IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 29, 2019, accepted June 10, 2019, date of publication June 13, 2019, date of current version June 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2922733

A Compatible Framework for RGB-D SLAM in Dynamic Scenes

LILI ZHAO, (Student Member, IEEE), ZHILI LIU, JIANWEN CHEN, (Senior Member, IEEE), WEITONG CAI, WENYI WANG, (Member, IEEE), AND LIAOYUAN ZENG

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Sichuan 611731, China

Corresponding author: Jianwen Chen (chenjianwen@uestc.edu.cn)

This work was supported by the Sichuan Science and Technology Program under Grant 2018RZ0070.

ABSTRACT Localization and mapping in a dynamic scene is a crucial problem for the indoor visual simultaneous localization and mapping (SLAM) system. Most existed visual odometry (VO) or SLAM systems are based on the assumption that the environment is static. The performance of a SLAM system may degenerate when it is operated in a severely dynamic environment. The assumption limits the applications of RGB-D SLAM in the dynamic environment. In this paper, we propose a workflow to segment the objects accurately, which will be marked as the potentially dynamic-object area based on the semantic information. A novel approach for motion detection and removal from the moving camera is introduced. We integrate the semantics-based motion detection and the segmentation approach with an RGB-D SLAM system. To evaluate the effectiveness of the proposed approach, we conduct the experiments on the challenging dynamic sequences of TUM-RGBD datasets. The experimental results suggest that our approach improves the accuracy of localization and outperforms the state-of-the-art dynamic-removal-based SLAM system in both severely dynamic and slightly dynamic scenes.

INDEX TERMS SLAM, dynamic environment, image segmentation.

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is essential to robotic automatization, which has been developed over thirty decades. SLAM plays an important role in autonomous driving, smart home and intelligent service, etc. When the robot is moving around, its sensor such as the camera could capture the walking people in such usage scenarios. For example, Fig. 1(a) represents the mall scenes [1], and Fig. 1(b) shows the office or expo scenes [2], while Fig. 1(c) represents the smart home scene. Then the localization of the robot would be corrupt using these consecutive images. Although many impressive SLAM systems have been presented and open-sourced [3]. Usually, the basic assumption of these SLAM systems is that the surroundings of the robot is static. However, the accuracy of the location reduces a lot in dynamically changing environment. That is because the dynamic objects could corrupt the mapping of the environment, which results in the wrong estimation of positions.

In recent years, researchers have proposed some approaches to solve this problem. Especially, with the emergence of low-cost depth cameras, RGB-D based visual SLAM extensively becomes popular in the last few years. The purely visual SLAM provides more stable data streams in most circumstances, where the RGB-D sensor like the Kinect is a standard equipment for most robots.

An important issue is how to remove outliers and handle false correspondences if the objects are moving in a common environment. There are several algorithms are proposed, such as the Random Sample Consensus (RANSAC) algorithm, Progressive Sample Consensus (PROSAC) algorithm, and Maximum Likelihood Estimation by Sample and Consensus (MLESAC) algorithm [4]–[6]. However, these algorithms are usually compatible while only small portion of feature points are outliers.

On the other hand, different from the situation in the static environment, the motions of the camera and the moving objects in the scene all need to be considered. So it is hard to determine whether the camera is moving or the objects ahead is moving. Motion detection from a still camera has been studied for several years [7]–[9], where the foreground subtraction (BS) [10] and optical flow [11] are exploited. On the other hand, these approaches are not directly applicable to many applications that involve moving cameras. Some new

The associate editor coordinating the review of this manuscript and approving it for publication was Ran Su.

(a) (b) (c)

FIGURE 1. The application scenes of SLAM. (a) the mall scenes. (b) the office or expo scenes. (c) the smart home scene.

researches on this problem appear in recent years [12]–[15]. These works mainly adopt motion segmentation approaches to handle the motion detection from a moving camera.

In this paper, we choose to use the boundary of the dynamic object to remove the feature points in the dynamic area. We adopt a semantic segmentation algorithm based on Mask-RCNN [16] network and an edge refinement algorithm to find the contour of the potentially dynamic objects. Then a novel approach that inspects the consistency of optical flow between the potentially dynamic area and the background area, is implemented to detect the real state of the potentially dynamic area. The pixels influenced by the dynamic area will be ignored in the feature-based visual SLAM system. And the static scenes enhance the constrains of the whole graph-based optimization framework. The main contributions of this paper can be summarized in three aspects:

- A novel approach to segment potentially dynamic objects precisely is proposed. The algorithm implements the motion removal based on the semantic information.
- A novel motion detection from moving cameras approach is adopted to inspect the consistency between the potentially dynamic-object area and the static-background area.
- We demonstrate that our proposed approach achieves the state-of-the-art localization performance on the TUM [17] dataset.

The remainder of this paper is organized as follows: Section II presents the related work in the past few years about dynamic removal and motion detection from a moving camera. Section III describes our proposed approach. The experimental results tested on the public TUM data sets are given in Section IV. Conclusion and discussion are drawn in Section V.

II. RELATED WORK

This section reviews the previous contributions, which are related to dynamic objects removal in SLAM and visual odometry (VO) system and motion detection from the moving camera. To remove the dynamic objects, the frequently used approaches in existed work can be generally classified into three categories.

The first category is based on inspecting the reprojection error. In [18] and [19], feature points are extracted between two consecutive images, and then matched by calculating the distance of the descriptors. By minimizing the reprojection error of feature points, the motion of camera will be estimated. The feature points whose reprojection errors increase above a predefined threshold will be ignored in the following steps.

The second category is based on the distance-transform error. Reference [20] and [21] used the distance transform in both direct and indirect approaches. The edges are extracted to compute the distance transform of each pixel. The match information can be available without descriptor matching. Then the motion of the camera between the current frame and the reference frame is obtained by reducing the distance transform errors, which saves the computational cost in an elegant way. Moreover, those pixels with a large distance transform error will be discarded.

The third category is based on the motion detection and blob segmentation approach. A dynamic removal approach for a general SLAM system was proposed in [22], which focused on the foreground segmentation of a scene. In [23], a RGB-D foreground segmentation algorithm was proposed, which could be used into the RGB-D slam system for the dynamic removal approach in some circumstances. In [24], Namdev *et al.* proposed an approach, where a dense optical flow was calculated between two consecutive frames. The potential motion information derived from optical flow is taken as the input of a graph-based segmentation algorithm. As the similar portions of potential motion information are clustered, the dynamic parts are obtained. Sun et al. [10] proposed an algorithm where the difference between two continuous images was computed to detect the contours of moving objects. Then the difference is quantized into a vector to segment dynamic objects. In [25], a RDSLAM was presented to compare the appearance and the structure of the image, which was able to detect the difference of GPU-SIFT features.

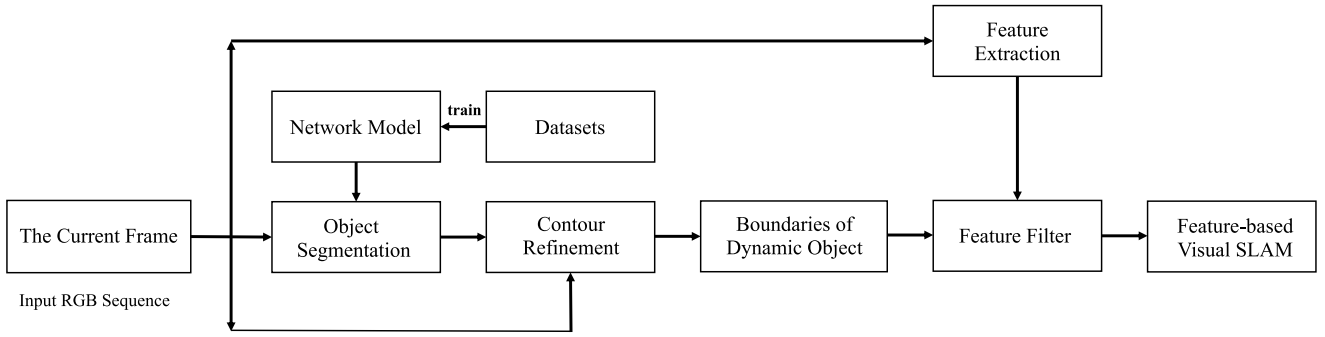


FIGURE 2. Overview of our proposed system.

And the outliers are discarded by an adaptively RANSAC-based approach in view of a prior. However, the accuracy is low and the system is only suitable for a part of scenes. Li and Lee [26] proposed a methodology to detect the edge of foreground in depth images. The weights of the static point at the edge derived from the depth image are calculated based on Student's t-distribution.

And an iterative closest point (ICP) algorithm is implemented by adding the information composed of static-point's weight, intensity weight and geometric weight. Yao *et al.* [27] proposed a real-time visual odometry combining the sparse edge alignment and minimizing reprojection error to obtain robust state estimation. In addition, the edge with the large reprojection error in the dynamic area will be discarded. However, the outliers in the dynamic area are also existed in the estimation phases. Our proposed approach will overcome this problem.

For detecting motion from a dynamic camera based on optical flow, contributions in [11] are made by researchers. The basic principle is to detect the consistency of optical flow between the speed of an object and its background. Then the moving objects is derived due to the object's motion, which deviates from the background radial pattern. A threshold screening strategy is adopted, which is a straightforward and efficient way to compare the motion deviation of pixels for motion determination. However, this approach is generally influenced by occlusion, noise or color changes, where the deep and effective representation of optical flow of the image is ignored.

III. OVERVIEW

The overview of our proposed SLAM system based on the feature points selection using the precise contour detection is illustrated in Fig. 2. Briefly, the system implements the following steps at each frame:

1) Pre-process the input by an instance-aware semantic segmentation of the RGB data.
2) Refine the boundaries of the dynamic objects by integrating the RGB edges with mask boundaries.
3) Take the extracted feature points and boundaries of dynamic objects to determine whether the feature is good or corrupt.
4) Estimate the 3D motion of camera for each input image using the information of good feature point.

The details of each component of the pipeline will be described below.

IV. METHODOLOGY

A. CONTEXT-AWARE PRECISE SEGMENTATION

1) OBJECT SEGMENTATION

For recognizing dynamic and potentially dynamic objects from a single image, an instance-aware semantic segmentation algorithm is implemented. Take the indoor environment as an example, where the main dynamic objects are persons. We use the Mask-RCNN, a state-of-the-art deep convolutional neural network for this task. For each input frame, the Mask-RCNN pre-trained by the COCO dataset [28] is adopted to detect and classify person instances. The person detections are computed independently in each frame. The edge and mask of each person instance will be extracted as Fig. 3.

As Fig. 3 shows, the first column is the original RGB image. The second column shows the drawn edges of the person instances on the gray image. The third column is the binary mask image. The fourth column shows the feature points highlighted with blue, which is detected outside the contour of the mask. Through Mask-RCNN, it can be noted that some useless feature points are still detected around the person instances .

2) CONTOUR REFINEMENT

As Fig. 3 denotes, the contours of the moving person is not precise as expected. To solve the problem, we propose a contours refinement algorithm. First, we implement a canny edge [29] detection algorithm to detect the edge in the original image. Second, the contour of the mask calculated by the semantic segmentation algorithm is repaired by the edge. Researches have shown that the centroid can be used to describe the distribution of pixels in contours. The first-order moments of each contour can be calculated. Assuming that (i, j) is the coordinate of the pixel in the contours and $g(i, j)$ is the value of the pixel (i, j), then the moments of the contour

FIGURE 3. Segmentation results.

can be derived as follows:

$$em_{pq} = \sum_{i=1}^{n}\sum_{j=1}^{n} i^p j^q g(i,j) \tag{1}$$

where pq and n represent the order of the geometric moment and the number of pixels located at the contour respectively. When both p and q are set as 0, the sum of the pixels' value on the contour is calculated as follows:

$$m_{00} = \sum_{i=1}^{n}\sum_{j=1}^{n} g(i,j). \tag{2}$$

Then, when p and q equal 1 and 0 respectively, the first order of the moment along the x-axis is derived.

$$m_{10} = \sum_{i=1}^{n}\sum_{j=1}^{n} ig(i,j). \tag{3}$$

Similarly, the first of the moment along the y-axis is obtained.

$$m_{01} = \sum_{i=1}^{n}\sum_{j=1}^{n} jg(i,j). \tag{4}$$

Finally, the x-coordinate and y-coordinate of the contour's centroid can be described as equation (5) and (6).

$$C_x = m_{10}/m_{00} \tag{5}$$

$$C_y = m_{01}/m_{00} \tag{6}$$

In the same way, the x-coordinate and y-coordinate of each edge's centroid are calculated. We take (e_x^i, e_y^i) as the coordinate of the i-th edge's centroid. Then the distance between the point (C_x, C_y) and (e_x^i, e_y^i) is calculated.

$$d = \sqrt{(C_x - e_x^i)^2 + (C_y - e_y^i)^2} \tag{7}$$

α represents the threshold of the distance, which is set as a matter of experience. When $d > \alpha$, the edge is removed and we reserve the original contour.

$$\begin{aligned} d > \alpha, \quad & e^i \in other\ object \\ d < \alpha, \quad & e^i \in mask_i \end{aligned} \tag{8}$$

After the preliminary selection according to the position of edge, we implement an edge-contour matching operation to further filter out the edge which dose not belong to the contour. Motivated by the observation that the edge and the contour from the same object could have the same trend, such as the radian, the slope and the direction of stretch, we propose an edge-contour matching algorithm to describe the similarity of these properties between an edge and a contour.

At the first step, given that edge is not with the same length as contour, it is necessary to determine which part of contour comes from the same position of one object as the edge. Connecting the centroid of edge and contour, the line intersects with contour at a point as Fig. 4.

To calculate this intersection on the contour, some principles in mathematics are used. Assuming that the centroid of this edge is $P_E = (x_1, y_1)$ and the centroid of this contour is $P_C = (x_2, y_2)$, then the function of the straight line passing through P_C and P_E is computed by,

$$\mathrm{y} = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1 \tag{9}$$

Given that the pixel in the contour is denoted as $\mathrm{P_n} = \left(x_{p_n}, y_{p_n}\right)$, the distance between the point and the line calculated above can be derived as,

$$\mathrm{diff}_n = \frac{y_2 - y_1}{x_2 - x_1}\left(x_{p_n} - x_1\right) + y_1 - y_{p_n} \tag{10}$$

According to the position of the point P_n, we could know the value calculated by the function listed above is positive when the point P_n under the line in the image coordinate system. But it is negative when the point P_n above the line.

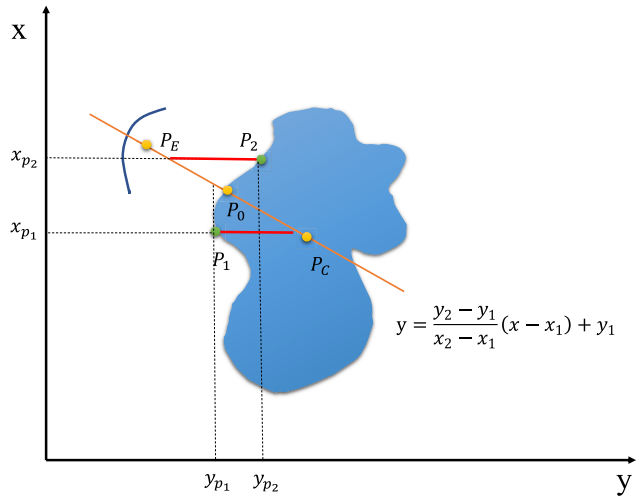


FIGURE 4. The illustration of the intersection calculation.

We calculate the $diff_n$ using all the known position of points on the contour, and then traverse the list of $diff_n$ to multiply two neighboring values. If the result of this product is negative, it suggests the two points are located on either side of the centroid of the edge of the contour respectively. Then we can derive that the intersection P_0 is located between the point $P_1 = (x_{p_1}, y_{p_1})$ and point $P_1 = (x_{p_1}, y_{p_1})$, which cause the value of $diff_1$ and $diff_2$ are negative.

To approximately calculated the coordinate of the intersection P_0, the principle of the similar triangle is applied. The coordinate of the intersection is listed as follows:

$$\begin{aligned} x_{P_0} &= \frac{|diff_1|}{|diff_1| + |diff_2|} |x_{p_1} - x_{p_2}| + x_{p_1} \\ y_{P_0} &= \frac{|diff_1|}{|diff_1| + |diff_2|} |y_{p_1} - y_{p_2}| + y_{p_1} \end{aligned} \tag{11}$$

After the calculation of the intersection P_0, we proposed an approach to construct descriptors for describing the properties stated before. We pick up the same number of points on the contour as the edge. These points are distributed around the intersection uniformly. The longest distance between the intersection and some points in picked point set are smaller than the shortest distance between the intersection and some points in unpicked set.

Two vectors $\vec{v_1}$ and $\vec{v_2}$ are calculated after points picking, which are formed by the points on the edge and the points picked from the contour. The angle of two vectors with respect to the optical center can be calculated by,

$$\begin{aligned} \theta &= \arccos\left(\frac{d_1^T d_2}{\sqrt{d_1^T d_1}\sqrt{d_2^T d_2}}\right) \\ &= \arccos\left(\frac{\left(K^{-1}x_1\right)^T \left(K^{-1}x_2\right)}{\sqrt{\left(K^{-1}x_1\right)^T \left(K^{-1}x_1\right)}\sqrt{\left(K^{-1}x_2\right)^T \left(K^{-1}x_2\right)}}\right) \\ &= \arccos\left(\frac{x_1^T \left(K^{-T}K^{-1}\right) x_2}{\sqrt{x_1^T \left(K^{-T}K^{-1}\right) x_1}\sqrt{x_2^T \left(K^{-T}K^{-1}\right) x_2}}\right) \end{aligned} \tag{12}$$

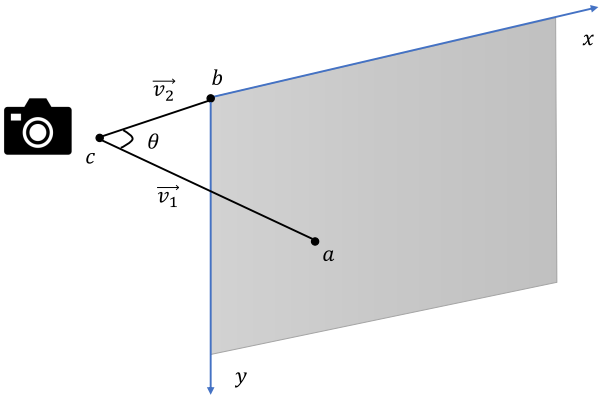


FIGURE 5. The description of $\vec{v_1}$ and $\vec{v_2}$.

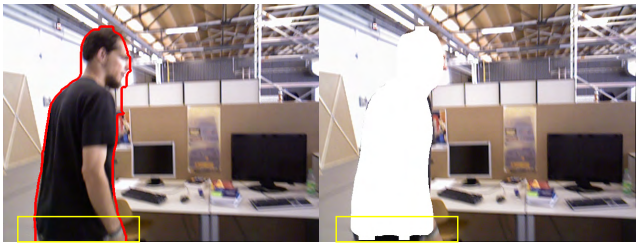


FIGURE 6. The results of the edge refinement. (a) the result of our proposed approach. (b) the result of Mask-RCNN.

which is related with the points in the image, the original point of the image coordinate and the optical center.

As shown in Fig. 5, a is one point in the image and b is the original point of the image coordinate, while c denotes the optical center. Then the Euclidean distance between the two vectors is calculated. Moreover, a threshold ξ is set to filter out the vectors constructed by those edges that do not belong to the contour.

When the edge is remapped to the contour, it can be found that the original contour may not be connected with the edge as expected. To make the contour and edge coincide nearly, a closed operation is employed. Finally, a more precise contour of the dynamic object can be obtained. Then, the contour information will be taken as the input of feature filter.

The segmentation result obtained by Mask-RCNN is shown in Fig. 6(b). As the contour marked with the yellow rectangle denotes, the contour obtained from the segmentation process is not precise as expected. In order to refine the edge of the contour, we propose an edge refinement algorithm as Algorithm 1 shows. Applying the algorithm on the Fig. 6(b), accurate contours are obtained which are highlighted using the red curve as shown in Fig. 6(a).

B. MOTION DETECTION

After potentially dynamic areas have been determined precisely, it is necessary to inspect the real state of the motion in this potentially dynamic areas from the moving camera.

Algorithm 1 ContourRefinement

procedure CONTOURREFINEMENT
contours ← *findcontours*(*mask_image*);
edges ← *Canny*(*RGB_image*);
(C_x, C_y) ← *CalMoments*(*contours*);
for edge in edges **do**
  (e_x^i, e_y^i) ← *CalculateMoments*(*edge*);
  d_1 ← *CalculateEuclideandistance*($(e_x^i, e_y^i), (C_x, C_y)$);
  if $d_1 <$ *threshold*$_1$ **then**
    f ← *CalculateLineFunctoinOfConnectedMoments*($(e_x^i, e_y^i), (C_x, C_y)$);
    $x_{\text{intersection}}, y_{\text{intersection}})$ ← *CalculateIntersection*(f, *contour*);
    $\vec{v_1}$ ← *DiscriptorFormation*(*edge*);
    $\vec{v_2}$ ← *DiscriptorFormation*($(x_{\text{intersection}}, y_{\text{intersection}})$, *contour*);
    d_2 ← *CalculateEuclideandistance*($\vec{v_1}, \vec{v_2}$);
    if $d_2 <$ *threshold*$_2$ **then**
      contours ← *remapping*(*edges*, *contours*);
    end if
  end if
end for
RefinementMask ← *CloseOperation*(*contours*);
RefinementContour ← *findcontours*(*RefinementMask*);
return *RefinementContour*;

Basically, we adopt an optical flow-based approach to check the consistency of potentially dynamic areas and background areas. Optical flow algorithm [30] has been studied generally over the past decades for this field, which performs strongly in motion detection. The general idea of the algorithm is to determine the point correspondences from two consecutive images, which is under the assumption of spatial-temporal consistency of the images. Two kinds of solutions for optical flow problems are stated, which are called sparse and dense solutions. The dense solution calculates optical flow values in the image pixel by pixel. However, the sparse solution just computes flow vectors on those points of interests. And the optical flow value of one pixel can be obtained from formulations as follows:

$$\tau(\mathrm{X}, \mathrm{u}) = \sum_{X_i \in S} \left[I_{l-1}\left(X_i\right) - I_l\left(X_i + u\left(X_i\right)\right) \right]^2 \tag{13}$$

For each 2D coordinate X_i in the set $\mathrm{S} \subset R^2$, the $I_{l-1}\left(X_i\right)$ is the pixel intensity of X_i at frame $l-1$. $I_l\left(X_i + u\left(X_i\right)\right)$ denotes the corresponding pixel intensity value in the frame l. $u\left(X_i\right)$ is the changes of X_i-coordinate between frame $l-1$ and frame l. It is defined as the minimizer of a criterion, which is computed over a local window centered on X_i. To minimize the cost function mentioned above, the flow vector $\left(\frac{u_x(X_i)}{dt}, \frac{u_y(X_i)}{dt}\right)$ of the point X_i can be found. $\frac{u_x(X_i)}{dt}$ is the derivative of coordinate changes with respect to time along the x-axis, and $\frac{u_y(X_i)}{dt}$ is the derivative of coordinate changes with respect to time along the y-axis. Here, the widely used Lucas-Kanade optical flow [30] approach is implemented to track sparse points inside and outside the potentially dynamic objects.

For an optical flow vector $p = (u, v)$, its orientation Φ and magnitude ρ is depicted as follows:

$$\phi = \begin{cases} \operatorname{atan} 2\left(\frac{v}{u}\right) * 180/pi, & \text{if atan2 } \left(\frac{v}{u}\right) > 0 \\ \left(360 + \operatorname{atan} 2\left(\frac{v}{u}\right)\right) * 180/\text{ pi}, & \text{otherwise} \end{cases} \tag{14}$$

$$\rho = \sqrt[2]{\left(u^2 + v^2\right)} \tag{15}$$

Then, similar to [15], a normalized histogram will be constructed for the potentially dynamic area and the background area, in which the range of each bin will be determined by the formulation shown as follows:

$$2\pi * \frac{r-1}{R} < \psi < 2\pi * \frac{r}{R} \tag{16}$$

where R is the number of bins and r is the serial number of bin from left to right. All the flow vectors will be divided into each bins according to their angle from the horizontal axis. In addition, all the flow vectors will be assigned to different clusters. The height of each bin will be computed as below:

$$\mathrm{H} = \frac{\sum_{\xi \in \text{bin}} \rho_\xi}{\sum_{\mu \in \text{area}} \rho_\mu} \tag{17}$$

where ρ_ξ means the magnitude of the flow vector in one bin, and the ρ_μ is the magnitude of flow vector in the potential dynamic area or the background area.

The motion descriptor vector will be constructed as $\mathrm{V} = (H_1, H_2, H_3, \ldots, H_R)$ for both potentially dynamic areas and background areas respectively. Finally, cosine distance will be calculated to determine whether the potentially dynamic

area is moving or not, which are as follows:

$$\cos\Delta = \frac{V_D \cdot V_B}{|V_D| \cdot |V_B|} \tag{18}$$

$$\cos\Delta > \tau, \quad \mathrm{D} \in \text{dynamic area} \tag{19}$$

$$\cos\Delta < \tau, \quad \mathrm{D} \in \text{stationary area} \tag{20}$$

where the D denotes the potentially dynamic area. B represents the background area and τ is a tolerance for measuring the state of motion. The threshold τ is set to avoid the degeneration in low-dynamic scenes. Therefore, we choose the statistical upper bound of the cosine distances of the dense optical flow vectors for the moving object in slightly dynamic scenes as τ, and it can be derived as:

$$\tau = max(\cos\Delta_1, \cos\Delta_2, \cos\Delta_3, \ldots) \tag{21}$$

where $cos\Delta_i$ represents the maximal cosine distance of the dense optical flow vectors in one scene of the TUM-RGBD dataset.

C. FEATURE SELECTION AND FEATURE-BASE VISUAL SLAM

We extract the ORB feature points on the image. It is operated by adding an accurate orientation component into FAST and rotation invariant into BRIEF [31].

Then the ORB feature points will be extracted both in the recognized potentially dynamic area and the background area. Given the results estimated by the approach shown in IV-B, if the area is estimated as the stationary area, the feature points in the area will be kept. Otherwise, the feature points will be ignored in the reference frame and the current frame.

In fact, the feature points inside the contour obtained in the previous section can be easily distinguished and removed. However, the value of the pixel around the dynamic object's contour changes suddenly. Then some unwanted feature points which would influence the accuracy of the location are detected around the contour. Inspired by the concept ''inflation area'', we will remove these feature points. The specific steps are illustrated as follows.

As Fig. 7 denotes, the P is the feature points and the red curve represents a part of the contour of the dynamic object. β represents the distance between the location of the feature points and the contour. We set five pixels along the normal vector of the contour as the threshold. When $\beta > 5$, the feature point will be reserved. But When $\beta > 5$, we will remove it.

In Fig. 8, the red points are the ORB feature points detected in consecutive frames. It can be seen in Fig. 8(a) that both the feature points located inside and around the dynamic object area are detected and highlighted. However, when we apply our proposed feature filter on these feature points, they are successfully removed as shown in Fig. 8(b).

The remaining feature points in previous section will be used to track the egomotion of the camera in ORB SLAM2 [32]–[34]. The estimation of the camera motion is based on the feature points detected in two consecutive RGB images. More specifically, these feature points will be taken into an iterative closest point (ICP) [35] algorithm to get the optimized camera's orientation and position in the front-end by minimizing the error between matched 3D points in world coordinates and key points. Further optimization results will be calculated in the back-end based on more constrains between all frames and loop closure detected.

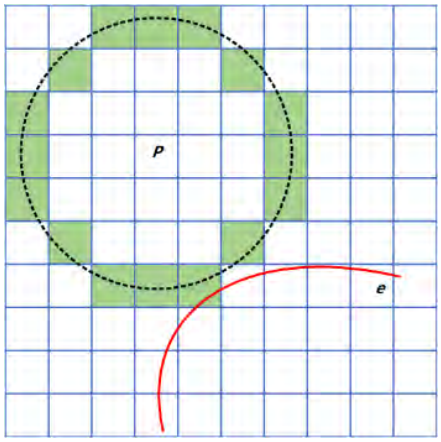
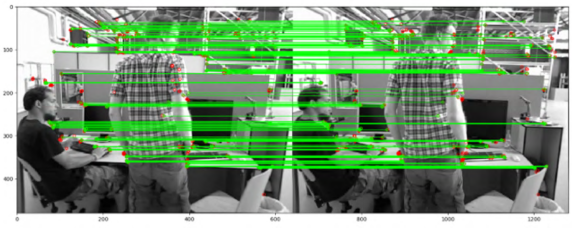
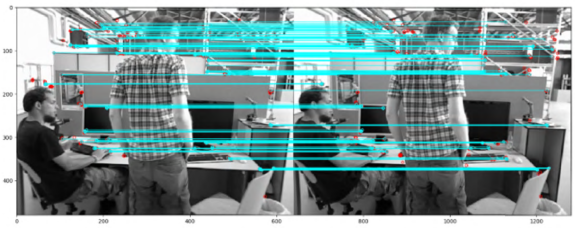


FIGURE 7. Feature points around the contour.



(a)



(b)

FIGURE 8. The experimental results of the feature filter.

V. EXPERIMENTAL RESULTS

The performance of our proposed approach is evaluated on the public RGB-D TUM dataset. The Mask-RCNN segmentation algorithm and edge refine algorithm are implemented on the graphics NVIDIA GeForce GTX 1060. The ORB SLAM2 was run with one CPU Intel Core i7-6700 Processor. We evaluated our proposed SLAM system including the loop closure detection and the map optimization, where Absolute

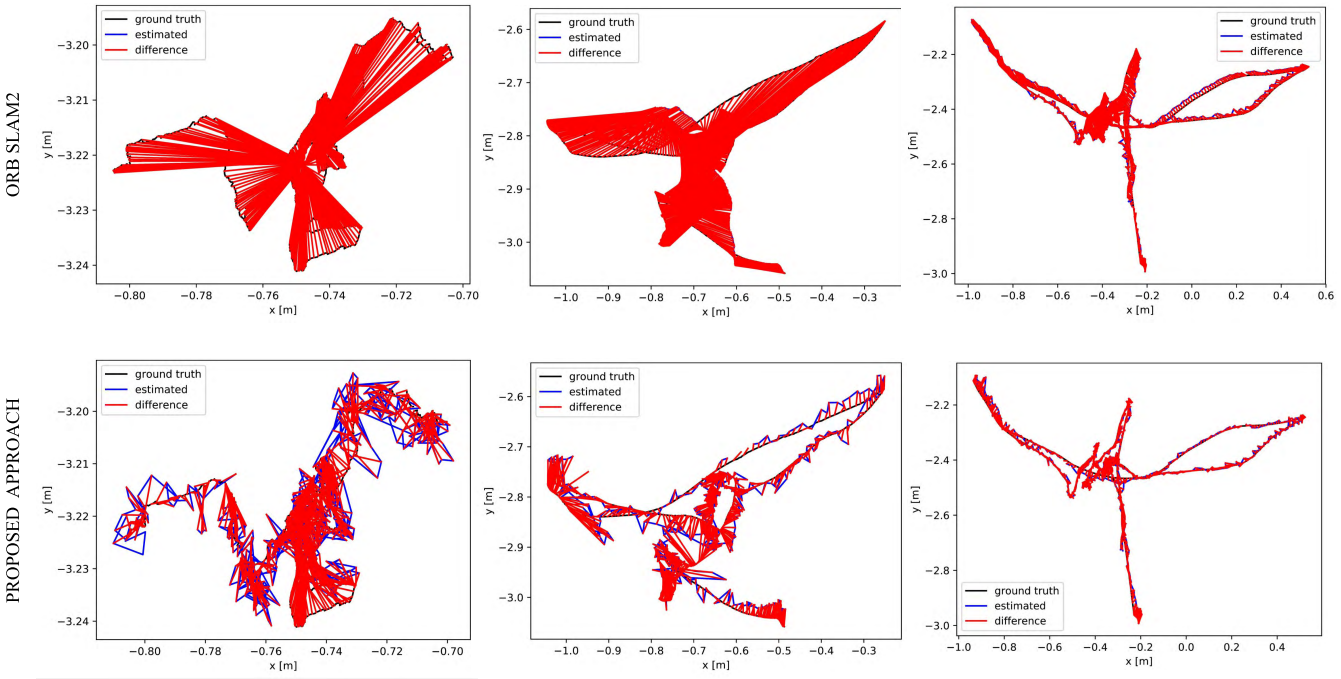


FIGURE 9. Trajectories estimated by ORB SLAM2 and proposed approach.

Trajectory Error (ATE) metric and Relative Pose Error(RPE) metric are used [17]. The estimated trajectories for sequences are compared with their ground truth.

For brevity, we use the words *fr*, *half*, *w*, *s*, *d*, *p* as representatives of sequences *freiburg*, *halfsphere*, *walking*, *sitting*, *desk*, *person*. *halfsphere* and *rpy* stand for the camera motion following a halfsphere-like trajectory and the camera rotating along the roll-pitch-yaw axes respectively. *static* represents the camera roughly kept in place manually. *xyz* depicts the camera moved along the x-y-z axes. The *sitting* sequences describe the scene that changes slowly as low-dynamic scene. Unlike the low-dynamic scene, the scene in the sequence *walking* is defined as the high-dynamic scene.

A. COMPARISON WITH THE ORIGINAL SLAM SYSTEM

Considering that our proposed approach is integrated with ORB SLAM2 system, it is important to compare the estimation results between the original SLAM system and our proposed SLAM system to show the improvements. Fig. 9 is the trajectories estimated by ORB SLAM2 and the proposed approach. The estimated trajectories of original ORB-SLAM2 are shown in the first row. At the same time, the estimated trajectories of our proposed approach are stated in the second row. The shorter the red lines are, the better localization result can be obtained. The Table 1 is the quantitative results by comparing estimated trajectories of each sequences with their ground truth file. Here, we only show the Root Mean Square Error (RMSE) and Standard Deviation Error (STD).

The proposed approach is tested both in low-dynamic scenes and high-dynamic scenes. We did not take the experiments in static scenes. It is noted that the results should keep invariant because there is no potentially dynamic area to be recognized. Moreover, the original ORB SLAMv2 with our approach achieves more improvements in high-dynamic scenes than in low-dynamic scenes. In the low dynamic sequences, there are only small parts that are moving with the low speed. The algorithm of the original ORB SLAMv2 can remove these outliers. However, the original system performs not well in the high dynamic environments. So the improvements of our approach are more notable in the high-dynamic environment relatively. It can be seen in the Table 1, in the high-dynamic scenes, the RMSE average drops up to 85.5% compared with original SLAM system. However, in the low-dynamic scenes, the average decrease is 34.6%. So our approach is proven to enhance the stability and robustness for original feature-based visual SLAM system in severely dynamic environments.

B. COMPARISON WITH STATE-OF-THE-ART VO SYSTEM

To compare our approach with recently state-of-the-art dynamic-removal based VO system, Relative Pose Error (RPE) [17] is adopted to quantify the odometry drift. Intensity-Assisted Iterative Closest Point with Static Point Weighting (SPW-IAICP) is the VO system in the research [26]. The detailed experimental results on the TUM dataset are provided. The work of [27] is used for comparison, which is EP-BASED VO for short, a VO system designed to handle the situations in dynamic environments. Fig. 10 is the

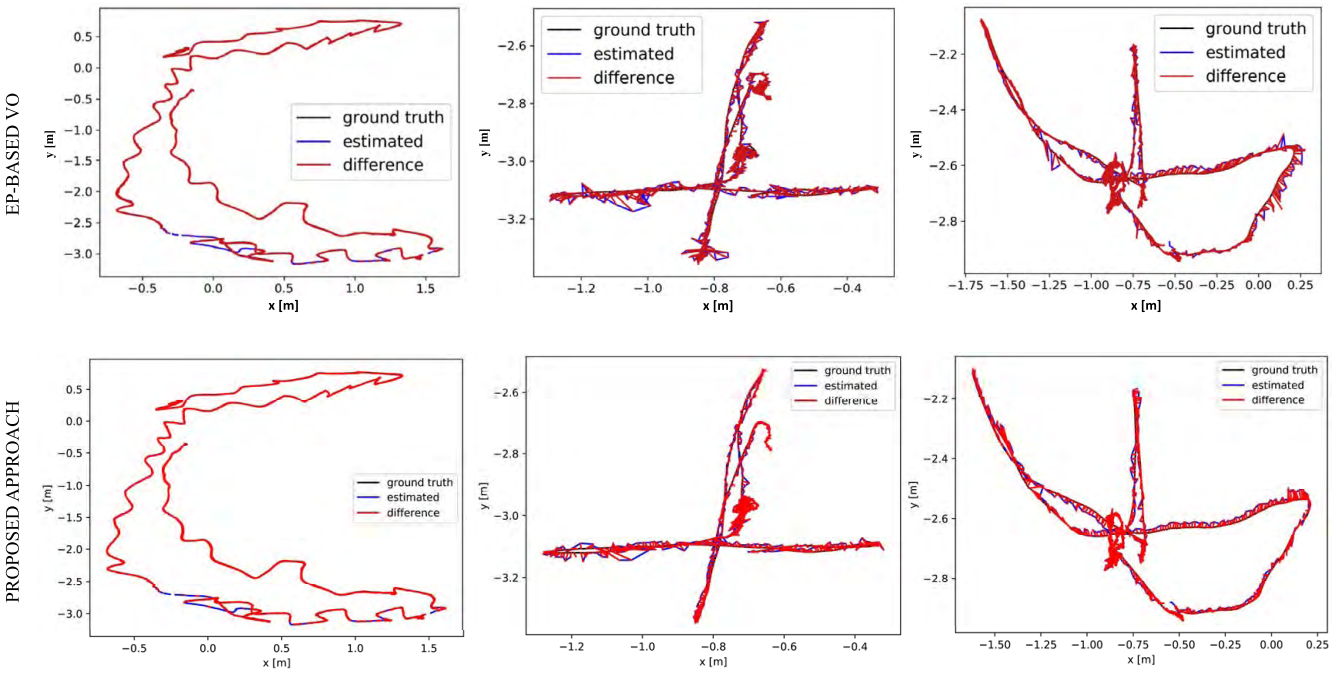


FIGURE 10. Trajectories estimated by EP-BASED VO and proposed approach.

TABLE 1. ATE [m] of ORB-SLAM2 and the proposed approach.

	ORB SLAM2		Our proposed approach		Improvements	
item(M)	RMSE	STD	RMSE	STD	RMSE	STD
fr3/w/xyz	0.2861	0.1317	**0.0139**	**0.0081**	95.10%	93.80%
fr3/w/static	0.025	0.0147	**0.007**	**0.0027**	72.00%	81.60%
fr3/w/rpy	0.1556	0.0765	**0.0299**	**0.0157**	80.80%	79.50%
fr3/w/half	0.3249	0.128	**0.0199**	**0.0097**	93.90%	92.40%
fr3/s/static	0.0068	0.0031	**0.0054**	**0.0028**	19.70%	8.60%
fr3/s/half	0.0321	0.0155	**0.0162**	**0.0073**	49.50%	52.90%

TABLE 2. RPE [m] of the SPW-IAICP, EP-based VO and the proposed approach.

Seq.	RMSE of translational drift [m/s]			RMSE of rotational drift $[deg/s]$		
	SPW-IAICP	EP-based VO	Our proposed approach	SPW-IAICP	EP-based VO	Our proposed approach
fr3/s/xyz	0.0219	**0.0109**	0.0137	0.8466	**0.4717**	0.5091
fr3/s/half	0.0389	0.0168	**0.0157**	1.8836	**0.569**	0.921
fr2/d/p	0.0173	0.0068	**0.0051**	0.8213	0.4359	**0.3561**
fr3/w/static	0.0327	**0.0101**	0.0107	0.8085	0.2571	**0.2433**
fr3/w/xyz	0.0651	0.0292	**0.0202**	1.6442	0.5847	**0.5415**
fr3/w/rpy	0.2252	0.0561	**0.0448**	5.6902	1.021	**0.9543**

trajectories estimated by EP-BASED VO and the proposed approach.

The RMSE values of the translational drift and rotational drift are shown in Table 2. And from the analyses and discussion in [27], it is clear that the EP-BASED VO achieves the state-of-the-art results. However, our proposed approach integrated with ORB SLAM2 outperforms the other two approaches for most dynamic sequences, which can be found in the Table 2.

Finally, a complete comparison between EP-based VO and our proposed approach is listed in TABLE 3. From [27], it can be known that the original ORB SLAM2 performs worse

TABLE 3. ATE [m] of the state-of-the-art EP-based VO and the proposed approach.

Seq.	EP-based VO		Our proposed approach		Improvements	
	RMSE	STD	RMSE	STD	RMSE	STD
fr3/s/xyz	**0.0087**	**0.0041**	0.009	0.0044	-3.40%	-7.30%
fr3/s/half	**0.0148**	0.0074	0.0162	**0.0073**	-9.40%	1.40%
fr2/d/p	0.006	0.0027	**0.0053**	**0.0026**	11.70%	3.70%
fr3/w/static	0.0078	0.004	**0.007**	**0.0027**	10.30%	32.50%
fr3/w/xyz	0.0222	0.0122	**0.0139**	**0.0081**	37.40%	33.60%
fr3/w/rpy	0.0388	0.0241	**0.0299**	**0.0157**	22.90%	34.80%

than the EP-based VO. However, the ORB SLAM2 with our approach is more stable and robust. The improvements indicate our proposed approach have better performance on the most of sequences. In addition, in the slightly dynamic environments, the increase in the RMSE is about 12%. In the highly dynamic environments, an average of 27.5% decrease in RMSE is achieved, where the maximum reaches up to 37.4%. But in very few scenes, the sequences of *xyz* and *half*, the improvement values are negative. Because in these scenes, there are always some objects whose few parts are moving. For our algorithm, once one dynamic object is detected, the whole blob of image will be discarded, even in the low-dynamic scenes. Therefore, some static parts of these objects will be also casted away. However, in EP-BASED VO, only the small parts of the objects are ignored. So in these few scenes, the EP-BASED VO could use all the information of the static parts and then perform better. This limitation in these very few scenes will be a point for our immediate future work. Given that the main scenes are the slightly dynamic and highly dynamic environments, hence the experimental results demonstrate that our approach removes the dynamic features more completely and is more robust in general.

VI. CONCLUSION

In this paper, a novel approach to overcome the degeneration of visual SLAM system in severely dynamic environments is proposed. Our approach takes the advantage of semantic information of the image to recognize potentially dynamic objects. A contour refinement algorithm is adopted to detect the objects precisely. Then motion detection is implemented to verify the consistency of optical flow in both potentially dynamic area and the background area. This approach will be taken as the front end of a feature-based visual SLAM system to track the trajectories and map simultaneously. The experimental results demonstrate that our approach improves the accuracy of localization for generally feature-based SLAM system and outperforms the state-of-the-art dynamic removal SLAM and VO system. For the high-dynamic sequences, the RMSE decreases about 95.1% compared with the original SLAM system and 37.4% compared with the state-of-the-art dynamic removal VO system on average. In the future, our work will focus on the degeneration in low-dynamic scenes because of discarding static-part information. And the implementation and optimization on a high frame rate will also be considered.

REFERENCES

[1] C. C. Loy, S. Gong, and T. Xiang, "From semi-supervised to transfer counting of crowds," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2256–2263.
[2] X. Alameda-Pineda, J. Staiano, R. Subramanian, L. Batrinca, E. Ricci, B. Lepri, O. Lanz, and N. Sebe, "SALSA: A novel dataset for multimodal group behavior analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1707–1720, Aug. 2016.
[3] G. Younes, D. Asmar, and E. Shammas, "A survey on non-filter-based monocular visual slam systems," in *Robotic Automation Systems*, vol. 98. 2016.
[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," in *Readings in Computer Vision*. 1987, pp. 726–740.
[5] O. Chum and J. Matas, "Matching with PROSAC-progressive sample consensus," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 220–226.
[6] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Comput. Vis. Image Understand.*, vol. 78, no. 1, pp. 138–156, 2000.
[7] M. Derome, A. Plyer, M. Sanfourche, and G. Le Besnerais, "Moving object detection in real-time using stereo from a mobile platform," *Unmanned Syst.*, vol. 3, no. 4, pp. 253–266, 2015.
[8] K. Kanatani and C. Matsunaga, "Estimating the number of independent motions for multibody motion segmentation," in *Proc. Asian Conf. Comput. Vis.*, 2002, pp. 7–12.
[9] A. Kundu, K. M. Krishna, and J. Sivaswamy, "Moving object detection by multi-view geometric techniques from a single camera mounted robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 4306–4312.
[10] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auton. Syst.*, vol. 89, pp. 110–122, Mar. 2017.
[11] Y. Sun, M. Liu, and M. Q.-H. Meng, "Invisibility: A moving-object removal approach for dynamic scene modelling using RGB-D camera," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2017, pp. 50–55.
[12] L. A. V. Souto, A. Castro, L. M. G. Gonçalves, and T. P. Nascimento, "Stairs and doors recognition as natural landmarks based on clouds of 3D edge-points from RGB-D sensors for mobile robot localization," *Sensors*, vol. 17, no. 8, p. 1824, 2017.
[13] J. Vertens, A. Valada, and W. Burgard, "SMSnet: Semantic motion segmentation using deep convolutional neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 582–589.
[14] M. Yazdi and T. Bouwmans, "New trends on moving object detection in video images captured by a moving camera: A survey," *Comput. Sci. Rev.*, vol. 28, pp. 157–177, May 2018.
[15] T. Chen and S. Lu, "Object-level motion detection from moving cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2333–2343, Nov. 2017.
[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, " Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[17] 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.

[18] D. Zou and P. Tan, "CoSLAM: Collaborative visual SLAM in dynamic environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 354–366, Feb. 2013.
[19] T. Li, V. Kallem, D. Singaraju, and R. Vidal, "Projective factorization of multiple rigid-body motions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–6.
[20] M. Kuse and S. Shen, "Robust camera motion estimation using direct edge alignment and sub-gradient method," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 573–579.
[21] Y. Ling, M. Kuse, and S. Shen, "Edge alignment-based visual–inertial fusion for tracking of aggressive motions," *Auton. Robots*, vol. 42, no. 3, pp. 513–528, 2018.
[22] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable RGB-D SLAM in dynamic environments," *Robot. Auton. Syst.*, vol. 108, pp. 115–128, Oct. 2018.
[23] Y. Sun, M. Liu, and M. Q.-H. Meng, "Active perception for foreground segmentation: An rgb-d data-based background modeling method," *IEEE Trans. Autom. Sci. Eng.*, to be published.
[24] R. K. Namdev, A. Kundu, K. M. Krishna, and C. Jawahar, "Motion segmentation of multiple objects from a freely moving monocular camera," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2012, pp. 4092–4099.
[25] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular SLAM in dynamic environments," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2013, pp. 209–218.
[26] S. Li and D. Lee, "RGB-D SLAM in dynamic environments using static point weighting," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2263–2270, Oct. 2017.
[27] E. Yao, H. Zhang, H. Xu, H. Song, and G. Zhang, "Robust RGB-D visual odometry based on edges and points," *Robot. Auton. Syst.*, vol. 107, pp. 209–220, Sep. 2018.
[28] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014.
[29] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[30] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas Kanade feature tracker description of the algorithm," *Intel Corp.*, vol. 5, nos. 1–10, p. 4, 2001.
[31] E. Rublee, V. Rabaud, K. Konolige, and R. G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, 2012, pp. 1–8.
[32] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
[33] R. Mur-Artal and J. D. Tardós, "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.
[34] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1027–1037, Oct. 2008.
[35] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vis.*, vol. 13, no. 2, pp. 119–152, 1994.

LILI ZHAO received the B.S. degree in electronic engineering from Zhengzhou University, Zhengzhou, China, in 2016. She is currently pursuing the Ph.D. degree in signal and information processing with the University of Electronic Science and Technology of China, Chengdu, China. From 2018 to 2019, she was with the Department of Electrical and Electronic Engineering, The Hong Kong University, Hong Kong, as a Visiting Ph.D. Student. Her research interests include video coding, image processing and machine learning.

ZHILI LIU was born in Hunan, China, in 1993. He received the B.S. degree in electronic engineering from the University of Electronic Science and Technology of China, in 2016, where he is currently pursuing the M.S. degree. His current research interests include the design of SLAM and path planning module for robot.

JIANWEN CHEN received the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 2007. From 2007 to 2010, he was a Staff Researcher with IBM Research, where he conducted research on wireless communications systems and multi-core video coding architectures. Since, 2010, he has been with the Image Communications Lab, University of California at Los Angeles (UCLA), Los Angeles, where he is currently focusing on video signal processing/enhancement, high efficiency video coding, and high performance computing architecture and application. He is currently a Professor with the University of Electronic Science and Technology of China.

His research interests include signal processing for video and communication systems, and in particular video coding algorithm design, video quality assessment, 3-D video coding, low-complexity video codec optimization, and wireless communication protocols and systems.

WEITONG CAI received the B.S. degree from the School of Electronic Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2017, where he is currently pursuing the M.S. degree with the School of Information and Communication Engineering. In 2015, he was an Exchange Student with National Chiao Tung University, Hsinchu, Taiwan. His current research interest includes video and light field super-resolution.

WENYI WANG received the B.S. degree in electrical engineering from Wuhan University, China, in 2009, the M.A.Sc. degree in electrical and computer engineering from the University of Ottawa, Canada, in 2011, and the Ph.D. degree from the School of Electrical Engineering and Computer Science, University of Ottawa, in 2016. He is currently the Lecturer with the University of Electronic Science and Technology of China. His research interest includes image and video processing.

LIAOYUAN ZENG received the master's and Ph.D. degrees from the ECE Department, University of Limerick, Ireland, in 2006 and 2011, respectively. He is currently a Research Assistant with the University of Electronic Science and Technology of China. He is also an Associate Professor with the University of Electronic Science and Technology of China. His research interests include wireless communication, cognitive radio, and multimedia transmission.