IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 22, 2019, accepted November 4, 2019, date of publication November 7, 2019, date of current version November 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952161

SOF-SLAM: A Semantic Visual SLAM for Dynamic Environments

LINYAN CUI AND CHAOWEI MA
Image Processing Center, School of Astronautics, Beihang University, Beijing 102206, China
Corresponding author: Linyan Cui (cuily@buaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61875003, and in part by the Fundamental Research Fund for the Central Universities of China under Grant YWF-19-BJ-J-125.

ABSTRACT Simultaneous Localization and Mapping (SLAM) plays an important role in the computer vision and robotics field. The traditional SLAM framework adopts a strong static world assumption for analysis convenience. How to cope with dynamic environments is of vital importance and attracts more attentions. Existing SLAM systems toward dynamic scenes either solely utilize semantic information, solely utilize geometry information, or naively combine the results from them in a loosely coupled way. In this paper, we present SOF-SLAM: Semantic Optical Flow SLAM, a visual semantic SLAM system toward dynamic environments, which is built on RGB-D mode of ORB-SLAM2. A new dynamic features detection approach called semantic optical flow is proposed, which is a kind of tightly coupled way and can fully take advantage of feature's dynamic characteristic hidden in semantic and geometry information to remove dynamic features effectively and reasonably. The pixel-wise semantic segmentation results generated by SegNet serve as mask in the proposed semantic optical flow to get a reliable fundamental matrix, which is then used to filter out the truly dynamic features. Only the remaining static features are reserved in the tracking and optimization module to achieve accurate camera pose estimation in dynamic environments. Experiments on public TUM RGB-D dataset and in real-world environment are conducted. Compared with ORB-SLAM2, the proposed SOF-SLAM achieves averagely 96.73% improvements in high-dynamic scenarios. It also outperforms the other four state-of-the-art SLAM systems which cope with the dynamic environments.

INDEX TERMS Computer vision, robotics, semantic, simultaneous localization and mapping.

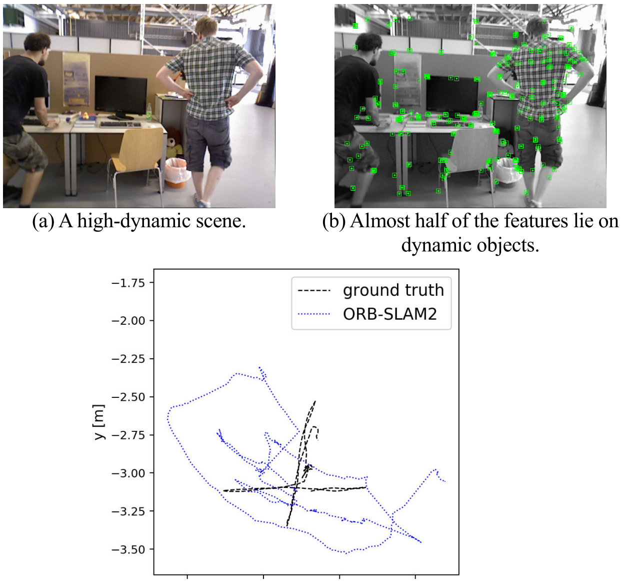
I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) constructs a map of the surrounding world using the data collected by the platform operating SLAM system, and simultaneously locates itself within the map. The sensors carried by the platform to observe the outside world can be various, such as monocular camera, stereo camera, RGB-D camera and lidar. When the sensors are visual sensors, the system is called visual SLAM system. Visual SLAM system is a fundamental and essential module for various kinds of upper applications, such as service robots, augmented reality and autonomous driving cars, where there is a need to estimate camera pose and reconstruct the three-dimensional model of the environment. In the last few decades, the visual SLAM problem has drawn considerable attention from many researchers and there has emerged many excellent visual SLAM systems, such as MonoSLAM [1], PTAM [2], ORB-SLAM [3], ORB-SLAM2 [4], LSD-SLAM [5], SVO [6], DSO [7]. These excellent works can achieve satisfactory performance when the environment is static or the dynamic elements are in minority so that they can be classified as outliers [8], using robust modules like RANSAC and robust cost function. However, the dynamic characteristic of environment is universal in practical applications and sometimes dynamic elements even occupy a large proportion of the scene, such as city streets where there are always moving people or cars. Due to the static world assumption, the accuracy of the standard SLAM systems mentioned above in such high-dynamic environments is reduced so greatly that the results may be totally unreliable.

Fig.1(a) shows a high-dynamic scene, where there are two people walking around. When applying ORB-SLAM2 in this scene, almost half of the extracted ORB features lie on the

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

two moving people, as is shown in Fig.1(b). These dynamic features can't be filtered out as outliers because their number is not in a minority. Due to the effect of these dynamic features, the estimated trajectory is totally unusable. As we can see in Fig.1(c), the blue dotted line is the trajectory estimated by ORB-SLAM2 in the scene shown in Fig.1(a), while the black dashed line is ground truth, their shapes are totally different. This means that the system has failed and can't provide reliable camera position and environment information. The failure is due to the static-world assumption of SLAM system which usually can't be satisfied. Therefore, extension for the standard SLAM system to cope with high-dynamic environment is needed.



(a) A high-dynamic scene.

(b) Almost half of the features lie on dynamic objects.

(c) Comparison of camera trajectory between ORB-SLAM2 estimation result and ground truth.

FIGURE 1. In a high-dynamic scene, the camera trajectory estimated by ORB-SLAM2 is totally unusable.

There has emerged many approaches focusing on enhancing the accuracy of standard SLAM system in dynamic environments. We will review these methods in the following part.

A. RELATED WORK

In visual SLAM systems toward high-dynamic environments, features are usually classified into two groups, static and dynamic features. Only static features are reserved to enhance the accuracy in high-dynamic environments. Various approaches are used to detect dynamic features in the scene and these approaches can be roughly classified into three types: dynamic features detection depending solely on geometry information, dynamic features detection depending solely on semantic information and dynamic features detection through naive combination of the results from geometry calculation and semantic information in a loosely coupled way.

In geometric approaches, the most relevant are as follows. Kundu *et al.* [9] construct the fundamental matrix from robot odometry to define two geometric constraints, one of which is derived from the epipolar geometry. According to the epipolar geometry constraint, a matched feature in the subsequent frame is most likely to be considered as dynamic if it resides too far from the epipolar line. The key in this kind of method is the estimation of the fundamental matrix, if a relatively reliable fundamental matrix can be acquired, then most of the dynamic features can be easily detected. The fundamental matrix can be acquired using purely visual method, such as the 5-point algorithm [10] or 8-point algorithm [11]. If other motion sensors are available, like inertial measurement unit (IMU), the camera motion can also be easily calculated through double integration of IMU data. Lin and Wang [12] detect dynamic features based on the observation that the existence of dynamic features in the calculation of pose estimation will significantly degrade the accuracy of the SLAM system. When a new feature is tracked, two local SLAMs are calculated, one without adding the newly detected feature while the other one with this feature under the assumption that this new feature is stationary. Instead of making hard decision to classify the feature as dynamic or static, the differences of the two results are temporally integrated using binary Bayes filter. After a fixed number of updates, this new feature can be classified as static if the log odds ratio is larger than a predetermined threshold, otherwise the feature is classified as moving. Utilizing the reprojection error to detect dynamic features is another kind of geometric approach. Zou and Tan [13] classify map points as dynamic or static at every frame by analyzing their triangulation consistency. They project features from the previous frame into the current frame and measure the reprojection error of the tracked features. The error should be small if the map point is static, otherwise the map point is classified as dynamic. Wang *et al.* [14] take current RGB image, previous image and current depth image as input, they firstly cluster the depth image into several objects, extract features in current RGB image and count the number and percentage of features on each object. Then features correspondences between current RGB image and previous RGB image are used to calculate fundamental matrix, which is subsequently used to filter out outliers, the number and percentage of remaining inliers on each object are counted again. The remaining inliers are used to calculate fundamental matrix one more time and the following procedure is the same as before. At last a moving objects judgment model is designed based on the statistical characteristics obtained above, and once an object is considered as moving, all features on it are eliminated. Sun *et al.* [15] adapt the codebook learning and inference mechanisms form to deal with the SLAM problem in dynamic environments. Their motion removal approach consists of two online parallel process : the learning process that builds and updates the foreground model; the inference process that pixel-wisely segments the foreground with the built model. Fan *et al.* [16] construct a camera motion model for the moving platform, then decompose the motion model into two parts: translation and rotation. At last, two constraints are proposed to locate the dynamic regions.

Approaches that solely depend on semantic information are straightforward. According to human's common sense and experience, the dynamic objects are usually people, car, etc., which can move by itself. With the quick development of deep learning in recent years, computer vision tasks such as object detection and semantic segmentation can be solved excellently and the accuracy can even outperform human. In SLAM system, when a new frame is coming, by applying advanced CNN architectures like YOLO [17], SSD [18], SegNet [19], Mask-RCNN [20], the semantic label of the extracted features can be acquired. Then features lying on semantically dynamic objects such as people or cars are considered dynamic and removed. Zhong *et al.* [21] use object detection network SSD to detect movable objects, such as people, dog, cat and car. For instance, once a person is detected, it is regarded as a potentially moving object whether it is walking or standing and all features belong to this region are removed. Zhang *et al.* [22] use YOLO to get semantic message, they consider features which are always located on the moving objects as unstable and filter them out. Wang *et al.* [23] propose a step-wise approach that consists of object detection and contour extraction to extract semantic information of dynamic objects in a more computationally efficient way. Xiao *et al.* [24] use SSD object detection network running in a separate thread to get prior knowledge about dynamic objects, and the features on dynamic objects are then processed through a selective tracking algorithm in the tracking thread, to significantly reduce the error of pose estimation.

Some recent works combine the dynamic detection results from geometry calculation and the semantic information. Yu *et al.* [25] use SegNet to get pixel-wise semantic label in a separate thread. If a feature is segmented to be ''person'', further moving consistency check is then conducted using epipolar geometry constraint. If the check result is dynamic, then all features with the semantic label ''person'' will be classified as dynamic and removed. This method actually treats features with label ''person'' as a whole and takes the intersect of two results: only features are both semantically and geometrically dynamic are considered as dynamic. Bescos *et al.* [26] combine the results of semantic segmentation from Mask R-CNN and multi-view geometry. They actually take the union of the two results: features either semantically dynamic or geometrically dynamic are all considered as dynamic.

B. MOTIVATION

The dynamic SLAM systems mentioned above do enhance the accuracy to some extent. However, they remove dynamic features either solely depend on geometry information, solely depend on semantic information, or naively combine the dynamic features removal results of them. Intuitively, geometry information and semantic information in these systems are loosely coupled. If we can find a way to make them tightly coupled, the dynamic features will be removed more effectively, which will lead to further improvement in system accuracy.

C. CONTRIBUTION AND OUTLINE

In this paper we propose a visual semantic SLAM system toward dynamic environment, i.e. Semantic Optical Flow SLAM (SOF-SLAM), which is built on ORB-SLAM2. This framework aims at making the system more accurate in dynamic environments. The proposed SOF-SLAM system can highly reduce the influence of dynamic objects in the environment using our dynamic features detection and removal approach, i.e., semantic optical flow, which detect dynamic features with geometry and semantic information in a tightly coupled way.

Our contribution can be summarized as follows: the proposed SOF-SLAM fully utilizes the complementary characteristic of motion prior information from semantic segmentation and motion detection information from epipolar geometry constraint, while the existing SLAM systems either solely depend on semantic information or geometry information, or naively combine the results of them to remove dynamic features. The dynamic features detection algorithm proposed in our SOF-SLAM, i.e., semantic optical flow, utilizes the semantic segmentation information to aid the calculation of epipolar geometry rather than simply results combination. Therefore our system can remove dynamic features more reasonably and effectively, which lead to more accurate results.

The rest of the paper is structured as follows: the proposed SOF-SLAM is described in Section 2. First, the system overview is presented, Second, a brief introduction to semantic segmentation of potentially dynamic features is given and its limitation is discussed. Third, how to segment real dynamic features with geometry constraint is presented, as well as its limitation. Subsequently we introduce the semantic optical flow algorithm, which detects the true dynamic features effectively. Section 3 evaluates the accuracy of our system on TUM RGB-D dataset and compares our system with the state-of-the-art SLAM systems toward dynamic environments. The qualitative experiments in real world are carried out as well. Finally, a summary is provided in Section 4.

II. SEMANTIC OPTICAL FLOW SLAM

In this section, the proposed SOF-SLAM system will be introduced in detail. The dynamic feature detection and removal method is the main aspect of the illustration.

A. SYSTEM OVERVIEW

The overview of the proposed SOF-SLAM can be seen in Fig.2. First, the procedure of ORB features extraction is conducted just the same as in original ORB-SLAM2, where both static and dynamic features are extracted. Then our proposed dynamic features detection approach, semantic optical flow, can remove dynamic features effectively. The remaining

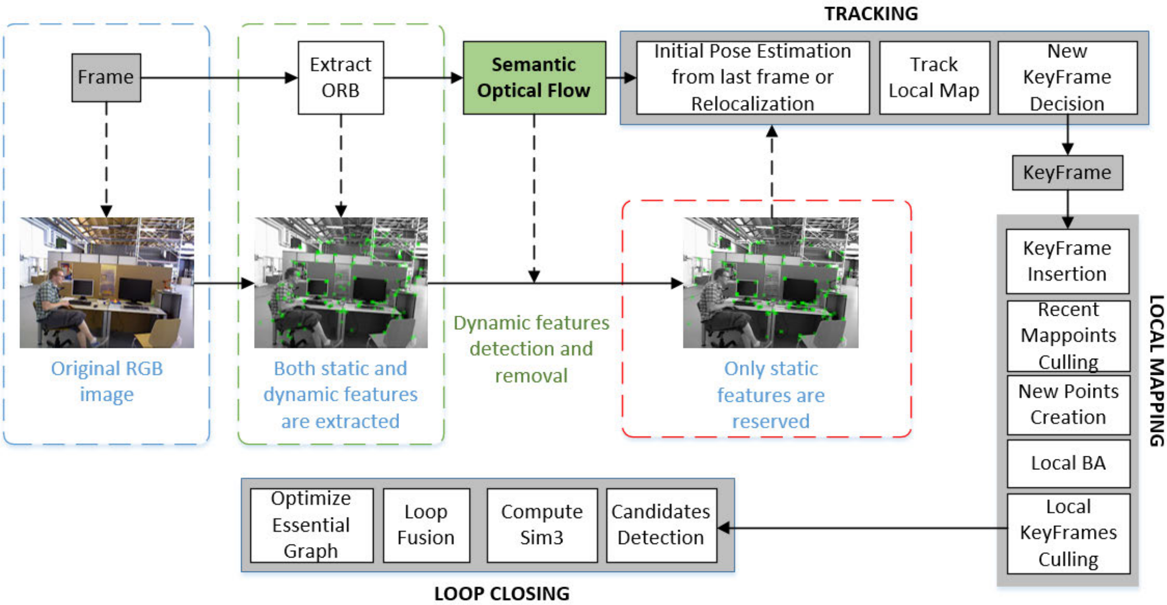


FIGURE 2. Overall-architecture for SOF-SLAM. Local mapping and loop closing threads are the same as ORB-SLAM2. The former one is used to maintain a local map of the surroundings and achieve best camera pose estimation within it, while the latter one is applied to detect loops and conduct loop correction to eliminate accumulated error. We integrate our dynamic features detection approach, semantic optical flow, into the tracking thread, by which dynamic features are effectively removed and only static features are fed to the following tracking procedure.

static features are reserved to engage in the following pose estimation of new frame in the tracking thread. The map point creation and map maintenance of local mapping thread, and the loop detection and loop correction procedure of loop closing thread keep the same as the original ORB-SLAM2. Semantic optical flow is the most important module of our framework, so we will concentrate on the discussion of this module.

B. SEMANTIC OPTICAL FLOW

The flowchart of semantic optical flow for dynamic features detection and removal is shown in Fig.3. On one hand, current RGB image is used to extract ORB features. On the other hand, current RGB image and previous RGB image, with the aid of the semantic segmentation result of current RGB image which is generated by SegNet running in another separate thread, are used to calculate semantic optical flow. The correspondences generated by semantic optical flow are utilized to get a reliable fundamental matrix, which is subsequently used to detect truly dynamic features effectively.

In our dynamic features detection approach, i.e. semantic optical flow, semantic prior and multiple view geometry are unified in a tightly coupled way to achieve effective dynamic features detection. Next we will demonstrate semantic optical flow in details from three aspects: the analysis of motion prior from semantic segmentation, the analysis of multiple view geometry constraint in dynamic features detection, and the way to utilize semantic and geometry information effectively in a tightly coupled form.

1) MOTION PRIOR FROM SEMANTIC SEGMENTATION

In the semantic segmentation thread, we use SegNet encoder decoder network to get pixel-wise semantic segmentation of each input image. The architecture of SegNet consists of two main modules: a encoder network and a decoder network. The encoder network consists of 13 convolution layers. Each encoder layer has a corresponding decoder layer, hence the decoder network has 13 layers as well. The input image is first fed to the encoder network, and each encoder in the encoder network performs convolution with a filter bank to produce a set of feature maps, which then pass through the processes of batch normalization, ReLU (Rectified Linear Unit) activation function and max-pooling. The feature maps produced by the encoder network are then fed to the decoder network. The decoder in the decoder network up-samples the input feature maps using the memorized max-pooling indices from the corresponding encoder feature maps. The up-sampling procedure can produce sparse feature maps. These feature maps are then convolved with a trainable filter bank to produce dense feature maps. A batch normalization step is then applied to each of these maps. The high dimensional feature representation at the output of the decoder network's final

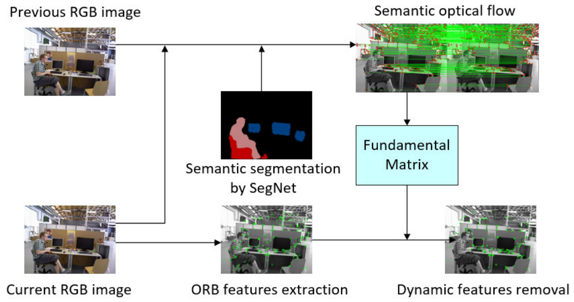


FIGURE 3. Flowchart of semantic optical flow for dynamic features detection and removal.

decoder is fed to a trainable soft-max classifier, which can produce the semantic label of each pixel.

We adopt the caffe implementation of SegNet to produce pixel-wise semantic segmentation. The SegNet model we use is trained on PASCAL VOC dataset, it can segment 20 classes in total (airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motor bike, person, potted plant, sheep, sofa, train, monitor).

According to the semantic label of the pixel, we can get some prior information about its motion characteristics. For example, if the label of the pixel is ''person'', which means the pixel lies on a person, we may assume that this pixel is dynamic with high confidence, as person tends to be moving in our common sense. If the label of the pixel is ''dining table'', we may assume this pixel to be static with high confidence. If the label of the pixel is ''chair'', the situation is different. Chairs can't move by itself, so it should be static under normal circumstance, but it is movable with the influence of other objects, such as the activity of people, so it is not suitable to make the decision whether the pixel on chairs is static or dynamic with high confidence. We consider pixels on this kind of object as ''potentially dynamic''.

Using SegNet, we can get some prior knowledge about the motion characteristic of the pixel, which is useful to boost the accuracy of the SLAM system in dynamic environment. One commonly used method of utilizing the prior knowledge of motion, is using the semantic segmentation results as masks to remove dynamic features. This idea is straightforward and is a simple but useful way to improve the accuracy of localization in dynamic environment. However, there are two limitations. We will then analysis them in detail.

First, as has been mentioned above, the prior knowledge of each pixel can be roughly divided into three categories: static, potentially dynamic and dynamic. Static features are reserved and dynamic features are removed in naive semantic SLAM, but as for potentially dynamic features, there are two ways to deal with them, either treat them as static or dynamic. Both ways can be troublesome.

Fig.4 shows a scene containing potentially dynamic objects. In this scene, there are two monitors and two chairs, whose properties we can get from the segmentation of SegNet should be potentially dynamic, while the actual motion characteristics of them are: the two monitors are static, the chair with wheel on the left is moving due to the man sitting on it, the chair on the right is static. If we treat features lying on all these objects as dynamic and remove them, the accuracy of localization will be worse, this is because there are lots of static features lying on the two monitors and the chair, the features lying on their corners are extremely distinctive, which means that this kind of features can provide accurate and reliable correspondences between consecutive frames. The decrease of the number of static features, especially the removal of distinctive features, will lead to worse accuracy. If we treat features lying on these objects as static and reserve them, the accuracy of localization will also be affected due to the dynamic features on the left chair. In other words, relying solely on semantic segmentation, some dynamic parts of the scene can't be properly handled.

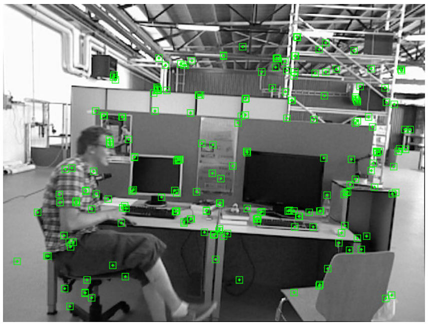
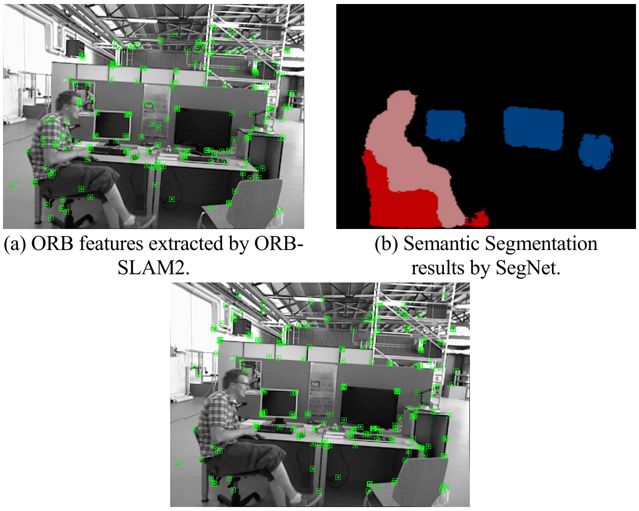


FIGURE 4. A scene containing potentially dynamic objects.



(a) ORB features extracted by ORB-SLAM2.

(b) Semantic Segmentation results by SegNet.

(c) Dynamic features filtering result.

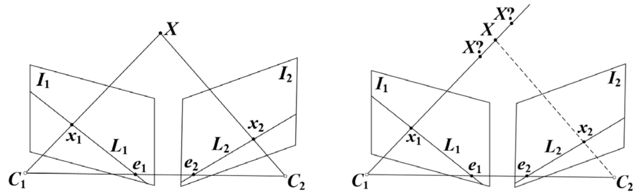
FIGURE 5. An example of filtering dynamic ORB features by using semantic segmentation results naively as masks.

Second, despite the great accuracy improvement of semantic segmentation due to the advanced CNN architectures emerging in recent years, the ambiguity of segmentation result near the boundary of objects is still unavoidable.

Fig.5 shows an example of filtering dynamic ORB features by using semantic segmentation results naively as masks. In Fig.5(a), the ORB features extracted by ORB-SLAM2 are evenly distributed in the image, we can see that there are lots of ORB features lying on the person, which is the main dynamic component in this scene. These dynamic features will decrease the localization accuracy of the SLAM system. Fig.5(b) shows the semantic segmentation results generated by SegNet, in which red, pink, blue and dark represent pixel label of chair, person, monitor and void respectively. In commonly seen semantic SLAM solution, the part of the image whose semantic label is person will be used as a mask to remove dynamic ORB features. The removal effect is shown in Fig.5(c), we can see that most of the dynamic ORB features lying on the person are removed, while there are still some features on the waist, leg and hand of the person reserved.

This is due to the incomplete segmentation of the person, which can be easily observed in Fig.5(b). One ORB feature on the hand of the person is segmented to have a semantic label of "void". Some ORB features on the waist and leg are mistaken to be "chair". Wrong semantic label leads to the reservation of dynamic features, which prevent the system from further localization accuracy improvement.

In SLAM problem toward dynamic environment, the semantic segmentation results from SegNet do help the removal of dynamic features, but the segmentation results are in fact independent from motion situation of the scene. That is, the segmentation result should be same whether an object in the scene is dynamic or not. So another source of information which can reflect the real motion situation of the scene is needed.



(a) The relationship between a static map point and its corresponding image points in two consecutive frames.

(b) The ideal epipolar constraint.

FIGURE 6. **Epipolar constraint of a static feature in multiple view geometry.**

2) MULTIPLE VIEW GEOMETRY CONSTRAINT

Geometric constraints leveraging epipolar geometry properties can be used to check whether a feature is dynamic or static. A static feature should satisfy epipolar constraint in multiple-view geometry, while a dynamic feature will violate the standard epipolar constraint. Fig.6(a) shows the relation between the corresponding image points in two consecutive frames. X is a static map point, which is imaged in two consecutive frames, x_1 at frame I_1 and x_2 at frame I_2. C_1 and C_2 are the optical center of camera for I_1 and I_2 respectively, the line joining C_1 and C_2 is called baseline. The baseline and map point X determine a plane π, which is called the epipolar plane. Plane π intersects with image plane I_1 and I_2 at line L_1 and L_2 respectively. L_1 and L_2 are called epipolar lines. The point of intersection of the baseline with the image plane is called epipole, i.e. e_1 and e_2 in Fig.6.

Suppose now that we only know x_1 in I_1, we want to find its correspondence x_2 in I_2, as is shown in Fig.6(b). Without depth information, we only know that the map point X lies in the ray back-projected from x_1, therefor we only know that x_2 lies in epipolar line L_2. This geometry constraint actually describes the mapping from a point in one image to a corresponding epipolar line in another image, the mapping relationship can be described by fundamental matrix $\boldsymbol{F}$:

$$p_2^T F p_1 = 0 \tag{1}$$

p_1 and p_2 are the homogeneous coordinates of the corresponding image points x_1 and x_2 respectively. Given a point x_1 in I_1 and the fundamental matrix $\boldsymbol{F}$, (1) provides a constraint that x_2 must satisfy if map point X is a static map point. Therefore we can use this constraint to classify whether the map point corresponding to an ORB feature is dynamic or not. However, because of the existence of the unavoidable uncertainty in feature extraction and the estimation of fundamental matrix $\boldsymbol{F}$, the two image points of a static map point may not strictly satisfy (1). Intuitively, image point x_2 doesn't lie exactly on the epipolar line determined by image point x_1 and fundament matrix $\boldsymbol{F}$, but lies very closely to it, just like x_2 in Fig.7. So we can compute the distance D between x_2 and the corresponding epipolar line L_2. If D is smaller than a predefined threshold, then the image point is considered as static, otherwise it is considered as dynamic.

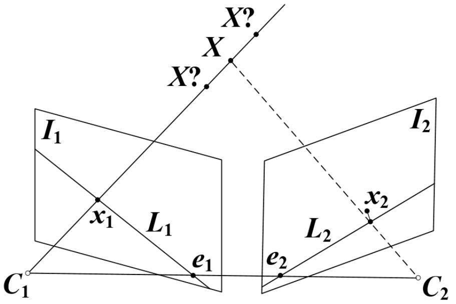


FIGURE 7. **A static feature should resides close to the epipolar line with all kinds of error and uncertainty.**

The key in the epipolar geometry is the estimation of fundamental matrix $\boldsymbol{F}$:

$$F = \begin{pmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{pmatrix} \tag{2}$$

$\boldsymbol{F}$ can be calculated with at least five pairs of feature correspondences, but usually the classic eight-point-algorithm is used. Take the matched image points x_1, x_2 in Fig.6 for example, we can write their homogeneous coordinates:

$$p_1 = (u_1, v_1, 1)\,, \quad p_2 = (u_2, v_2, 1) \tag{3}$$

(u_1, v_1), (u_2, v_2) are the pixel coordinates of x_1 and x_2 respectively. we can get (4) by combining (1)(2)(3):

$$(u_1, v_1, 1) \begin{pmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{pmatrix} \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = 0 \tag{4}$$

Let f denote a vector which contains all elements of Fundamental matrix $\boldsymbol{F}$:

$$f = (f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9)^T \tag{5}$$

By expanding (4), we can get a equation about variable f :

$$(u_1u_2, u_1v_2, u_1, v_1u_2, v_1v_2, v_1, u_2, v_2, 1)f = 0 \tag{6}$$

There are nine unknown elements in f, but due to the scale free characteristic of Fundamental matrix $\boldsymbol{F}$, the degree of freedom of f can be reduced to be 8. Therefore if we have 8 pairs of image points correspondences between two consecutive frames, we can calculate $\boldsymbol{F}$ by solving the equation set consisting 8 equations in the form of (6) .

As for finding corresponding image points between two consecutive frames, optical flow is a convenient and effective way. To reduce the effect of wrong correspondences, RANSAC is adopted. In fact, this method has the chicken-and-egg characteristic. In order to detect dynamic features using epipolar geometry constraint, the fundamental matrix $\boldsymbol{F}$ should be estimated first. On the other hand, we have to use the correspondences of static map points in consecutive frames to estimate the fundamental matrix $\boldsymbol{F}$. Therefore there is a limitation in the procedure of calculating optical flow: most features in the scene have to be static so that RANSAC can reduce the effect of the few remaining dynamic features.

3) DYNAMIC FEATURES DETECTION IN TIGHTLY COUPLED FORM

In order to overcome the above drawbacks of solely using semantic segmentation prior or multiple view geometry constraint to cope with dynamic features, our proposed semantic optical flow utilizes semantic and geometry information in a tightly coupled way to detect dynamic ORB features. Here we use ‘‘tightly coupled way’’ to contrast with the traditional methods which also combine geometry information and semantic information to remove dynamic features. We consider these traditional methods as ‘‘loosely coupled ways’’. This is because these traditional methods firstly utilize geometry or semantic information to detect dynamic features separately, then the two results are combined through a voting module. There are two voting strategies: if two separate results are both dynamic, the final result is dynamic [25], or if either one of the separate results are dynamic, the final result is dynamic [26]. We firstly use semantic information to get a relatively reliable fundamental matrix $\boldsymbol{F}$, then $\boldsymbol{F}$ is used to detect truly dynamic features through geometry constraint. In our approach, fundamental matrix serves as the bridge that links these two sources of information in a unified framework and only one decision is made whether a feature is dynamic or not. The detailed procedure is explained in the following part.

First, we use SegNet to get the motion prior, then when calculating optical flow from current frame to last frame of current frame, the motion prior is used as mask to remove correspondences of features that are dynamic and potentially dynamic. Only reliable correspondences are reserved, as is shown in Fig.8. The correspondences of semantically static features rather than all correspondences are then used to calculate the fundamental matrix $\boldsymbol{F}$.

With the fundamental matrix F calculated above, epipolar line constraint is utilized to find truly dynamic features. In our implementation, we chose 1 pixel as the threshold, the feature in current frame, whose corresponding feature in last frame is more than 1 pixel apart from the epipolar line, is considered as truly dynamic.

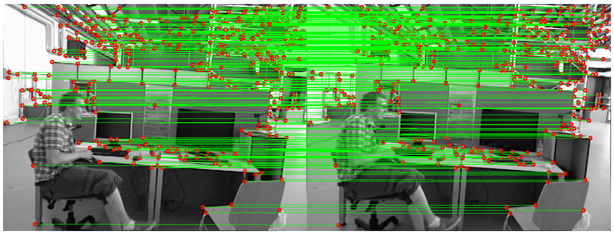
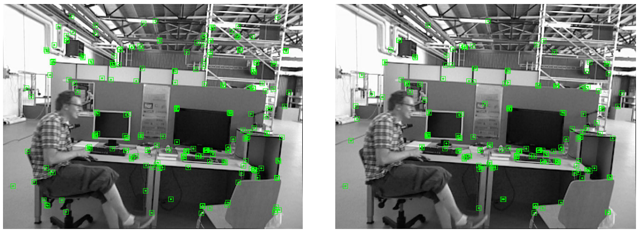


FIGURE 8. An example of semantic optical flow.



(a) Dynamic features removal effect of naive semantic method.

(b) Dynamic features removal effect of our method.

FIGURE 9. Comparison of the dynamic features removal effects between naives semantic method and our method.

Fig.9(a) shows the dynamic features removal result of naive semantic method which solely utilize semantic segmentation result. Fig.9(b) shows the dynamic features removal effect of our semantic optical flow method. First, we can see that almost all features lying on the moving person are removed. Comparing with naive semantic method, our method overcomes the incomplete and inaccurate segmentation characteristic of SegNet. Second, the distinctive features on the two monitors and the chair on the right are confirmed to be static, while the features on the left chair, which has wheels and is moving with the person, are confirmed as dynamic.

III. EVALUATION

We have carried out an experiment of our SOF-SLAM system in public TUM RGB-D dataset to evaluate its performance in dynamic environments. First, we compare our SOF-SLAM framework with the original RGB-D ORB-SLAM2 system and naive semantic ORB-SLAM2 to verify the improvement of our system. Naive semantic ORB-SLAM2 is a system we build on ORB-SLAM2, which solely uses the semantic information generated by SegNet to remove semantically dynamic ORB features, and can be used as reference to clarify that the dynamic features removal approach of our system is more effective. Besides, we compare our approach with the state-of-the-art SLAM systems in dynamic environments using possible results published in the original papers. Further, we demonstrate the performance of our system in a laboratory environment.

A. EVALUATION ON TUM RGB-D DATASET

The TUM RGB-D dataset provides lots of sequences which were captured at 30Hz and a resolution 640×480. The ground truth trajectories are given by a high-accuracy

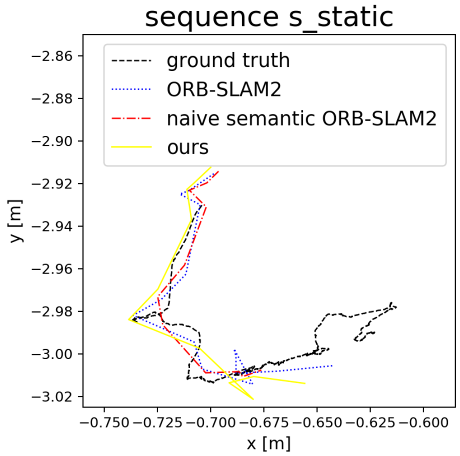
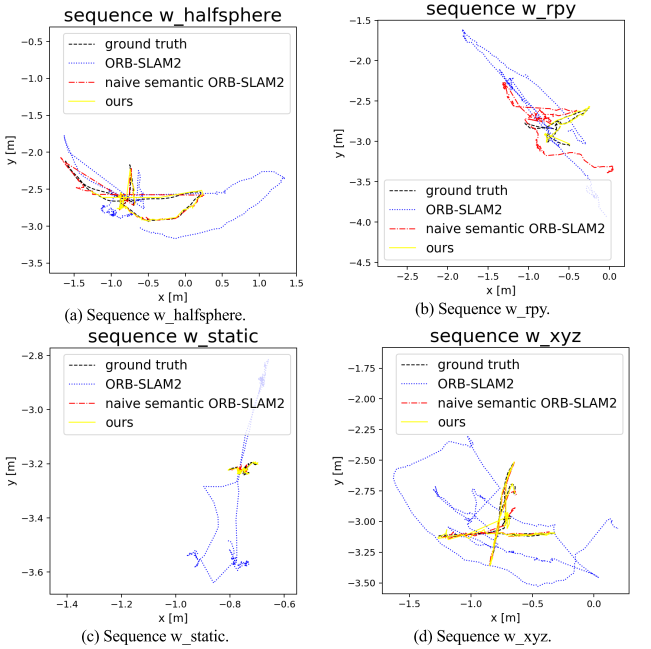


FIGURE 10. Trajectory comparison in low-dynamic sequence.



(a) Sequence w_halfsphere.

(b) Sequence w_rpy.

(c) Sequence w_static.

(d) Sequence w_xyz.

FIGURE 11. Trajectory comparison in high-dynamic sequence.

motion-capture system which is equipped with eight 100Hz cameras. We chose sequences that contain dynamic elements to carry out the experiment, i.e., sequence s_static, w_rpy, w_static, w_xyz. In the five chosen dynamic sequences of TUM RGB-D dataset, people are the main dynamic elements. The word before the underline of the sequence name denotes the state of people in the scene: ''s'' means sitting and ''w'' means walking. The word after the underline of the sequence name denotes the motion of the camera. Sequence s_static is a representation of low-dynamic environment, while the remaining four sequences are representations of high-dynamic environments.

We run ORB-SLAM2, naive semantic ORB-SLAM2 and our system on the five chosen dynamic sequences. The camera trajectories estimated by these three systems are plotted together with ground truth in one figure. We project the 3D trajectories into 2D plane, and utilize the 2D trajectories to exhibit the accuracy of these systems qualitatively and intuitively. If an estimated trajectory coincides with ground truth trajectory more perfectly, the corresponding system is more accurate. The comparison results are shown in Fig.10 and Fig.11. In low-dynamic sequence s_static, the trajectories of three systems are all very close to ground truth. In the high-dynamic sequences, our proposed SOF-SLAM and naive semantic ORB-SLAM2 are close to the ground truth, while the difference between trajectory estimated by ORB-SLAM2 and ground truth is very large. That is because the dynamic elements in low-dynamic environment can be classified as outliers by ORB-SLAM2 and eliminated by the robust modules in SLAM system, such as RANSAC and robust kernels. However, in the high-dynamic scenes, the outliers detection method in ORB-SLAM2 is no longer applicable. In contrast, the proposed semantic optical flow and the semantic information itself are very helpful for dynamic features detection and removal.

Further qualitative comparison of these three systems is carried out to verify the effectiveness of the SOF-SLAM. We calculate the RMSE of ATE(Absolute Trajectory Error) as the qualitative evaluation metric for our experiment. For each approach we run all the sequences five times to get median, mean, minimum and maximum RMSE results, which can reduce the impact of system's non-deterministic nature. The results are shown in Table 1.

According to the results in Table 1, we can see that in low-dynamic sequence s_static, the results of the three approaches are actually very close. In the remaining four sequences representing high-dynamic environment, both our system and naive semantic ORB-SLAM2 makes great accuracy improvement comparing the original ORB-SLAM2, nearly all results, including median, mean, minimum and maximum RMSE, are reduced by an order of magnitude. That is because the semantic information is very helpful to remove dynamic features, and great accuracy improvement is achieved. However, the semantic information itself owns uncertainty just like we have stated in section II. In high-dynamic conditions, still some dynamic features can't be removed effectively. The semantic optical flow proposed in this work fully take advantage of feature's dynamic characteristic hidden in semantic and geometry information, the dynamic features are further removed effectively. Therefore, our system achieves the highest localization accuracy.

In Table 2, we show the improvement in the form of percentage. Comparing against the original ORB-SLAM2, every RMSE statistical result of our system in high-dynamic sequences achieves more than 90% improvement, among which the highest one reaches 98.49%. The improvement of our system against naive semantic ORB-SLAM2 is also shown in Table 2, the average accuracy improvement of median, mean, minimum and maximum RMSE achieves 50.18%, 60.65%, 44.40% and 63.33% respectively.

TABLE 1. Comparisons of RMSE [m] in dynamic sequences of TUM RGB-D dataset for ORB-SLAM2, naive semantic ORB-SLAM2 and our approach.

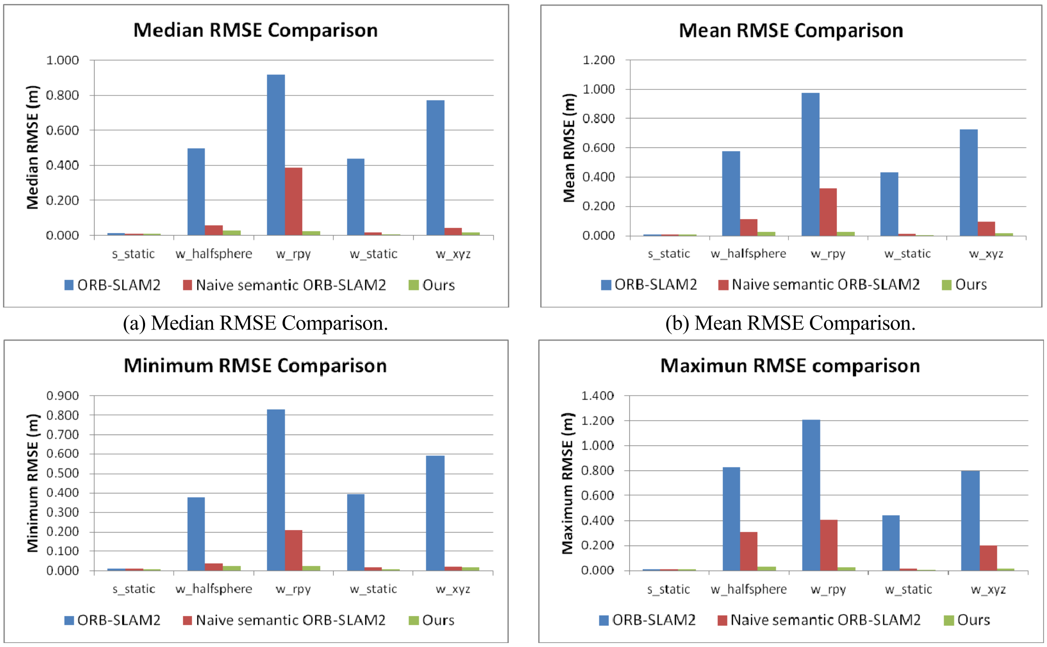
Sequence	ORB-SLAM2				Naive Semantic ORB-SLAM2				Ours			
	Median	Mean	Min	Max	Median	Mean	Min	Max	Median	Mean	Min	Max
s_static	0.012	0.012	0.010	0.012	**0.010**	**0.010**	0.010	**0.011**	**0.010**	**0.010**	**0.007**	0.012
w_halfsphere	0.497	0.576	0.375	0.826	0.056	0.112	0.038	0.310	**0.029**	**0.029**	**0.024**	**0.034**
w_rpy	0.916	0.976	0.828	1.210	0.386	0.323	0.211	0.406	**0.027**	**0.027**	**0.023**	**0.030**
w_static	0.437	0.429	0.394	0.445	0.016	0.016	0.015	0.016	**0.007**	**0.007**	**0.006**	**0.007**
w_xyz	0.771	0.726	0.590	0.800	0.041	0.096	0.019	0.202	**0.018**	**0.018**	**0.017**	**0.020**

TABLE 2. Accuracy improvement of naive semantic ORB-SLAM2 against orb-slam2, our approach against orb-slam2 and our approach against naive semantic ORB-SLAM2.

Sequence	Improvement of naive Semantic Approach against ORB-SLAM2				Improvement of our approach against ORB-SLAM2				Improvement of our approach against naive semantic ORB-SLAM2			
	Median	Mean	Min	Max	Median	Mean	Min	Max	Median	Mean	Min	Max
s_static	**18.98%**	**14.93%**	8.33%	**13.30%**	13.87%	12.98%	**30.00%**	3.40%	-6.31%	-2.30%	23.64%	-11.42%
w_halfsphere	88.68%	80.57%	89.93%	62.43%	**94.25%**	**94.97%**	**93.61%**	**95.91%**	49.17%	74.13%	36.57%	89.10%
w_rpy	57.91%	66.87%	74.57%	66.46%	**97.03%**	**97.20%**	**97.26%**	**97.49%**	92.93%	91.54%	89.24%	92.53%
w_static	96.34%	96.33%	96.12%	96.38%	**98.49%**	**98.48%**	**98.49%**	**98.43%**	58.61%	58.64%	60.93%	56.52%
w_xyz	94.73%	86.82%	96.78%	74.72%	**97.71%**	**97.53%**	**97.16%**	**97.45%**	56.49%	81.23%	11.62%	89.93%

Fig.12 shows the comparison of three system in the form of bar chart intuitively.

Besides, we also compare our approach with the state-of-the-art SLAM systems in dynamic environment, using possible results from the original papers. DS-SLAM [25], DynaSLAM [26], Detect-SLAM [21] and the system proposed by Zhang *et al.* [22] are adopted for comparisons. All the four papers mentioned above are developed upon ORB-SLAM2, and they use RMSE ATE as quantitative metric to compare with original ORB-SLAM2 to show their great accuracy in dynamic environment. However, there is no comparison between them. Our proposed SOF-SLAM is built upon ORB-SLAM2 as well, so we choose median RMSE ATE of our system to carry out accuracy comparison with them. The results are shown in Table 3. As we can observe, the accuracy of our system is far better than DS-SLAM [25], Detect-SLAM [21], the system proposed by Zhang *et al.* [22], and is on par with DynaSLAM [26]. However, we notice that the results of ORB-SLAM2 in same sequence are different between these papers and ours. The results of original ORB-SLAM2 in [21], [25] is very close with what we get when we run ORB-SLAM2 in the same sequence, while the results in [22], [26] are far better than what we get. This may be due to the difference of evaluation details of RMSE or some other difference of experiment condition. Therefore, in order to verify the effectiveness of our system objectively, we choose the relative RMSE reduction (i.e. relative accuracy improvement) of each system with respect to the original ORB-SLAM2 as the evaluation metric. The relative metric is more reasonable as it can eliminate the accuracy difference caused by other factors which are not related to the dynamic features processing algorithm. The new comparison result is shown in Table 4, we can see that the accuracy improvement of our system is only lower than DS-SLAM in low-dynamic sequence s_static, and is better in the remaining four high-dynamic sequences. The reason why DS-SLAM is better than ours in the low-dynamic sequence, is that DS-SLAM adopts the intersect of dynamic features detection results of



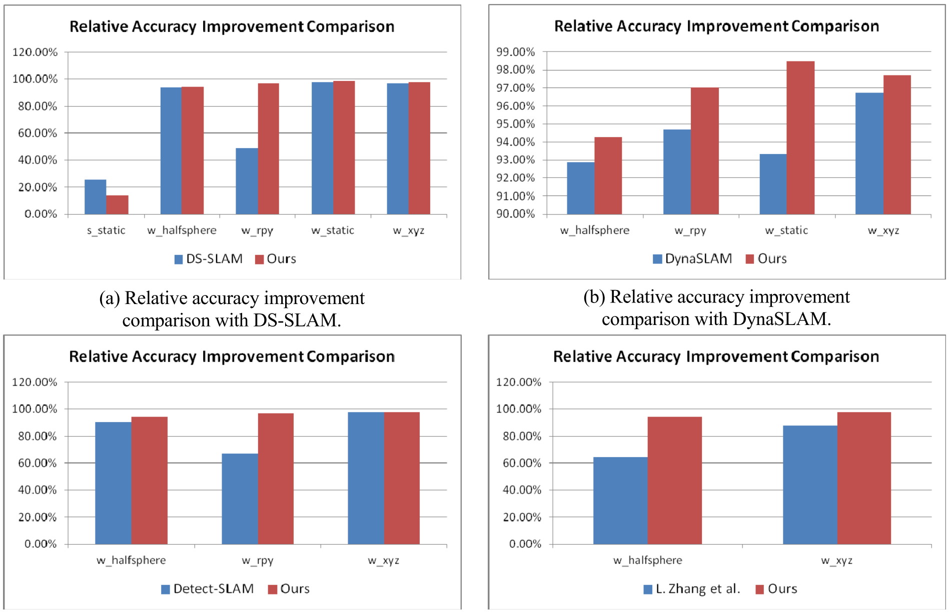
(a) Median RMSE Comparison.

(b) Mean RMSE Comparison.

(c) Minimum RMSE Comparison.

(d) Maximum RMSE Comparison.

FIGURE 12. Comparison of median, mean, minimum and maximum RMSE in the intuitive form of bar chart for ORB-SLAM2, naive semantic ORB-SLAM2 and our system.



(a) Relative accuracy improvement comparison with DS-SLAM.

(b) Relative accuracy improvement comparison with DynaSLAM.

(c) Relative accuracy improvement comparison with Detect-SLAM.

(d) Relative accuracy improvement comparison with L. Zhang et al.

FIGURE 13. Comparison of relative accuracy improvement in dynamic environments between our system and DS-SLAM, DynaSLAM, Detect-SLAM, the system by L. Zhang et al.

geometry method and semantic method which means features are tend to be reserved while our method tends to remove features. In low-dynamic environment, dynamic features are few and their effect are easy to be eliminated, so more features lead to higher accuracy. As for the other three systems, our system is better than them in all five sequences. In general, our system achieves the best results in the four high-dynamic sequences. Fig.13 shows the accuracy superiority of our system against the state-of-the-art SLAM systems in dynamic environment intuitively in the form of bar chart.

B. EVALUATION IN REAL ENVIRONMENT

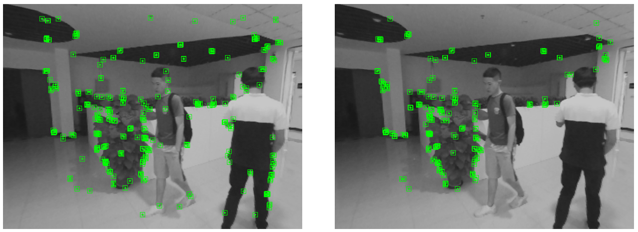
Experiment in real-world environment is also carried out to demonstrate the effectiveness of our system. The RGB images and corresponding depth data are captured by MYNT depth camera. In our real-world experiment scene, there are

TABLE 3. Comparisons of RMSE [m] for our system against the state-of-the-art in dynamic sequences of TUM RGB-D dataset.

Sequence	DS-SLAM	DynaSLAM	Detect-SLAM	L. Zhang et al.	Ours
s_static	**0.0065**	-	-	-	0.010
w_halfsphere	0.0303	**0.025**	0.0514	0.0636	0.029
w_rpy	0.0442	0.035	0.2959	-	**0.027**
w_static	0.0081	**0.006**	-	-	0.007
w_xyz	0.0247	**0.015**	0.0241	0.0336	0.018

TABLE 4. Comparisons of relative RMSE [m] reduction for our system against the state-of-the-art in dynamic sequences of TUM RGB-D dataset.

Sequence	DS-SLAM	DynaSLAM	Detect-SLAM	L. Zhang et al.	Ours
s_static	**25.94%**	-	-	-	13.87%
w_halfsphere	93.76%	92.88%	90.72%	64.31%	**94.25%**
w_rpy	48.97%	94.71%	66.94%	-	**97.03%**
w_static	97.91%	93.33%	-	-	**98.49%**
w_xyz	96.71%	96.73%	97.62%	87.92%	**97.71%**



(a) ORB features extraction with ORB-SLAM2.

(b) ORB features extraction with our system.

FIGURE 14. Comparison of the ORB features extraction situation between ORB-SLAM2 and our system. The dynamic features in walking people are removed with our method.

two people walking around, and the camera moves clockwise into a circle.

Fig.14 shows the comparison of ORB features extraction situation between ORB-SLAM2 and our system in our real-world experiment scene. In Fig.14(a), there are lots features extracted by ORB-SLAM2 lying on the two walking people, while in Fig.14(b), almost all features extracted by our system are on the static background.

Fig.15 shows comparison of estimated camera trajectory between ORB-SLAM2 and our system. The yellow trajectory estimated by our system perfectly forms a closed loop just as how the camera moves, which qualitatively reflects the accuracy of our system, while the blue trajectory estimated by ORB-SLAM2 is unable to return to the origin at end due to the existence of dynamic ORB features.

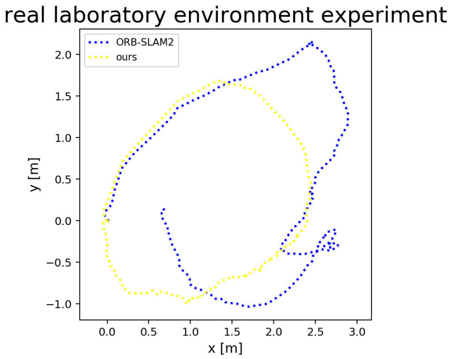


FIGURE 15. Qualitative comparison of estimated camera trajectory between ORB-SLAM2 and our system.

IV. CONCLUSION AND DISCUSSIONS

In this paper, we have presented a semantic visual SLAM system, i.e. SOF-SLAM, building on ORB-SLAM2. We add a separate thread running SegNet to get pixel-wise semantic segmentation and a new approach called semantic optical flow is proposed to detect and remove dynamic features effectively. Our system can overcome the drawback of solely utilizing either semantic or geometry information, and avoid naively combine them. We utilize them in a tightly coupled way, which lead to more reasonable dynamic features removal. To verify the effectiveness of our SOF-SLAM which integrates semantic optical flow, we carry out experiments in public TUM RGB-D dataset and in real laboratory environment. The results show that, our system achieves great improvement on original ORB-SLAM2 in localization accuracy. In high-dynamic sequences, our system shows averagely 96.73% accuracy improvement against the original ORB-SLAM2. In addition, the comparison with the four state-of-the-art SLAM systems in dynamic environments shows that our system achieves the highest relative RMSE reduction with respect to the original ORB-SLAM2.

However, there are still more ongoing works on our system. Our system may be improved in two aspects. First, our system only utilizes the information of two consecutive frames, current frame and last frame, to detect dynamic features in the current frame. We are considering using more image frames, which may provide more abundant temporal information, to determine the motion characteristic of features. Second, our system currently adopts a hard decision way to decide whether a feature is dynamic or not. Further improvement may be achieved by adopting a probabilistic framework to calculate the probability of features being dynamic, which will make our system more robust.

REFERENCES

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[2] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 1–10.

[3] 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.

[4] 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.

[5] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[6] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, May/Jun. 2014, pp. 15–22.

[7] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2016.

[8] R. Castle, G. Klein, and D. W. Murray, "Video-rate localization in multiple maps for wearable augmented reality," in *Proc. Int. Symp. Wearable Comput.*, 2008, pp. 15–22.

[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. Intell. Robots Syst.*, 2009, pp. 4306–4312.

[10] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–777, Jun. 2004.

[11] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, pp. 133–135, 1987.

[12] K.-H. Lin and C.-C. Wang, "Stereo-based simultaneous localization, mapping and moving object tracking," in *Proc. Intell. Robots Syst.*, 2010, pp. 3975–3980.

[13] 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.

[14] R. Wang, W. Wan, Y. Wang, and K. Di, "A new RGB-D SLAM method with moving object detection for dynamic indoor scenes," *Remote Sens.*, vol. 11, no. 10, p. 1143, 2019.

[15] 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.

[16] Y. Fan, H. Han, Y. Tang, and T. Zhi, "Dynamic objects elimination in SLAM based on image fusion," *Pattern Recogn. Lett.*, vol. 127, pp. 191–201, Nov. 2018.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[19] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[20] K. He, G. Gkioxari, P. Dollar, and R. B. Girshick, "Mask R-CNN," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[21] F. Zhong, S. Wang, Z. Zhang, C. Chen, and Y. Wang, "Detect-SLAM: Making object detection and SLAM mutually beneficial," in *Proc. Workshop Appl. Comput. Vis.*, 2018, pp. 1001–1010.

[22] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, "Semantic SLAM based on object detection and improved octomap," *IEEE Access*, vol. 6, pp. 75545–75559, Oct. 2018.

[23] Z. Wang, Q. Zhang, J. Li, S. Zhang, and J. Liu, "A computationally efficient semantic SLAM solution for dynamic scenes," *Remote Sens.*, vol. 11, no. 11, p. 1363, 2019.

[24] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, "Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment," *Robot. Auton. Syst.*, vol. 117, pp. 1–16, Jul. 2019.

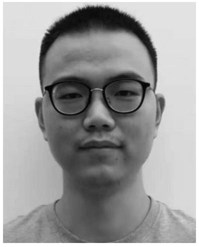
[25] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *Proc. Intell. Robots Syst.*, 2018, pp. 1168–1174.

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

LINYAN CUI was born in Hengshui, Hebei, China. She received the B.S., M.S., and Ph.D. degrees from the School of Astronautics, Beihang University, Beijing, China, in 2006, 2008, and 2013, respectively.

She joined the Image Processing Center, Beihang University, as an Assistant Professor, in 2013, where she is currently an Associate Professor. She has published more than 36 SCI articles in *Optics Express* and other international journals. Her research interests include SLAM, computer vision, turbulence-degraded image restoration, and theoretical modeling of optical waves in atmospheric turbulence.

Dr. Cui was awarded the Excellent National Doctoral Dissertations in China in the field of Aeronautical and Astronautical Science and Technology, in 2016.



CHAOWEI MA received the bachelor's degree from Beihang University, Beijing, China, where he is currently pursuing the master's degree with the School of Astronautics. His research interests include computer vision, 3D reconstruction, and SLAM.