Dynamic Visual SLAM Based on Semantic Information and Multi-View Geometry

1st Junyi Shi
State Key Laboratory of Robotics and System
Harbin Institute of Technology
Harbin, China
syf758521@foxmail.com

2nd Fusheng Zha*
State Key Laboratory of Robotics and System
Harbin Institute of Technology
Harbin, China
zhafusheng@hit.edu.cn

3rd Wei Guo
State Key Laboratory of Robotics and System
Harbin Institute of Technology
Harbin, China
wguo01@hit.edu.cn

4th Pengfei Wang
State Key Laboratory of Robotics and System
Harbin Institute of Technology
Harbin, China
wangpengfei1007@163.com

5th Mantian Li
State Key Laboratory of Robotics and System
Harbin Institute of Technology
Harbin, China
limt@hit.edu.cn

***Abstract*—Visual Simultaneous Localization and Mapping (Visual SLAM) is considered to be one of the important foundations for mobile robots to move toward intelligence, which gives robots the ability to autonomously locate and construct maps in an unknown environment. In the past decades, great progress has been made in the field of visual SLAM, relatively mature algorithm system and program architecture are gradually developed. However, the current researches on visual SLAM mostly assume that the surrounding environment is static, which greatly limits the application of SLAM systems in real world. Aiming at the urgent need of mobile robots for precise localization and map construction in dynamic environment, methods are proposed in this paper. Image semantic segmentation based on deep learning and multi-view geometry methods are combined to recognize and segment moving objects. Only background features are used for camera tracking to avoid the impact of moving objects. Furthermore, an experimental platform is built using RGB-D camera and motion capture device, the effectiveness of the algorithm is verified by public datasets and real scene data.**

***Index Terms*—semantic SLAM, dynamic mapping, multi-view geometry**

I. INTRODUCTION

The application fields of mobile robots continue to expand to high difficulty and intelligence, which requires robots to have higher environmental cognition and adaptability. Among them, the ability to determine its own position in three-dimensional space and be able to remember the surrounding environment is one of the core foundations of robot intelligence. The technology that gives the robot this ability is called Simultaneous Localization and Mapping (SLAM). SLAM technology relies on the onboard sensors of the robot to model the surrounding environment while localization to generate a map. Ideally, the map can be recognized by the robot and used for navigation.

*Corresponding author: Fusheng Zha

Sensors that can be mounted on mobile robots to sense external environment information mainly include visual sensors represented by cameras and ranging sensors represented by lidar. In the trend of increasing requirements for cost control and intelligence, most researchers have turned their attention to low-cost visual sensors. For SLAM applications in indoor environments, RGB-D cameras are favored by researchers for their unique advantages of simultaneously obtaining color and depth information.

Traditional researchs on visual SLAM are under the assumption of static environment, which means during the whole process of program, the objects observed by the camera won't move independently. Taking feature-based visual SLAM as an example, after extracting image features and matching them, the front-end threads of the system often estimate the motion between frames and construct pose maps by using motion only bundle adjustment [1] or PnP algorithm. In dynamic environment, the position of feature points may change, resulting in a huge error that can't be removed by the objective function of pose optimization. The optimization direction of iterative algorithm will be greatly affected by the problem, and a completely wrong interframe motion estimation result will eventually be produced.

In order to solve the above problems, we propose a method of moving pixel detection based on semantic information and multi-view geometry. The contribution of our work can be summarized as follows:

- We propose a method to preliminarily estimate the motion of camera by using the background feature points. The reprojection error and the geometric residual of the pixels in the current frame are calculated, and the judgment criteria are designed to screen out the dynamic pixels.
- We propose an algorithm to determine the moving objects

978-1-7281-9888-0/20/$31.00 ©2020 IEEE

by combining the semantic segmentation result and multi-view geometry method. The camera motion is accurately solved by using the method of optimizing point-line feature fusion and re-projection error.

- We present an extensive evaluation of the framework using public datasets and real scene experiment, the localization result and the construction quality in the dynamic scene of the proposed algorithm are tested.

II. Related work

In most SLAM systems, dynamic objects are considered as abnormal data, which are neither included in maps nor used for pose calculation in camera tracking. RANSAC which used in ORB-SLAM [2] and robot error function that used in PTAM [3] are two typical outlier elimination algorithms.

Due to the appearance of related novel methods, several SLAM systems could deal with dynamic scene more precisely. In feature-based SLAM, Tan et al. [4] detect changes by projecting map features into the current frame for appearance and structure verification. Wangsiripitak and Murray [5] track the known three-dimensional objects in the scene. Likewise, Li and Lee [6] use deep edge points, which have relative weights, indicating the probability that they belong to dynamic objects. Generally, direct methods are more sensitive to dynamic objects in scenarios.

Alcantarilla et al. [7] is one of the most relevant work specially designed for dynamic scenes. Stereo camera is used to detect moving objects through scene stream representation. Wang and Huang [8] use RGB optical flow to segment dynamic objects in the scene. Kim et al. [9] suggest that the static part of the scene can be obtained by calculating the difference between continuous depth images projected on the same plane.

Sun et al. [10] calculated the intensity difference between consecutive RGB images. Pixel classification is accomplished by quantizing the segmentation of depth images. Lifelong models cannot be estimated when a predefined dynamic object remains static (e.g., a parked car or a sitting person). On the other hand, Riazuelo et al. [11] could detect these priori dynamic objects, but cannot detect changes in static objects, such as chairs pushed by someone or balls thrown by someone. That is to say, the former method successfully detects moving objects and the latter method successfully detects multiple moving objects. Ambrus et al. [12] propose a method to segment dynamic objects by combining dynamic classification and multi-view geometry.

III. System Overview

As shown in Figure 1, after the RGBD camera acquires the depth and color images, we use Mask-RCNN [13] to segment the color images in order to obtain the masks which contain various object categories. Then we extract the image features including point and line features based on semantic information.

Since the label of each pixel in the image are known, after obtaining the correct matching results of local features, the inter-frame pose can be quickly calculated by using only ORB feature points of the background, and then the moving objects can be detected and removed using algorithms mentioned below. Futhermore, the result of the preliminary calculation is initialized, and the pose of the removed moving object image are recalculated more accurately based on point and line feature. According to the result of pose calculation, it can be judged whether the frame is a key frame or not. If a key frame is detected, its background pixels will be converted into point clouds, and will be sent to the mapping thread for point cloud processing and fusion. At the same time, the loop detection thread calculates the description vector of the current frame and compares it with the previous key frames. We use DBoW2 [14] as our loop detection algorithm. The most similar key frames will be detected and loop closure of the pose map will be completed.

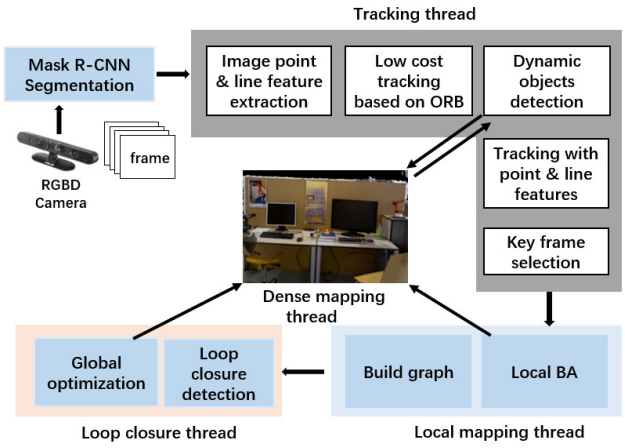


Fig. 1. System Overview

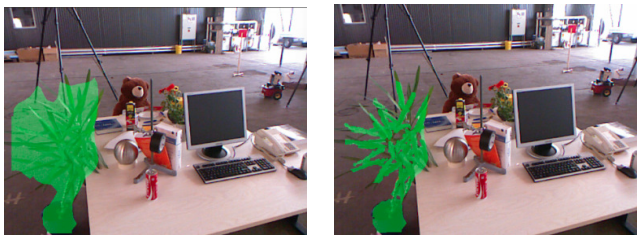
IV. Dynamic SLAM System

A. Semantic Segmentation

Firstly, we use deep network Mask-RCNN [13] for semantic instance segmentation of input images. Aiming at recognizing common objects and scenes in daily life, we use COCO dataset [15] to train the network, and further refine the segmentation result through three-dimensional geometric information. After training, the common objects such as human, chairs and displays can be recognized, but the precision of the target segemntation is still lacking. The mask area of an object often contains a large number of background or pixels of other objects, which may lead to classification errors of feature extraction algorithms based on semantic information. In order to solve this problem, the segmentation result of the neural network need to be further refined.

The main idea of our refine algorithm is to introduce three-dimensional geometric information. Assuming that each object's surface point cloud has good geometric consistency, using KD tree-based filter and clustering algorithm to remove outliers, and finally re-projecting to the original image, the

refined segmentation results can be obtained as shown in Figure 2. It can be seen that the points incorrectly classified as plants are well removed from the mask, and the refinement results are convenient for subsequent processing.

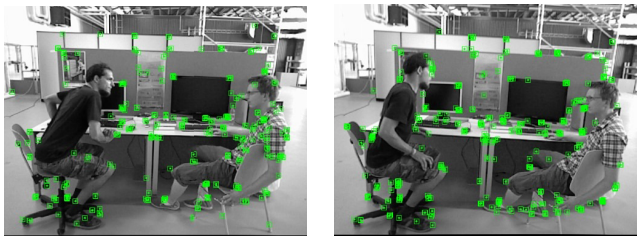


(a) image output from Mask-RCNN (b) image after refined segmentation

Fig. 2. The result of refine segmentation.

B. Point Feature Extraction & Dynamic Feature Points Filtering

Then, image pyramids are built and meshed for each frame of image. On the basis of obtaining semantic information, a certain number of ORB feature points are extracted from each layer of the pyramid grid and their BRIEF descriptors are calculated for matching. When the dynamic object texture is abundant and occupies a large area in the image, the camera pose estimation will be greatly affected by the feature points on the dynamic object. Therefore, it is necessary to eliminate the interference of feature points on dynamic objects when tracking camera in SLAM system. So, we propose a method to filter feature points based on semantic information. Taking the removal of feature points on human body as an example, for all detected Oriented FAST corners, if the classification is human then discard it, as shown in Figure 3.



(a) Feature points before adding semantic information (b) Feature points after removing human body feature points

Fig. 3. The result of filtering feature points.

It can be seen that a large number of feature points in the left picture are concentrated on the man's shirt. After filtering, the feature points on the human body in the right picture are effectively eliminated, which achieves the needs of SLAM system for further matching and calculation. Same as searching the feature points on human body, we also obtain the respective labels for the feature points located on other objects according to their pixel coordinates for subsequent tracking algorithms. However, it can be seen that the feature points in the edge region of the human body are not well removed, which is related to the accuracy of semantic segmentation. This problem can be solved in a certain extent by morphological processing of the mask image obtained by the neural network segmentation, but the segmentation of the neural network may also be incomplete or invalid. In this case, we further use the geometry method which is discussed later in this paper to determine whether the pixel is moving.

C. Line Feature Extraction

If the area of dynamic objects in the image is too large during the localization process, removing it from the image may cause the problem of missing feature points. Considering that there are a large number of line elements in the scene which are not affected by the light, in order to improve the stability and adaptability of the system, the line features are added as landmarks and are fused with the point features for camera tracking and pose optimization. LSD(Line Segment Detector) algorithm [16] is used to extract the line segments in the image according to the slope of the straight line. According to scale information, the extracted lines are merged to reduce computational complexity and improve the robustness of the program. Then the LBD descriptors [17] of the extracted lines are stored in the current data frame together with the BRIEF descriptor.

After using the above methods to extract and merge line features, we need to improve the efficiency of line matching between frames. Similar to the processing of feature points, the system assigns the detected line features to the corresponding labels according to the semantic classification of the region in each frame. As shown in Figure 4, the straight lines on the human body can be accurately removed.

Fig. 4. The result of removing human body line features.

D. Motion Estimation

After obtaining the feature matching between two frames, we transform the pose of the three-dimensional point features and line features, and further project them to the plane of current image. If the pose transformation is accurate and the feature matching result is correct, the projection of the current frame should coincide with the position of the matching points and lines. But in fact, because of the errors of depth measurement, feature extraction and feature matching, there will be error in feature re-projection and feature matching of the current frame. The size of the error provides a benchmark

to measure the accuracy of the pose estimation, optimizing the error to make it as small as possible will lead to a more accurate inter-frame pose estimation result.

Traditional feature point-based SLAM method only optimizes the re-projection error of feature points. In order to improve the robustness of the system in weak feature environment, line features are added. Therefore, the re-projection error of line features should be added to the whole optimization objective function. Because of the requirement of running in dynamic environment, the SLAM system constructed in this paper solve the pose of each frame image twice based on point-line fusion as shown in Figure 1.

The first one is the pose estimation based on feature point re-projection. The first pose estimation result is used for dynamic object detection, and the second point-line fusion pose estimation is used after removing the interference of the moving object. As shown in Figure 5, the point reprojection error is given by:

$$\Delta p(\xi) = p' - p_{project} = (u', v')^T - (u, v)^T \tag{1}$$

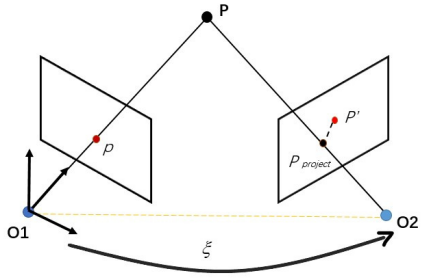


Fig. 5. The point reprojection error.

where $p'$is the i-th detected point in the second frame, and $p_{project} = (u, v)^T$ is the projected point from the first frame to the second one. We define the line equation in the second image as the cross product between the endpoints p_e and p_s of the line in homogeneous coordinates as:

$$l = \frac{p_e^h \times p_s^h}{|p_e^h \times p_s^h|} \tag{2}$$

As shown in Figure 6, the line reprojection error is defined as the sum of the euclidean distances d_e and d_s between the projected line segment endpoints and the detected line in the image plane. That is:

$$\Delta l(\xi) = \left[l^T \cdot \left[{p^s_{project}}^h \quad {p^e_{project}}^h \right] \right]^T \tag{3}$$

where ${p^s_{project}}^h$ and ${p^e_{project}}^h$ are the homogeneous coordinates of the projections of the two endpoints. Assuming that all errors conform to the normal distribution, from the perspective of maximum likelihood estimation, the optimal pose expression obtained by combining formulas (1) and (3) is as followss:

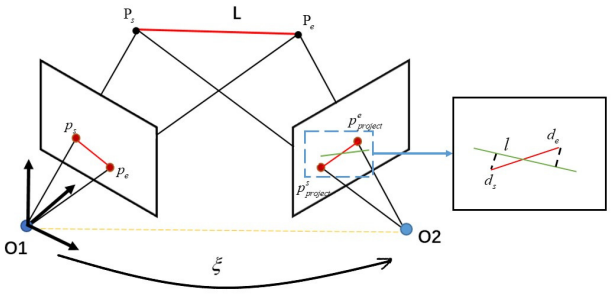


Fig. 6. The line reprojection error.

$$\boldsymbol{\xi}^* = \underset{\xi}{\operatorname{argmin}} \left\{ \sum_i^{N_p} \Delta \mathbf{p}_i(\boldsymbol{\xi})^\top \boldsymbol{\Sigma}^{-1}_{\Delta \mathbf{p}_i} \Delta \mathbf{p}_i(\boldsymbol{\xi}) + \sum_j^{N_l} \Delta \mathbf{l}_j(\boldsymbol{\xi})^\top \boldsymbol{\Sigma}^{-1}_{\Delta \mathbf{l}_j} \Delta \mathbf{l}_j(\boldsymbol{\xi})^\top \right\} \tag{4}$$

where N_p and N_l are the numbers of feature points and feature lines. Solving optimal pose is a typical non-linear least squares problem. Solving optimal pose is a typical non-linear least squares problem, the Gauss Newton method is used to solve the problem on the manifold space iteratively [18].

E. Dynamic Points Detection Based on Background Point Projection

Due to the inaccuracy of the results of semantic segmentation and the possibility of independent motion of other objects, it is difficult to get a good quality point cloud map only by removing dynamic objects based on semantic information. Figure 7 is a point cloud map obtained by splicing a single frame point cloud after directly removing the human pixels. Because the edges of the human body can not be accurately segmented, there are a lot of noise generated by human walking. The map can't distinguish noise from obstacles, so it is useless to robot navigation. In view of the above situation, we propose a geometric method based on background point projection and optical flow tracking to detect moving objects, and a better segmentation result is obtained by combining semantic information.

The basic idea of moving objects detection based on background point projection is shown in Figure 8. For a point P in space, if it is not blocked by other objects from both angels, its three-dimensional coordinates should satisfy the projection equation of two cameras, as shown in Figure 8(a). As shown in Figure 8(b), if point P is blocked by an object that did not exist before, the distance of O2-A can be obtained from the RGB-D camera. At the same time, the distance of O2-P can be obtained from the projection equation, and the two distances are not equal. Therefore, the difference between the two distances can be used as a criterion to measure whether the pixels in the current frame belong to the moving object.

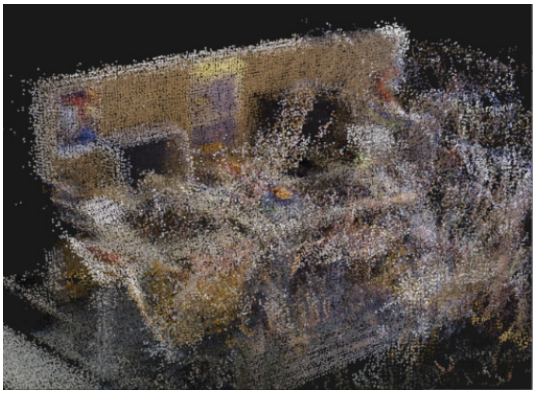
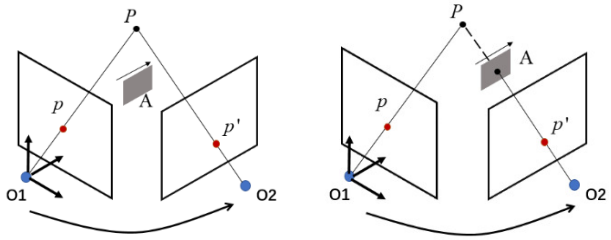


Fig. 7. Point clouds formed by removing pixels belong to human



(a) Unblocked geometric relation (b) Blocked geometric relation

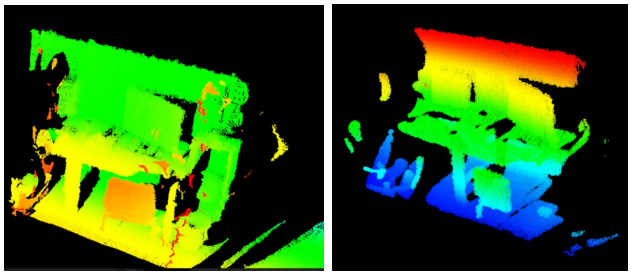
Fig. 8. Dynamic Object Detection Based on Geometric

The key to implement the above method is to acquire the reference point cloud for dynamic object detection in the current frame. The most direct method is to convert all the valid points of the point cloud map into the current camera coordinate system, projecting them to the image plane, and intercepting the standard size of the point cloud and depth map to compare the depth. However, the time complexity of this method will explode rapidly with the increase of map size. By using key frame selection, a fixed number of key frames are used for point cloud splicing as a reference point cloud.

Considering that the key frames are insufficient at the beginning of the system operation, the previous frames of images remove dynamic objects only based on semantic information. At this time, due to the inaccurate segmentation of point clouds, there will still be some dynamic object remains, which will interfere with the detection results. Therefore, it is necessary to remove the point cloud by using the method of point cloud processing. Firstly, the point cloud is downsampled to reduce the computational complexity of the subsequent processing. Then, whether there are enough adjacent points in a certain size range for each point in the point cloud is counted. If not, it is removed from the point cloud, as shown in Figure 9.

F. Dynamic Objects Detection Based on Optical Flow Tracking

A good performance is achieved by using the method of background point projection in the overlap area between the



(a) Original point cloud (b) Point cloud after processing

Fig. 9. Comparison of the point cloud before and after processing

current frame and reference frame, but it can't work in the unique observation area of the current frame. Moreover, if the point cloud generated by the reference frame eliminates the voids generated by dynamic objects, the method can't achieve better results in the void area. Therefore, in addition to using the above methods to detect moving objects, we also use a moving object detection method based on optical flow tracking [19].

The method of moving object pixel detection based on point tracking is shown in Figure 10. Point P_1 is a three-dimensional space point observed in reference frame $O1$. The point moves to point P_2 when the current frame $O2$ start observing. The projection of P_1 on the reference frame image is point P. The projection of P_2 on the current frame image is point P'. The rigid body transformation matrix between the current frame and the reference frame can be obtained by optimizing the reprojection error of feature points, we can easily unify the three-dimensional coordinates of P_1 and P_2 into the same coordinate system. On this basis, whether the feature points belong to the moving object can be determined by calculating the displacement of the tracked point. Taking the moving point as the seed point, more accurate moving object pixel blocks can be acquired by using the region growing algorithm.

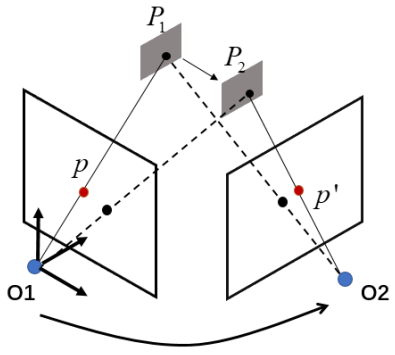


Fig. 10. Dynamic Object Detection Based on Point Tracking

Because of the lack of ORB feature points on weak texture objects, only tracking the extracted ORB feature points will ignore the moving objects with relatively simple texture. Therefore, we use the strategy of extracting the target points evenly from the pixels in the object recognition frame and

tracking them using the median optical flow algorithm. The time range of this algorithm is limited between the two near key frames, and the non-object points in the recognition frame will be gradually removed during the tracking process.

The characteristic of the median optical flow method is that each tracking point is tracked back and forth twice. The bias between the original point and the back tracking point is taken as the criterion to evaluate the tracking quality of the point, and half of the bias is extracted in each inter-frame tracking. Because the system needs to track objects between key frames for a long time, if we remove half of the points in each tracking, it will cause the problem of insufficient follow-up tracking points.

Therefore, some improvements has been made to the median optical flow algorithm. Firstly, when selecting target points, the object recognition frame is expanded to ensure that the selected target points can cover the whole object's pixel area. In addition, during the first frame tracking, the minimum bias of the half points with poor tracking quality is used as the criterion to evaluate the tracking quality of the follow-up target points. At the same time, NCC (Normalized Cross Correlation) verification [20] is used on the tracking points, which is more robust to illumination change.

Relatively stable tracking effect without too much computation can be maintained by using the above method. When inserting the next key frame, the pixel coordinates of the target point in the previous frames can be obtained and the three-dimensional coordinates of the target point at each time can be calculated by combining the pose of each frame. The total displacement D of the target point can be calculated according to these coordinates. If D is greater than the set threshold, the point is considered to be a dynamic point. Then the local growing algorithm of multiple sub-points is used to get the moving object's pixel blocks and remove them in the process of mapping, which means a tracking thread is completed. For the same object, when inserting the key frame, the tracking point is re-initialized in the recognition box to enter the next tracking.

Because of the working principle of RGB-D camera, the measurement value of depth information on the edge of an object often appears a large change, which may cause the edge point to be misdetected as a dynamic point. In order to solve the problem, for a pixel point detected as a dynamic object, $4 * 4$ pixels around it are collected and its standard deviation is calculated. If the standard deviation is bigger than a certain threshold, which means there is a sharp depth change near the point, the point will be regarded as the edge point of the object, and it will be reset as a static point. As shown in Figure 11, the marked points in the graph are all the target points in the tracking process. The green points are the static points under this threshold, and the red points are the dynamic target points.

The above improved optical flow tracking algorithm only obtains some discrete target points. Considering the need to completely remove dynamic objects during the construction process, it is necessary to find the edge of the object better

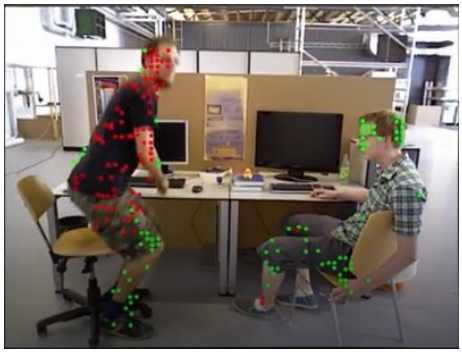
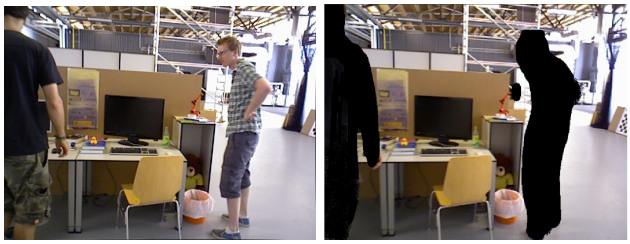


Fig. 11. Result Of Dynamic Object Detection

when the object type is identified. In this paper, based on the mask obtained by deep learning, the image is segmented using region growing algorithm. The seed point used is the point where the median optical flow is used for tracking and judged as motion. Starting from the set seed point, it gradually traverses the pixels around the point and divides the pixels with similar properties into a class until all the pixels that meet the conditions are successfully included [21].

As shown in Figure 12, the moving human body is completely cut out from the image. This method makes up for the shortcomings of the point cloud reprojection method in the unrepeated observation area. After combining these two moving object detection methods and semantic information, moving objects in the image can be segmented more accurately. The results of the comprehensive experiment can be seen in the following chapter.



(a) Original image (b) Region growing result

Fig. 12. Segmentation result of region growing

V. Experimental results

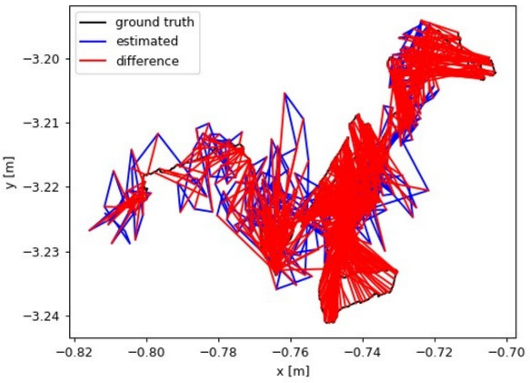
We perform a public dataset and real-world experiments to evaluate the proposed system. In the first experiment, we compare the proposed method with another state-of-art algorithm on public datasets. We perform a numerical analysis to show the accuracy of our system in details. We further test our system in real-world dynamic environment.

A. Dataset Comparison

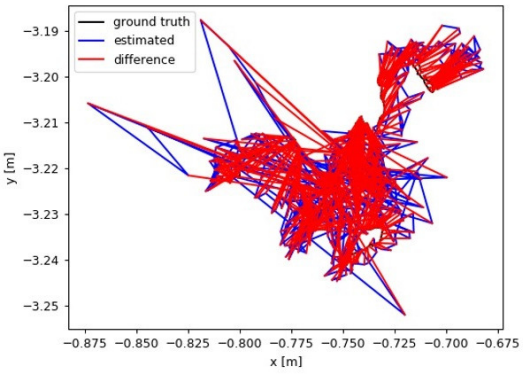
We evaluate our proposed system using the TUM RGB-D SLAM dataset [22]. The dataset contains the color and depth images of a Microsoft Kinect sensor along the ground-truth trajectory of the sensor. The data was recorded at full frame

rate (30 Hz) and sensor resolution (640x480). The ground-truth trajectory was obtained from a high-accuracy motion-capture system with eight high-speed tracking cameras (100 Hz).

We compare our system with ORB-SLAM2 [2] in RGB-D mode, a state-of-the-art SLAM system. We show the result of two sequences, freiburg2/walking_static and freiburg2/walking_halfsphere, in detail. For the sequence freiburg2/walking_static, the trajectory is shown in Figure 13. The camera of this data sequence basically remains stationary. The main dynamic object in the scene is a person, and the pixels occupied by it may be extracted with many features. The root-mean-square error (RMSE), mean error, median error, standard deviation, minimum error and maximum error of the squence shown in Table I, which is evaluated by an absolute trajectory error (ATE) [23]. The localization accuracy of our method in this sequence is slightly higher than that of ORB-SLAM2, both of methods maintain high accuracy in this dynamic environment. But as shown in Figure 13, there are multiple outliers during the tracking process by using ORB-SLAM2, its maximum error is much more greater than our methods.



(a) Ours



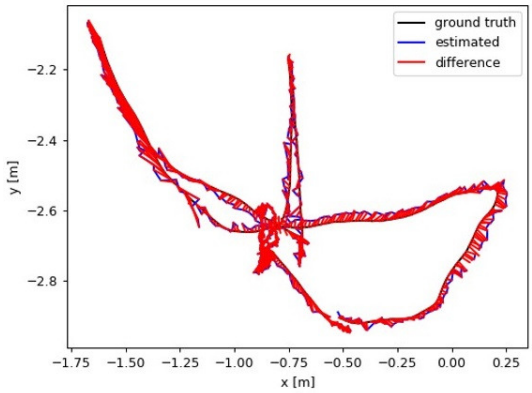
(b) ORB-SLAM2

Fig. 13. Trajectory in freiburg2/walking_static, compared with ORB-SLAM2.

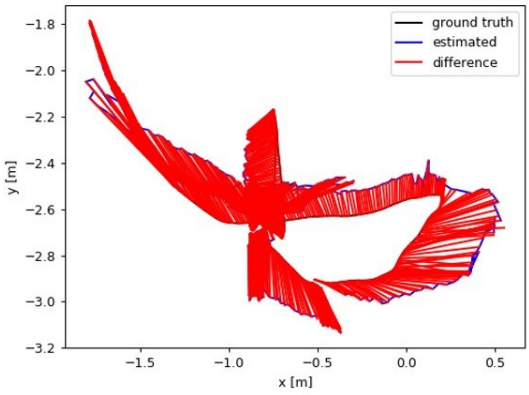
For the sequence freiburg2/walking_halfsphere, the trajectory is shown in Figure 14. In this data sequence, the camera is also moving in translation and rotation while the person is moving. As shown in Table II, the pose error of our method is much smaller compared with ORB-SLAM2, and it can still maintain a high accuracy in dynamic environment.

TABLE I
COMPARISON IN FREIBURG2/WALKING_STATIC

	ORB-SLAM2(cm)	OURS(cm)
RMSE	1.5780	**1.2880**
Mean Error	**1.1185**	1.1399
Median Error	**0.9807**	1.0383
Standard Deviation	1.0415	**0.5997**
Min Error	0.0769	**0.0739**
Max Error	11.9245	**3.2535**



(a) Ours



(b) ORB-SLAM2

Fig. 14. Trajectory in freiburg2/walking_halfsphere, compared with ORB-SLAM2.

B. Real-World Experiments

The sensor we use is ASUS Xtion pro, a RGB-D camera (60Hz) which can obtain both color and depth image. Motion capture device is used to record the camera's movement trajectory and is compared as the real value with the estimated value obtained by our method. The device we use is Raptor-12HS camera group from Motion Analysis, as shown in Figure

TABLE II
COMPARISON IN FREIBURG2/WALKING_HALFSPHERE

	ORB-SLAM2(cm)	OURS(cm)
RMSE	26.0688	**3.6537**
Mean Error	24.1605	**2.9837**
Median Error	22.8477	**2.4801**
Standard Deviation	9.7905	**2.1087**
Min Error	3.9979	**0.1523**
Max Error	64.2841	**3.2535**

15. The device can obtain high-resolution images with a resolution of 4096*3072 at a rate of 300Hz.

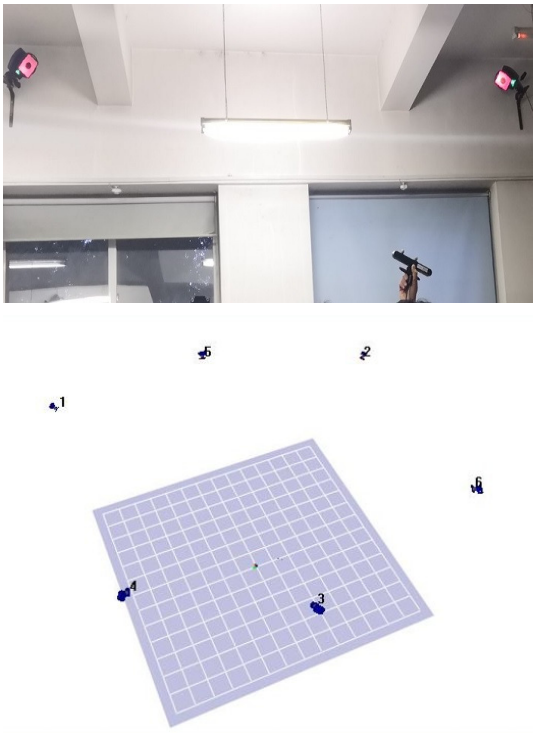
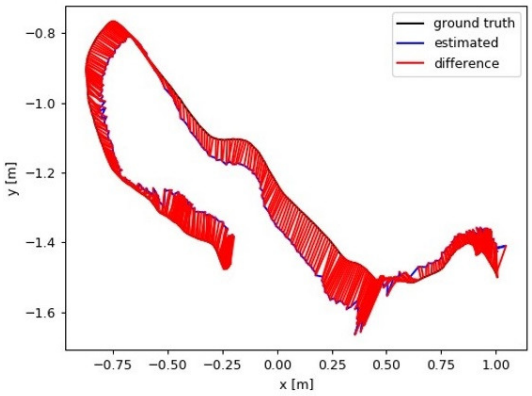


Fig. 15. The motion capture device

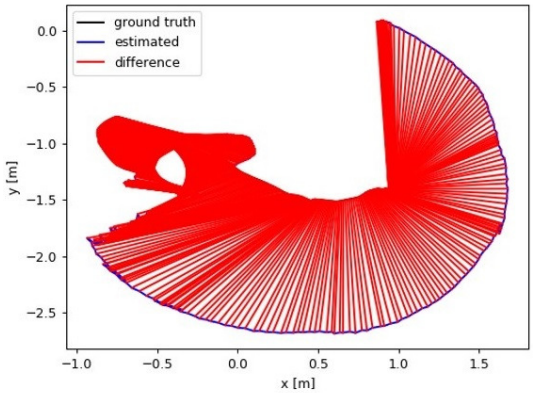
As shown in Table III and Figure 16, the pose error of our method is much smaller compared with ORB-SLAM2, our work can still maintain a high accuracy in real-world dynamic environment. The real-world dynamic environment and mapping result is shown in Figure 17, a clear and dense point cloud map can be consturcted.

TABLE III
COMPARISON IN REAL-WORLD ENVIRONMENT

	ORB-SLAM2(cm)	OURS(cm)
RMSE	96.0068	**9.4537**
Mean Error	90.2604	**8.4756**
Median Error	82.8477	**6.3805**
Standard Deviation	89.7805	**7.1097**
Min Error	23.6978	**3.1623**
Max Error	114.9381	**18.2635**



(a) Ours



(b) ORB-SLAM2

Fig. 16. Trajectory in real-world environment, compared with ORB-SLAM2.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a dynamic visual SLAM system which combines semantics segmentation and multi-view geometry to make it more robust for camera tracking and map building in dynamic environment using RGB-D camera. Our system can accurately track the camera in a dynamic environment and create a static, reusable scene map. We tested the system on public datasets and real-world environments. The test results show that the system is more robust and stable than the current advanced visual SLAM system in dynamic environment.

However, because time-consuming algorithms such as neural network and point cloud processing are used in our method, its real-time performance is greatly reduced. Next step, we will parallelize semantic segmentation and camera tracking and increase the optimization of the algorithm to improve its real-time performance as much as possible.

ACKNOWLEDGMENT

Our research is supported by National Natural Science Foundation of China (Grant No.61773139), the Foundation for Innovative Research Groups of the National Natural Science

(a) Real-world dynamic environment

(b) Mapping result

Fig. 17. Mapping result of real-world environment.

Foundation of China (Grant No.51521003), Shenzhen Science and Technology Research and Development Foundation (Grant No.JCYJ20190813171009236) and Shenzhen Science and Technology Program (Grant No.KQTD2016112515134654).

REFERENCES

[1] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.

[2] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[3] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 2007, pp. 1–10.

[4] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular slam in dynamic environments," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2013, pp. 209–218.

[5] S. Wangsiripitak and D. W. Murray, "Avoiding moving outliers in visual slam by tracking moving objects," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 375–380.

[6] S. Li and D. Lee, "Rgb-d slam in dynamic environments using static point weighting," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2263–2270, 2017.

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

[8] Y. Wang and S. Huang, "Motion segmentation based robust rgb-d slam," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*. IEEE, 2014, pp. 3122–3127.

[9] D.-H. Kim and J.-H. Kim, "Effective background model-based rgb-d dense visual odometry in a dynamic environment," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1565–1573, 2016.

[10] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving rgb-d slam in dynamic environments: A motion removal approach," *Robotics and Autonomous Systems*, vol. 89, pp. 110–122, 2017.

[11] L. Riazuelo, L. Montano, and J. Montiel, "Semantic visual slam in populated environments," in *2017 European Conference on Mobile Robots (ECMR)*. IEEE, 2017, pp. 1–7.

[12] R. Ambrus, J. Folkesson, and P. Jensfelt, "Unsupervised object segmentation through change detection in a long term autonomy scenario," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 1181–1187.

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[14] D. Galvez-Lpez and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. p.1188–1197, 2012.

[15] 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 *European conference on computer vision*. Springer, 2014, pp. 740–755.

[16] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 4, pp. 722–732, 2008.

[17] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.

[18] J.-L. Blanco, "A tutorial on se (3) transformation parameterizations and on-manifold optimization," *University of Malaga, Tech. Rep*, vol. 3, 2010.

[19] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7^th International Joint Conference on ArtificialIntelligence*, 1997.

[20] L. Quan, *Image-based modeling*. Springer Science & Business Media, 2010.

[21] A. Tremeau and N. Borel, "A region growing and merging algorithm to color segmentation," *Pattern Recognition*, vol. 30, no. 7, pp. 1191–1203, 1997.

[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[23] J. Sturm, N. Engelhard, F. Endres, and W. Burgard, "A benchmark for the evaluation of rgb-d slam systems," in *IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2012.