction (ms)
Sitting_static	12.981	36.465	144.085	121.388
Walking_static	14.575	36.109	205.484	296.667
Walking_xyz	14.146	36.150	180.533	338.903
Walking_rpy	12.345	36.640	200.591	274.928
Walking_ halfsphere	12.484	36.536	161.401	244.241
Average time	13.306	36.380	178.419	255.225

that the dynamic features detected by ORB-SLAM3 are assumed as static features to estimate the trajectory, which leads the wrong results. While in MISD-SLAM system, these dynamic features are removed and the impact on trajectory prediction are reduced significantly.

Furthermore, we compare ORB-SLAM3 [6] and our MISD-SLAM system in other dynamic sequences. The results are shown in Figure 6, where the trajectory of ORB-SLAM3 [6] is green line, the MISD-SLAM is blue line, and the ground truth is black dotted line. In Figures 6(a) and 6(b), the three trajectories are very close which indicates the accuracies of both systems are high in low-dynamic sequences. However, in Figure 6(c) and 6(d), the trajectories of ORB-SLAM3 [6] are deformed seriously, while the trajectories of MISD-SLAM are still close to the ground truth. We perform a quantitative evaluation of the two SLAM systems in different sequences in Table 1, using the values of root mean squared error (RMSE), mean error, and standard deviation (S.D.) of absolute trajectory. It can be seen that due to the removal of dynamic feature points, MISD-SLAM significantly reduce the impact of dynamic objects and effectively

Table 5: Comparison of the time performance to other methods. The total time includes the time of feature extraction, semantic segmentation, and dynamic objects removal but without the time of map construction.

Methods	Total time (ms)
DS-SLAM [18]	76.46
DynaSLAM [19]	286.47
Detect-SLAM [20]	340.00
MISD-SLAM (ours)	228.11

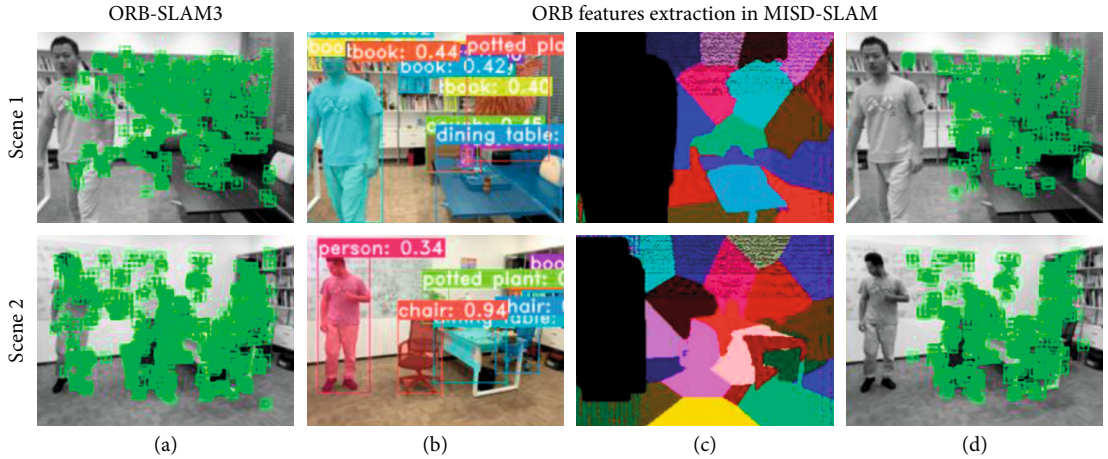


Figure 9: Experiments in real-world environments and comparison of ORB features extraction results between ORB-SLAM3 [6] and MISD-SLAM. From left to right, the first column (a) is the ORB features extraction results in ORB-SLAM3 [6]. (b)–(d) are the processes of MISD-SLAM, which are, respectively, the instance segmentation, clustering results, and ORB features extraction results after dynamic features removal in MISD-SLAM.

estimate the correct trajectory, which demonstrates high robustness and correctness of the system.

To validate the effectiveness and application of the proposed method, we replace ORB-SLAM3 [6] to another backbone, PL-SVO [37], and evaluate its performance of trajectory accuracy. PL-SVO [37] is a monocular visual odometry algorithm combining point features and line features with RGB image inputs, which is appropriate for this experiment of backbone replacement. We processed PL-SVO [37] by adding the modules of instance segmentation and dynamic features removal to remove the point features and line features located in the dynamic objects. We compare original PL-SVO and processed PL-SVO in the dynamic sequences of TUM RGB-D datasets. The quantitative results are shown in Table 2. It can be seen that the performance of accuracy is improved in all of the dynamic sequences. Especially, the improvement in sequence *walking_xyz* is over 90%. Figure 7 shows the qualitative results by visualizing their trajectories, where the green line is the trajectory of original PL-SVO, the blue line is the trajectory of processed PL-SVO, and the black dotted line is the ground truth. Figures 6 and 7 indicate that the original ORB-SLAM3 and the original PL-SVO are impacted by the dynamic objects in the environments, and their trajectory accuracy is low compared to the ground truth, especially in high-dynamic sequences. Processed by the proposed method, the dynamic features are removed, and the accuracy performance is improved significantly, which validates the effectiveness and application of the proposed method. Although we provide the experiment using another backbone, our goal is not to explore the effect of different backbones. We focus our attention on the following experiments based on MISD-SLAM with the backbone of ORB-SLAM3.

4.2. Comparison with Other Visual SLAM Systems. In this part, we adopt dynamic sequences of TUM RGB-D datasets [17] to compare our MISD-SLAM against the five state-of-the-art visual SLAM systems, DS-SLAM [18], DynaSLAM [19], Detect-SLAM [20], SOF-SLAM [21], and SaD-SLAM [22] which have been proposed for dynamic environments and semantic tasks in last three years. The results are shown in Table 3, and the results of the five SLAM systems come from published papers. The results of our MISD-SLAM gained after running on the datasets for five times and taking the average values. The sequence *sitting_static* is in low-dynamic, and *walking* series is in high-dynamic. MISD-SLAM performs better in sequences of *sitting_static*, *walking_xyz*, and *walking_halfsphere*.

The experiment results indicate the high performance of our system. MISD-SLAM removes dynamic features according to the result of instance segmentation, then remaining pixels with potential movement are detected and removed through multiview geometry constraint method. After these two steps, the moving image pixels are deleted, and the impact of dynamic objects is reduced.

However, if there are too few features, the system may estimate wrong pose, or even track failure. Compared with the other four systems, MISD-SLAM reduces the impact of dynamic objects to improve the accuracy of pose evaluation and remains static features as many as possible to improve the robustness in dynamic environments, which improves the performance of accuracy and robustness.

4.3. Semantic 3D Maps. This part presents the semantic 3D dense point cloud maps built by MISD-SLAM system. Figure 8 compares two maps built in the high-dynamic sequence of TUM RGB-D datasets [17], *walking_xyz*. Figure 8(a) is reconstructed without dynamic pixels removal. It can be seen that the dynamic persons reduce the accuracy of camera pose estimation, which make the things misplaced. Besides, the moving persons are modeled in the map. Therefore, Figure 8(a) is corrupted and difficult to use. Figure 8(b) is constructed after dynamic pixel removal, which reduces the influence of dynamic objects, so that the static background can be reconstructed with the accurate camera pose. The map with dynamic pixels removal is

Figure 10: The semantic 3D point cloud maps of real environments. The figures in (a, b) are the semantic 3D point cloud maps. The figures in (c, d) are the color images of experimental scenes.

capable to provide a precise and semantic map of static background in dynamic indoor scenes.

4.4. Evaluation of Time Performance. Time performance is another important indicator to evaluate the proposed method. We evaluate the time performance in four major modules: ORB feature extraction, instance segmentation, dynamic pixels removal, and semantic map construction. The results shown in Table 4 are the running time of corresponding modules in each sequence and the average time of all the sequences. The time performances of ORB feature extraction and instance segmentation are achieved in real time. The most time-consuming module is the semantic map construction module. But it only operates in keyframes, which is selected from input frames so the number of keyframes is less than the number of input frames. And the semantic map construction module operates in parallel with other modules. Therefore, the time cost of semantic map construction module affects little to the whole process. The multiview geometry constraint method slowdowns the process compared to other modules in MISD-SLAM. However, compared to the modules with similar function, MISD-SLAM has higher time performance than DynaSLAM [19] (333.68 ms in sequence *walking_halfsphere* and 235.98 ms in sequence *walking_rpy*) and Detect-SLAM [20] (310 ms), due to its reduction of computational complexity by K-means clustering algorithm and voting method.

Furthermore, Table 5 shows the comparison of the time performance to other methods, including DS-SLAM [18], DynaSLAM [19], and Detect-SLAM [20]. Given that some methods do not provide the time of map construction in their papers and the large variability of map construction time in different hardware conditions of computing, rendering, and displaying, it is more fair to compare the total time except map construction. Table 5 lists the total time except map construction to represent the time performance, which is the sum of the average time of feature extraction, semantic segmentation, and dynamic objects removal. Among these methods, DS-SLAM [18] is optimized for real time, and the other three methods are not optimized for real time. It can be seen that MISD-SLAM has higher time performance among the not optimized methods and achieves comparable performance to the optimized method. The comparison results of time performance indicates that the proposed method plays an important role in reducing the computational complexity and achieves high time performance.

4.5. Experiments in Real-World Environments. Experiments in real-world environments are carried out to evaluate the performance of the MISD-SLAM system in real scenes.

The images of RGB and depth are captured by iPad Pro with 320×240 resolution. The experiment scene is an office, where a person is walking around, and the camera is doing translational motion.

In Figure 9, the images from left to right are raw ORB feature extraction image of ORB-SLAM3 [6] and images in our system including instance segmentation image, clustering image, and ORB features extraction image after dynamic feature removal. It can be seen that the moving people are detected in the instance segmentation image. The dynamic features located in the dynamic objects are removed. The removal of dynamic features reduces the influence of dynamic objects for better camera pose estimation, and semantic mapping. Furthermore, after K-means clustering and multiview geometry constraint, dynamic pixels in the scene are removed significantly.

Figure 10 shows the 3D semantic map constructed by MISDSLAM system in these two real scenes. Because the dynamic features are removed, the camera pose can be correctly estimated. The system back-projects the static image pixels into 3D space based on the camera pose to build a static point cloud map of the real scene. 2D map with semantic information generated by 3D semantic point cloud map can be applied in navigation and planning tasks [18, 38].

5. Conclusions

In this paper, we propose a novel multimodal semantic SLAM system (MISD-SLAM), which could perform robustly in dynamic environments and build a semantic 3D point cloud map. MISD-SLAM builds three main processes: instance segmentation, dynamic pixel removal, and semantic 3D map construction. An instance segmentation network [15] is introduced to provide semantic knowledge of surrounding environments. The ORB features located on the predefined dynamic objects are removed directly. In this way, MISD-SLAM effectively reduces the impact of dynamic objects to provide precise pose estimation. Then, combining multiview geometry constraint with K-means clustering algorithm, our system removes the undefined but moving pixels. Meanwhile, a 3D dense point cloud map with semantic information is reconstructed. Moreover, experiments are carried out on challenging sequences of TUM RGB-D datasets [17] as well as the real-world scenes to evaluate the performance of MISD-SLAM. Compared to original ORB-SLAM3 [6] and the state-of-the-art SLAM systems, the results indicate that our method significantly improves the localization accuracy and system robustness, especially in high-dynamic environments.

However, there exists some limitations in MISD-SLAM. First, the process of dynamic objects is not flexible enough, because the objects may be static in some frames and be dynamic in other frames. Second, the depth range of the RGB-D camera is restricted, which limits its application in larger scenes. In the future, the developments of MISD-SLAM will focus on optimizing the strategy of dynamic objects removal in the reconstructed map and improving the real-time performance. Furthermore, we will adopt inertial measurement unit (IMU) to expand the scope of application in larger environments.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 62073004) and Science and Technology Plan of Shenzhen (no. JCYJ20190808182209321).

References

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[2] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, 2007.

[3] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: large-scale direct monocular SLAM," in *Europeon Conference on Computer Vision (ECCV)*, pp. 834–849, 2014.

[4] 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.

[5] 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.

[6] 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 Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[7] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and M achine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[8] M. Labbe and F. Michaud, "Online global loop closure detection for large-scale multi-session graph-based SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2661–2666, 2014.

[9] Z. Zhao, Y. Mao, Y. Ding, P. Ren, and N. Zheng, "Visual-based semantic SLAM with landmarks for large-scale outdoor environment," in *China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, pp. 149–154, 2019.

[10] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Europeon Conference on Computer Vision (ECCV)*, pp. 21–37, 2016.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2961–2969, 2017.

[14] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: real-time instance segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 9157–9166, 2019.

[15] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT++: better real-time instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 1108–1121, 2022.

[16] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[17] 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 (IROS)*, pp. 573–580, 2012.

[18] C. Yu, Z. Liu, X. Liu et al., "DS-SLAM: a semantic visual SLAM towards dynamic environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1168–1174, 2018.

[19] 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.

[20] F. Zhong, S. Wang, Z. Zhang, C. Chen, and Y. Wang, "Detect-SLAM: making object detection and SLAM mutually beneficial," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1001–1010, 2018.

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

[22] X. Yuan and S. Chen, "SAD-SLAM: a visual SLAM based on semantic and depth information," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4930–4935, 2020.

[23] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2881–2890, 2017.

[24] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, http://arxiv.org/abs/1412.7062.

[25] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[26] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, http://arxiv.org/abs/1706.05587.

[27] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Europeon Conference on Computer Vision (ECCV)*, pp. 801–818, 2018.

[28] J. Zhang, M. Henein, R. Mahony, and V. Ila, "VDO-SLAM: a visual dynamic object-aware SLAM system," 2020, http://arxiv.org/abs/2005.11052.

[29] T. Zhang and Y. Nakamura, "PoseFusion: dense RGB-D SLAM in dynamic human environments," in *International Symposium on Experimental Robotics*, pp. 772–780, Springer, Cham, 2020.

[30] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7291–7299, 2017.

[31] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: real-time dense SLAM and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[32] 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)*, pp. 3849–3856, 2018.

[33] M. Rünz and L. Agapito, "Co-fusion: real-time segmentation, tracking and fusion of multiple objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4471–4478, 2017.

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

[35] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: common objects in context," in *Europeon Conference on Computer Vision (ECCV)*, pp. 740–755, Springer, Cham, 2014.

[36] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision, 2nd Edn. Cambridge University Press*, Cambridge University Press, 2000.

[37] R. Gomez-Ojeda, J. Briales, and J. Gonzalez-Jimenez, "PL-SVO: semi-direct monocular visual odometry by combining points and line segments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4211–4216, 2016.

[38] N. Sünderhauf, F. Dayoub, S. McMahon et al., "Place categorization and semantic mapping on a mobile robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5729–5736, 2016.