Moving Object Segmentation and Detection for Robust RGBD-SLAM in Dynamic Environments

Wanfang Xie, Peter Xiaoping Liu, *Fellow, IEEE*, and Minhua Zheng, *Member, IEEE*

***Abstract*— Localization accuracy is a fundamental requirement for Simultaneous Localization and Mapping (SLAM) systems. Traditional visual SLAM (vSLAM) schemes are usually based upon the assumption of static environments, so they do not perform well in dynamic environments. While a number of vSLAM frameworks have been reported for dynamic environments, the localization accuracy is usually unsatisfactory. In this article, we present a novel motion detection and segmentation method using Red Green Blue-Depth (RGB-D) data to improve the localization accuracy of feature-based RGB-D SLAM in dynamic environments. To overcome the problem due to under-segmentation generated by the semantic segmentation network, a mask inpainting method is developed to ensure the completeness of object segmentation. In the meantime, an optical flow-based motion detection method is proposed to detect dynamic objects from moving cameras, allowing robust detection by removing irrelevant information. Experiments performed on the public Technical University of Munich (TUM) RGB-D data set show that the presented scheme outperforms the state-of-art RGB-D SLAM systems in terms of trajectory accuracy, improving the localization accuracy of RGB-D SLAM in dynamic environments.**

***Index Terms*— Dynamic environments, localization accuracy, motion detection, object segmentation, Red Green Blue-Depth (RGB-D)-simultaneous localization and mapping (SLAM).**

I. Introduction

THE camera is an important sensor in many applications [1], especially in visual Simultaneous Localization and Mapping (vSLAM) systems, and these cameras include monocular, stereo, or Red Green Blue-Depth (RGB-D) ones. Many practical vSLAM systems rely on cameras to perceive and understand the environment [2]–[6]. The RGB-D SLAM is, thus, one specific type of vSLAM system, for which the RGB-D camera is used as the main sensor. In recent years, the RGB-D camera has attracted much attention since it can easily capture the depth and RGB images [7]. In this study, we focus on how to improve the localization accuracy of feature-based RGB-D SLAM systems with dynamic environments.

Most traditional vSLAM systems rely on the static environments assumption, which is not the case for many real-life applications. In dynamic environments, the feature points on moving objects could influence the performance of vSLAM.

Accurate segmentation of moving objects is the prerequisite for removing all feature points on moving objects. The moving object segmentation approaches in vSLAM systems can be divided into two categories: one relies on the motion differences between dynamic and static objects, while the other combines the semantic segmentation network with the motion differences.

First, moving object segmentation relies on the motion differences between dynamic and static objects. Fan *et al.* [8] proposed two different constraints according to the change of depth information of dynamic feature points to determine dynamic object regions. Sun *et al.* [9] proposed a tracking and segmentation strategy to identify dynamic objects, and then, in the work [10], they proposed a motion removal approach. In highly dynamic environments, this approach achieves satisfactory performance. Wang and Huang [11] integrated the fundamental matrix to solve the oversegmentation and undersegmentation problems existing in the method [12]. However, these segmentation approaches for moving objects will degrade under certain conditions, such as a large camera motion between consecutive frames.

In order to improve the effectiveness of moving object segmentation, some researchers combine semantic information with the motion differences to segment moving objects. Bescos *et al.* [13] employed both neutral network and multiview geometry algorithm to segment moving objects. Nevertheless, the multiview geometry algorithm is time-consuming. Yu *et al.* [14] used SegNet to segment objects, followed by using an epipolar line constraint to detect moving points from all feature points. Their method reduced the influence of dynamic objects on RGB-D SLAM. Cui and Ma [15] proposed a novel detection approach combining the semantic information and epipolar geometry constraint to detect dynamic objects. They improved the performance of the original ORB-SLAM system in dynamic environments. However, due to the moving camera, the noises of optical flow were not concerned. Wang *et al.* [16] incorporated the Fully Convolutional Instance-aware Semantic Segmentation (FCIS) into the RGB-D SLAM system. They proposed a segmentation method for moving objects according to the reprojection error

Manuscript received June 23, 2020; accepted September 16, 2020. Date of publication September 25, 2020; date of current version November 25, 2020. This work was supported in part by the National Science Foundation of China under Grant 61773051, Grant 61761166011, and Grant 51705016. The Associate Editor coordinating the review process was Dr. Emanuele Zappa. *(Corresponding authors: Wanfang Xie; Peter Xiaoping Liu.)*

Wanfang Xie and Minhua Zheng are with the School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: 18121291@bjtu.edu.cn).

Peter Xiaoping Liu is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: xpliu@sce.carleton.ca).

Digital Object Identifier 10.1109/TIM.2020.3026803

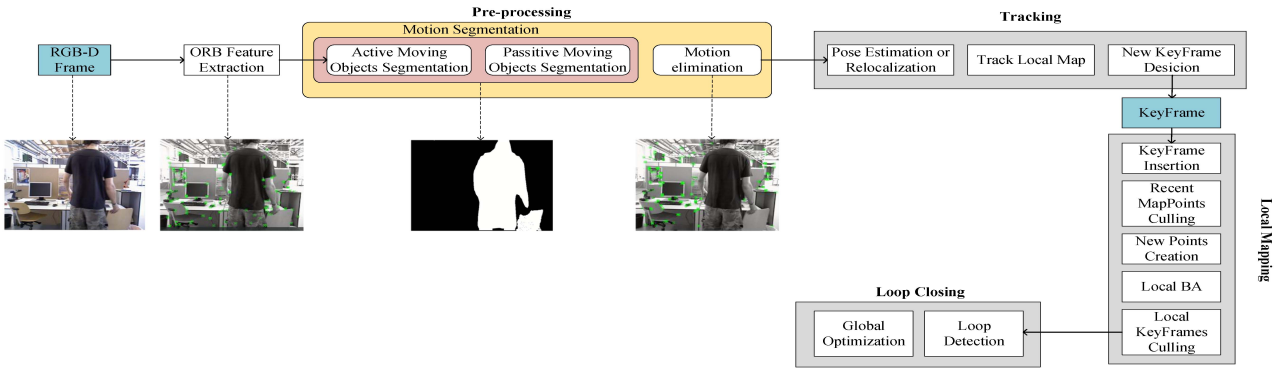


Fig. 1. Overview of our RGB-D SLAM system that is based on ORB-SLAM2. Our main work here is in the preprocessing stage.

and the semantic segmentation. Xu *et al.* [17] proposed a novel RGB-D SLAM system focusing on 3-D reconstruction. Their moving object segmentation approach integrated a set of factors about geometric, photometric, and semantic information. Zhong *et al.* [18] used single-shot multibox detector (SSD) to segment moving objects in keyframes. For other frames, they proposed a propagation model to update the moving probability of feature points. This way makes it possible to assure the real-time requirements of the RGB-D SLAM system. However, in the aforementioned studies, the undersegmentation problem in the semantic segmentation network degrades the localization accuracy of vSLAM because some feature points from moving objects cannot be removed. The problem of undersegmentation is most likely to occur in humans, as a human is a pervasive moving object. Therefore, we need to assure the completeness of segmentation to a human when the semantic segmentation networks are not reliable.

In this article, the moving objects in dynamic environments are divided into two categories, i.e., active moving objects, such as pedestrians, and passive moving objects, such as human-pushed chairs. Our moving object segmentation method coalesces in the RGB-D SLAM front end as a preprocessing stage. The RGB-D data provided by the Technical University of Munich (TUM) [19], [20] and the Imperial College London and National University of Ireland Maynooth (ICL-NUIM) [21] are used for experiments. The main contributions of this work can be summarized as follows.

1) A novel moving object segmentation scheme is developed and integrated with the ORB-SLAM2 to improve the localization accuracy in dynamic environments.
2) In the segmentation of active moving objects (humans), we propose a mask inpainting algorithm depending on the depth information to assist semantic information. It solves the undersegmentation problem in the semantic segmentation network.
3) In the segmentation of passive moving objects, we improve the motion detection method based on the Lucas–Kanade (LK) optical flow [22] for moving objects under dynamic environments, making the motion detection using a moving camera more reliable.

The rest of this article is organized as follows. In Section II, we state the specific problems in dynamic environments and give the details of our proposed method. In Section III, we provide experimental details and results. In addition, we discuss the strong and weak points of our method. In Section IV, we make our conclusions and perspectives for future directions. Acknowledgments are included in Section V.

II. Proposed Approach

A. Problem Statement

Generally, the vSLAM problem can be described by observation model

$$z_{k,j} = h(l_j, x_k) + v_{k,j} \tag{1}$$

where x_k denotes camera pose in world coordinate at time k, and at this time, one landmark is observed, and l_j denotes its position. $z_{k,j}$ represents the measurement of camera to a landmark j at time k. Let $v_{k,j}$ denote the measurement noise. The function $h(\cdot)$ represents the process of converting the position of a landmark in world coordinate into the 2-D pixel coordinate.

We optimize camera poses and landmark locations by minimizing the sum of reprojection error

$$X^*, L^* = \arg\min_{X,L} \sum_{k=1}^{n} \sum_{j=1}^{m} \left\| z_{k,j} - h(l_j, x_k) \right\|^2 \tag{2}$$

where $X \triangleq \{x_k\}_{k=1}^{n}$ and $L \triangleq \{l_j\}_{j=1}^{m}$ represent the camera poses and the positions of landmark.

In static environments, we can solve this least square problem correctly by iteration. However, in dynamic environments, moving landmarks cause mismatches with measurements; this is a problem of wrong data association. To solve the problem in dynamic environments, we need to remove the feature points from moving objects.

B. Approach Overview

The overview of our RGB-D SLAM system can be seen in Fig. 1. First, we perform ORB feature extraction on each RGB image. Both static and dynamic feature points can be extracted. There are two steps in preprocessing, i.e., motion segmentation and motion elimination. The former is extremely important in our work. We use our methods to segment active and passive moving objects consecutively. Then, the dynamic

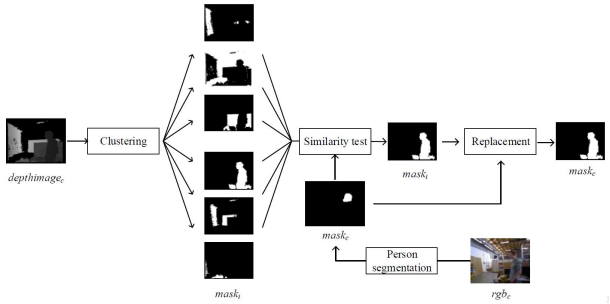


Fig. 2. Workflow of our mask inpainting framework.

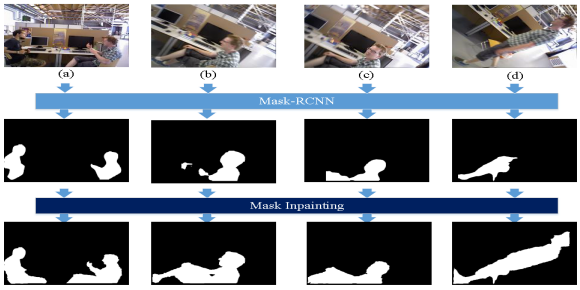


Fig. 3. Our mask inpainting results.

feature points can be eliminated according to the segmentation. The subsequent processing in our framework is the same as ORB-SLAM2.

C. Segmentation for Active Moving Objects (Humans)

In our segmentation for active moving objects, we use MaskRCNN [23] to segment just humans. Since humans are the most common moving objects in the environments, the completeness of human segmentation has a great impact on the accuracy of pose estimation. In many circumstances, MaskRCNN is able to generate a complete segmentation of a human, but, in other cases, there can be an undersegmentation problem. In order to overcome this issue, we proposed the mask inpainting method that is demonstrated in Fig. 2.

1) Mask Inpainting: A better cluster effect can be obtained when we use depth images rather than RGB images because objects are easier to be recognized by the depth values [9]. The current depth image depthimage_c is segmented into K clusters by using the K-means clustering algorithm [24]. A complete mask of human can be acquired by setting K to six in our work. We have verified it by using a large number of images of different scenes. The binarization method is employed to represent each cluster mask_i, such that the foreground pixel and the background pixel values are set to 255 and 0, respectively.

In order to find the human cluster from the six different options, we design an image similarity test that refers to the perceptual hashing [25]. The similarity S_i between each cluster mask_i and the segmentation mask_c can be calculated by the following:

$$S_i = \frac{\text{Area}_c \cap \text{Area}_i}{\text{Area}_c}, \quad i \in \{1, 2, 3, 4, 5, 6\} \tag{3}$$

where Area_c and Area_i represent the foreground in mask_c and each cluster mask_i, respectively. The mask_c is produced by MaskRCNN on each frame. Since the human cluster has the highest S_i, we then replace the undersegmentated mask_c with the mask_i according to the highest S_i.

It is not necessary to inpaint each mask because many masks of human are complete, and the clustering process is time-consuming. In our algorithm, the decision on whether to inpaint is made according to p, which is the ratio of number_l to number_c. number_c and number_l represent the number of pixels that make up the foreground of mask_c in the current and last frame, respectively

$$p = \text{number}_l / \text{number}_c. \tag{4}$$

The ratio p will be stable when the complete mask of humans continues to be generated from the last frame to the current frame, whereas the ratio will change a lot when an incomplete mask appears in the current frame. A range of ratio p is made to control inpainting depending on this change. The inpainting condition can be described as follows:

$$p \in \alpha, \quad \text{Inpainting} \tag{5}$$

$$p \notin \alpha, \quad \text{No Inpainting} \tag{6}$$

where α represents 1.3–3.3, and if p is in range α, then there is a mask_c that should be inpainted.

TABLE I

COMPARISONS OF IOU BETWEEN THE UNDERSEGMENTATED RESULTS AND THE MASK INPAINTING RESULTS

Image	IoU	
	under-segmentation	mask inpainting
(a)	0.53	**0.83**
(b)	0.46	**0.87**
(c)	0.57	**0.89**
(d)	0.31	**0.92**

Fig. 3 demonstrates that our mask inpainting method is able to replace the incomplete masks with the inpainting masks. The first row contains four RGB images in the TUM data set. The images in the second row are the undersegmentated results of MaskRCNN. The third row shows the inpainting results of our method. In addition, according to Table I, the comparisons between the Intersection over Union (IoU) of the undersegmentated results and the ground truth, as well as the IoU of the inpainting results and the ground truth, show the superiority of our mask inpainting method. We take the human area in the four RGB images as the ground truth.

D. Segmentation for Passive Moving Objects

In this section, we present our segmentation method for passive moving objects, such as the human-pushed chair and table. Since we cannot predict what they are, it is impossible to segment them directly with MaskRCNN. Therefore, motion detection is a prerequisite for the segmentation of passive moving objects.

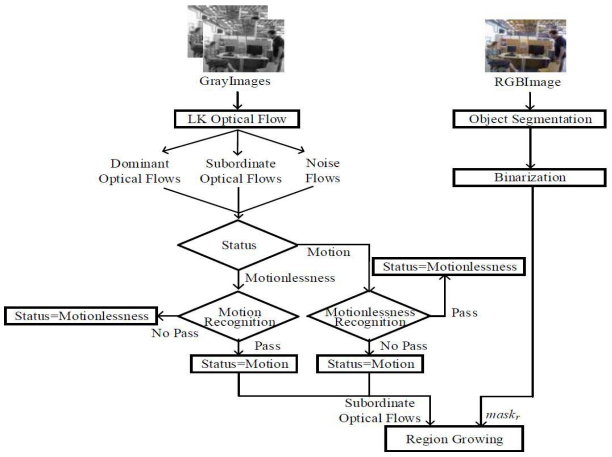


Fig. 4. Flowchart of our detection and segmentation for passive moving objects.

Our motion detection method is based on the LK optical flow. We divide the global optical flows in each frame into different categories, including dominant optical flows [26], subordinate optical flows, and noises. The motion of the camera causes dominant optical flows that have similar motion information with the camera and are the majority in every frame. Most of the optical flows located at static objects are dominant optical flows. The motion of moving objects generates subordinate optical flows that have similar motion information with moving objects, but the subordinate optical flows are less than the dominate optical flows. The third optical flow type is the noises that are the key factor degrading the moving object detection.

Fig. 4 shows how passive moving objects are detected and segmented. Our motion detection method, which is shown on the left-hand side of the figure, aims to identify the subordinate optical flow points from global optical flow points. The subordinate optical flow points are used to further segment objects from the segmentation mask_r generated by MaskRCNN with the region growing algorithm [27].

Our motion detection method includes mainly motion recognition and motionlessness recognition modules. Either of the two modules will be implemented once on each frame. According to the status that is set to the motionlessness in the initial frame, we perform a motion recognition module that is used to detect passive moving objects. It removes the optical flows caused by the camera motion and the noises with three tests: dominant flow test, point feature descriptor (PFD) test, and motion consistency test. Every test removes optical flow points from global optical flow points. If some subordinate optical flow points pass the motion detection module, we set the status to motion, which will activate the motionlessness recognition module in the next frame. This module determines whether the object is still moving based on the comparison of the motion information between subordinate and dominant optical flows. If the two motion information is close, the status is set to motionlessness. Otherwise, the status will still be in motion in the next frame. More details about our motion detection method can be seen in the following.

1) Removing Dominate Optical Flows in Motion Recognition: The optical flow points in each frame are divided into four areas evenly. The coordinates of every optical flow point are determined by the following rules: the difference of the horizontal axis between every two optical flow points is 20 pixels, and the difference of the vertical axis is 15 pixels. We need to remove those points located at the human since the human has been segmented first. For the remaining points, we calculate the corresponding points in the next frame using the LK optical flow. The following formula expresses the optical flow matchings between two frames:

$$\gamma_p^t = \gamma_p^{t+1} - V \tag{7}$$

where γ_p^t is the location of every optical flow point p in current frame t, γ_p^{t+1} is the corresponding location of γ_p^t in next frame $t+1$, and V stands for the optical flow vector.

The motion information of every optical flow point needs to be calculated in the current frame, which is the magnitude ρ_p^t and the direction d_p^t.

For an optical flow vector $V = (\Delta x, \Delta y)$, its magnitude ρ and angle θ are calculated as follows:

$$\rho = \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{8}$$

$$\theta = \arctan(\Delta x / \Delta y). \tag{9}$$

In our method, the direction d is an integer between 1 and 12. We equally divide 360° into 12 parts and use one integer to represent angle θ.

After we acquire the magnitude and the direction of every optical flow point, the dominate flow test is used to remove the dominant optical flow points. We use all magnitude and direction data to acquire the one that is the majority in each frame by a statistic method. Depending on the value of the majority magnitude and direction, the dominant optical flow points can be removed.

After removing the dominant optical flow, we need to distinguish the subordinate optical flows from noises.

2) Reducing the Impact of Noises in Motion Recognition: We design a PFD that includes the motion and pixel information of optical flow points. For the motion information, we have calculated the magnitude ρ and direction d in advance. For pixel information, we can directly acquire the gray value g and the coordinate γ_x, γ_y of the optical flow points.

Two strategies are used to reduce the impact of noises. The first one is PFD test dealing with the optical flow points Q that have passed the dominate flow test in the current frame. The PFD test obeys the rule that the optical flow points on passive moving object (i.e., subordinate optical flows) have similar PFD parameters. In contrast, there should not be any optical flow point that shares similar PFD parameters with noise points. According to this property, we can remove some noise points. We set a similarity threshold for each parameter in the PFD test

$$\left|\rho_q^t - \rho_{Q-q}^t\right| \leq \tau_1 \tag{10}$$

$$\left|d_q^t - d_{Q-q}^t\right| \leq \tau_2 \tag{11}$$

$$\left|g_q^t - g_{Q-q}^t\right| \leq \tau_3 \tag{12}$$

$$\left|\gamma_{x_q}^t - \gamma_{x_{Q-q}}^t\right| \leq \tau_4 \tag{13}$$

$$\left|\gamma_{y_q}^t - \gamma_{y_{Q-q}}^t\right| \leq \tau_5 \tag{14}$$

where the subscript q indicates any optical flow point that has passed the dominate flow test, $Q - q$ indicates the rest of optical flow points, and τ_{1-5} mean five thresholds that are fixed, i.e., 1, 1, 10, 60, and 45 in our work.

The noise points with different PFD parameter values from other optical flow points can be removed by our PFD test. However, there are still some noise points that cannot be recognized by the PFD test. The motion consistency test is designed to remove them.

The motion consistency test is another strategy to remove the noises, and it takes advantage of the motion similarity of objects between two frames. This test deals with the remaining optical flow points that have passed both the dominate flow test and the PFD test in the current frame. They are compared with the optical flow points that are in the same position with them and also have passed the two tests in the last frame. The noise points in the current frame cannot hold their motion information in the last frame but the subordinate optical flow points can, because they represent the motion of an object. Based on this rule, the subordinate optical flows in the current frame can be identified. We also set the thresholds for motion information parameters to quantify the similarity

$$\left|\rho_s^t - \rho_r^{t-1}\right| \leq \mu_1 \tag{15}$$

$$\left|d_s^t - d_r^{t-1}\right| \leq \mu_2 \tag{16}$$

where μ_1 and μ_2 are both 1 in the motion consistency test. The subscript s and r indicate any remaining optical flow point that has passed both the dominate flow test and PFD test in the current and last frame, respectively. Besides, they have the same coordinates in the current and last frames

$$\gamma_{x_s}^t = \gamma_{x_r}^{t-1} \tag{17}$$

$$\gamma_{y_s}^t = \gamma_{y_r}^{t-1}. \tag{18}$$

The motion recognition module in our motion detection method includes three tests (i.e., dominant flow test, PFD test, and motion consistency test). If some subordinate optical flow points pass the three tests, it means that we have detected the moving object, and we set the status to motion for the next frame. On the contrary, the status will still be motionlessness.

The scene in which one human is sitting and the other human is pulling a chair in TUM sequences is used to test the motion recognition module. The experimental results of the three tests can be seen in Fig. 5. It can be seen in Fig. 5(b) that the dominate optical flow points that are the majority have been removed by our dominate flow test. In Fig. 5(c), the PFD test is used to keep the optical flow points with similar PFD parameter values. However, some clusters consist of noise points at the left of the chair. It can be seen in Fig. 5(d) that we are able to remove the noise points that have the similar PFD parameter values by the motion consistency test and keep the subordinate optical flow points located at the chair. In the next step, we will segment passive moving objects with the subordinate optical flow points.

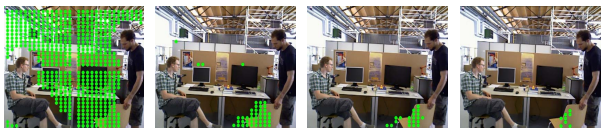


Fig. 5. (a) Global optical flow points. (b) Result of the dominate flow test. (c) Results of the PFD test. (d) Results of the motion consistency test. The experimental scene is from the TUM data set.

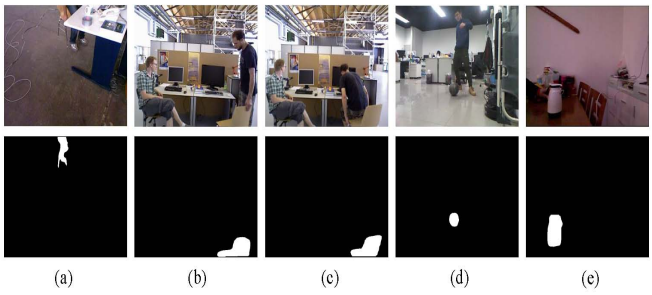


Fig. 6. Passive moving objects segmentation in various challenging scenes.

3) Motionlessness Recognition: We perform the motionlessness recognition module if the status is motion. In the last frame, if the motion information of the subordinate optical flow points is equal to the motion information of the dominant optical flow points, it means that the object has stopped moving, and the status is set to motionlessness for the next frame. On the contrary, it means that the object is still moving, and the status continues to be motion. The subordinate optical flow points in the last frame are used to assist the segmentation of the passive moving objects in the current frame.

4) Getting Mask: The MaskRCNN pretrained by the COCO data set [28] has the capability to detect and segment 80 types of objects. We employ MaskRCNN to segment objects that have the potential to be passive moving ones in an indoor environment, such as chair, book, and suitcase. These 80 different types of objects are the most common ones in our daily routine life. For other objects outside of this scope, they can be detected if relevant training data sets are available. The region growing algorithm is used to segment further the object segmentation mask_r from MaskRCNN.

Fig. 6 demonstrates the segmentation of passive moving objects in various scenarios. The first three scenes (a man pulls a chair) are from the TUM data set, the fourth (a man kicks a ball) was recorded in our laboratory, and the last one (an arm pushes a bottle) is from the Princeton data set [29].

III. Experiments

The sequences in the TUM and ICL-NUIM data sets were used to evaluate the localization accuracy of our RGB-D SLAM scheme. We also tested the time required for the major modules in our scheme. In addition, three dynamic scenes were used to test the performance of our system on dense 3-D mapping.

TABLE II
COMPARISONS OF RMSE OF ATE [m] FOR OUR RGB-D SLAM SYSTEM WITH ORB-SLAM2

Seq.	ORB-SLAM2 [4]			Our system			Imp.
	Mean	Max	Min	Mean	Max	Min	Mean
Offi/room/traj2	0.013	0.017	0.011	0.018	0.023	0.015	-38.462%
Liv/room/traj1	0.163	0.247	0.111	0.200	0.254	0.118	-22.984%
Fr1/room	0.072	0.092	0.048	0.072	0.102	0.059	0.447%
Fr1/floor	0.056	0.063	0.046	0.055	0.060	0.052	1.282%
Fr3/walk/half	0.495	0.739	0.354	0.028	0.032	0.025	94.343%
Fr3/walk/rpy	0.789	0.933	0.483	0.033	0.038	0.030	95.817%
Fr3/walk/static	0.381	0.428	0.320	0.010	0.014	0.007	97.375%
Fr3/walk/xyz	0.678	0.742	0.579	0.014	0.015	0.014	97.935%
Fr3/sit/half	0.033	0.056	0.021	0.019	0.021	0.017	42.424%
Fr3/sit/rpy	0.022	0.031	0.019	0.043	0.102	0.021	-95.455%
Fr3/sit/static	0.008	0.010	0.007	0.007	0.008	0.006	12.500%
Fr3/sit/xyz	0.009	0.010	0.009	0.013	0.014	0.012	-44.444%
Fr2/desk/ps	0.007	0.007	0.006	0.007	0.008	0.006	0.000%

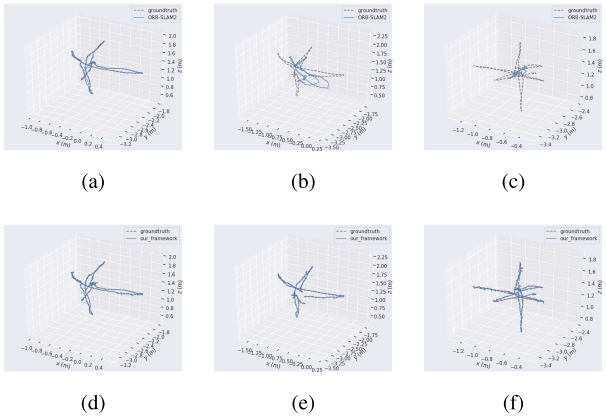


Fig. 7. Camera trajectories for the sequences Fr3/sit/half, Fr3/walk/half, and Fr3/walk/xyz with ORB-SLAM2 (the upper row) and our RGB-D SLAM (the lower row). (a) Fr3/sit/half. (b) Fr3/walk/half. (c) Fr3/walk/xyz. (d) Fr3/sit/half. (e) Fr3/walk/half. (f) Fr3/walk/xyz.

A. Experimental Setup

Different dynamic environments' sequences in the TUM data set and static environments' sequences in the ICL-NUIM data set were used to assess the localization accuracy of the presented RGB-D SLAM scheme. Every sequence contains both RGB images and depth images with a size of 640 × 480. In addition, it offers the ground truth of the camera poses for evaluating the localization accuracy of the RGB-D SLAM system. The words "Offi, Liv, Fr, walk, sit, half, and ps" in Table II mean "Office, Living, Freiburg, walking, sitting, halfsphere, and person," respectively.

An image sequence of our laboratory in dynamic scene was recorded with the Kinect V1. This scene and two dynamic scenes in the TUM data set were used for dense point cloud mapping.

We ran our algorithm on a workstation with the GPU NVIDIA RTX TITAN and Intel Xeon CPU ES-2620 v4. The evo tool [30] was used to assist the measurements.

B. Quantitative Evaluation on Pose Estimation

We evaluate our RGB-D SLAM framework by comparing it with four developed RGB-D SLAM systems. One is ORB-SLAM2 [4], and our system is based on it. The other three are MR-SLAM [10], DS-SLAM [14], and SOF-SLAM [15]. They are designed for dynamic environments particularly.

The absolute trajectory error (ATE) is used to evaluate the localization accuracy of vSLAM widely. The ATE represents the global consistency between the estimated trajectory and the ground truth. It directly indicates the localization accuracy of the system. More details about ATE can be seen in [19]. We present the root-mean-square error (RMSE) of ATE in Table II. The improvements in the tables are calculated by the following formula:

$$P = \left(1 - \frac{F}{C}\right) \times 100\% \tag{19}$$

where F represents the value generated by our RGB-D SLAM framework, C stands for the value of ORB-SLAM2, and P represents the improvement value.

In Table II, we ran ORB-SLAM2 and our system ten times to obtain the mean, maximum, and minimum of RMSE, respectively. In the static environment sequences (the upper four sequences), there is no big difference in the mean RMSE between ORB-SLAM2 and our system. In the highly dynamic sequences (the middle four sequences), the localization errors of ORB-SLAM2 are worse due to its ineffectiveness in dealing with feature points from moving objects. However, its localization accuracy in dynamic environments can be improved significantly using our moving objects removal method. Our motion elimination method is difficult to improve the localization accuracy of ORB-SLAM2 in relatively undynamic sequences (the lower five sequences). The reason is that the people making gestures only are identified as "moving objects" by our segmentation method, and thus, many feature points on the static parts of people are removed.

Besides, in Table III, we compared our system with three state-of-the-art RGB-D SLAM systems that also use the methods of removing moving objects. Note that MR-SLAM depends only on motion cues to segment moving objects, but DS-SLAM and SOF-SLAM use both motion and semantic cues. The data of the three existing systems are from the relevant articles. Our RGB-D SLAM system outperforms MR-SLAM in terms of localization accuracy and is superior to DS-SLAM and SOF-SLAM in some sequences. However, for some sequences, such as the Fr3/walk/static, the accuracy of our system is lower than that of DS-SLAM and SOF-SLAM.

C. Qualitative Evaluation on Pose Estimation

We demonstrated the camera trajectories estimated by both our framework and the ORB-SLAM2 in Fig. 7. The trajectories estimated by our framework are much closer to the ground truth than ORB-SLAM2 in the highly dynamic sequences (Fr3/walk/half and Fr3/walk/xyz). In the relatively undynamic sequences (Fr3/sit/half), the trajectories generated by both frameworks are very close to the ground truth.

TABLE III

COMPARISONS OF RMSE OF ATE [m] FOR OUR SYSTEM WITH THREE MOTION REMOVAL RGB-D SLAM SYSTEMS DESIGNED FOR DYNAMIC ENVIRONMENTS

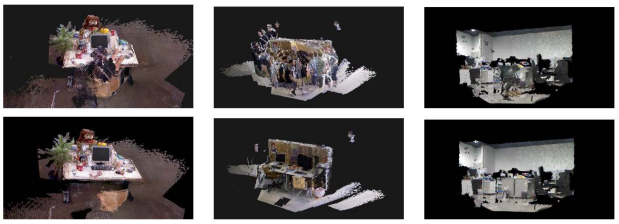
Seq.		MR-SLAM [10]	DS-SLAM* [14]	SOF-SLAM* [15]	Our system*
Highly dynamic sequences	Fr3/walk/half	0.067	0.0303	0.029	**0.028**
	Fr3/walk/rpy	0.073	0.4442	**0.027**	0.033
	Fr3/walk/static	0.033	0.0081	**0.007**	0.010
	Fr3/walk/xyz	0.066	0.0247	0.018	**0.014**
Relatively undynamic sequences	Fr3/sit/half	0.066	-	-	**0.019**
	Fr3/sit/rpy	-	-	-	0.043
	Fr3/sit/static	-	**0.0065**	0.010	0.007
	Fr3/sit/xyz	0.051	-	-	**0.013**
	Fr2/desk/ps	-	-	-	0.007

* means that the system uses both motion and semantic information to segment moving objects. All data about MR-SLAM, DS-SLAM and SOF-SLAM come from [10] [14] [15].

TABLE IV

TIME EVALUATION

Module	Semantic segmentation	Mask inpainting	Motion recognition	Motionlessness recognition
Time(ms)	225.8	634.6	37.6	50.4



(a) Fr2/desk/ps (b) Fr3/walk/xyz (c) Our laboratory

Fig. 8. Dense 3-D mapping. (a) Fr2/desk/ps. (b) Fr3/walk/xyz. (c) Our laboratory.

D. Evaluation on Time

Table IV shows the time consumption of the major modules in our method. The semantic segmentation and mask inpainting are time-consuming compared with other modules. Note that the motion and motionlessness recognition modules also need time for the processing of optical flow information and the region growing.

The average time of processing each frame in our RGB-D SLAM is 0.42 s, and it will be up to 1.10 s when the mask inpainting is needed. The time required for ORB-SLAM2 is about 34 ms for each frame in our testing sequences.

E. Dense 3-D Mapping

The 3-D mapping [31] of the environments is another important task of RGB-D SLAM. In our study, three dynamic scenes are used for reconstruction, the first two are the Fr2/desk/ps and the Fr3/walk/xyz in the TUM data set, and the third one is from our laboratory.

In Fig. 8, the results in the first row are generated by the poses estimated by ORB-SLAM2 and the unprocessed images. The results in the second row are formed by the poses estimated by our system and the images processed by our motion detection and segmentation method. It can be seen from Fig. 8 that our method produces better dense reconstructions in dynamic environments since we use more accurate camera poses to assist the 3-D mapping. In contrast, the poses estimated by ORB-SLAM2 will make the reconstruction confusing in dynamic environments, for example, the ghosting of the table in the scene Fr3/walk/xyz. In the meantime, our dense reconstructions do not contain much information about moving objects.

F. Discussion

The feature points on the moving objects influence pose estimation. The presented scheme improves the localization accuracy in dynamic environments by removing the feature points on moving objects, especially in highly dynamic environments.

However, our segmentation method can be improved in a number of ways. For example, the low-textured background or surface of objects will affect the performance of the proposed moving objects detection method.

There is also space for improvement in terms of computation efficiency. Both the MaskRCNN and the mask inpainting algorithm are computationally costly. Therefore, the presented RGB-D SLAM framework may not be the best choice for applications that prioritize real-time performance. In future work, we will try to replace the MaskRCNN with a time-efficient neural network for object segmentation or use other schemes to segment moving objects.

IV. CONCLUSION

We introduce a moving object segmentation and detection approach based on RGB-D data to improve the localization accuracy of RGB-D SLAM in dynamic environments. In our approach, we use two different ways to segment active and passive moving objects, respectively. To segment active moving objects (humans), a mask inpainting method is proposed to repair the incomplete mask produced by MaskRCNN. With a complete mask, the dynamic feature points on humans can be removed completely. The motion detection based on the LK optical flow is used for passive moving objects. It includes

motion recognition and motionlessness recognition modules, which make the motion detection under moving camera more reliable. Different scenarios from the TUM and ICL-NUIM data set and the image sequence recorded by ourselves were used to test our approach. Experimental results showed that our method can improve the localization accuracy of the ORB-SLAM2 system in dynamic environments, especially in highly dynamic environments.

There are some disadvantages to our method that needs to be further investigated in the future. For example, more detailed segmentation and robust detection for motion are possible directions. The time efficiency is another aspect to be improved.

REFERENCES

[1] S. Shirmohammadi and A. Ferrero, "Camera as the instrument: The rising trend of vision based measurement," *IEEE Instrum. Meas. Mag.*, vol. 17, no. 3, pp. 41–47, Jun. 2014.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[3] J. Al Hage, S. Mafrica, M. El Badaoui El Najjar, and F. Ruffier, "Informational framework for minimalistic visual odometry on outdoor robot," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2988–2995, Aug. 2019.

[4] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[5] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[6] S. Chiodini, R. Giubilato, M. Pertile, and S. Debei, "Retrieving scale on monocular visual odometry using low-resolution range sensors," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 8, pp. 5875–5889, Aug. 2020.

[7] L. Chen, J. Xu, P. X. Liu, and H. Yu, "A RGB-guided low-rank method for compressive hyperspectral image reconstruction," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E101.A, no. 2, pp. 481–487, 2018.

[8] Y. Fan, H. Han, Y. Tang, and T. Zhi, "Dynamic objects elimination in SLAM based on image fusion," *Pattern Recognit. Lett.*, vol. 127, pp. 191–201, Nov. 2019.

[9] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auto. Syst.*, vol. 89, pp. 110–122, Mar. 2017.

[10] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable RGB-D SLAM in dynamic environments," *Robot. Auto. Syst.*, vol. 108, pp. 115–128, Oct. 2018.

[11] Y. Wang and S. Huang, "Towards dense moving object segmentation based robust dense RGB-D SLAM in dynamic scenarios," in *Proc. 13th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Dec. 2014, pp. 1841–1846.

[12] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1187–1200, Jun. 2014.

[13] B. Bescos, J. M. Facil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.

[14] C. Yu *et al.*, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1168–1174.

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

[16] K. Wang *et al.*, "A unified framework for mutual improvement of SLAM and semantic segmentation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5224–5230.

[17] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "MID-fusion: Octree-based object-level multi-instance dynamic SLAM," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 5231–5237.

[18] F. Zhong, S. Wang, Z. Zhang, C. Chen, and Y. Wang, "Detect-SLAM: Making object detection and SLAM mutually beneficial," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1001–1010.

[19] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

[20] J. Sturm, W. Burgard, and D. Cremers, "Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark," in *Proc. IEEE/RJS Int. Conf. Intell. Robot Syst. (IROS)*, 2012, pp. 1–7.

[21] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1524–1531.

[22] B. D. Lucas and T. Kanade, "An iterative image restoration technique with an application in stereo vision," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.

[23] K. He, G. Gkioxari, P. Dollar, and R. B. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988.

[24] J. Hartigan and M. Wong, "A k-means clustering algorithm: Algorithm as 136," *Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.

[25] *Phash Website*. Accessed: Mar. 4, 2020. [Online]. Available: https://github.com/aetilius/pHash

[26] L. Zhu and Y. Zhou, "Background subtraction in mobile cameras by MRF-MAP based optical flows," in *Proc. 7th Int. Conf. Image Graph.*, Jul. 2013, pp. 170–174.

[27] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, Jun. 1994.

[28] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.

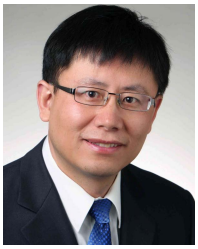
[29] S. Song and J. Xiao, "Tracking revisited using RGBD camera: Unified benchmark and baselines," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 233–240.

[30] *EVO Website*. Accessed: Mar. 4, 2020. [Online]. Available: https://github.com/konanrobot/evo

[31] H. Chen, D. Sun, W. Liu, X. Huang, and P. X. Liu, "An automatic registration approach to laser point sets based on multi-discriminant parameter extraction," *IEEE Trans. Instrum. Meas.*, early access, Jun. 18, 2020, doi: 10.1109/TIM.2020.3003360.

Wanfang Xie received the B.Sc. degree from the Kunming University of Science and Technology, Kunming, China, in 2018. He is currently pursuing the M.Sc. degree with the School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing, China.

His current research interests include visual simultaneous localization and mapping (SLAM) and computer vision.



Peter Xiaoping Liu (Fellow, IEEE) received the B.Sc. and M.Sc. degrees from Northern Jiaotong University, Beijing, China, in 1992 and 1995, respectively, and the Ph.D. degree from the University of Alberta, Edmonton, AB, Canada, in 2002.

He has been with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada, since July 2002, where he is currently a Professor. He is also an Adjunct Professor with the School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing. He has published more than 300 research articles. His research interests include interactive networked systems and teleoperation, haptics, surgical simulation, robotics, and system control.

Dr. Liu is a Licensed Member of the Professional Engineers of Ontario (P.Eng) and a fellow of the Engineering Institute of Canada (FEIC). He has served as an Associate Editor for several journals, including the IEEE/ASME TRANSACTIONS ON MECHATRONICS, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and IEEE ACCESS.

Minhua Zheng (Member, IEEE) received the B.Sc. degree from Beihang University, Beijing, China, in 2010, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2015.

She is currently a Lecturer with the School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing. Her interests include robotics and intelligent systems, human–robot interaction, social robotics, and virtual surgery systems.