Visual SLAM Based on Dynamic Object Removal*

Hanjie Liu, Guoliang Liu and Guohui Tian
School of Control Science and Engineering
Shandong University
Jinan, Shandong, China
liuguoliang@sdu.edu.cn

Shiqing Xin
School of Computer Science and Technology
Shandong University
Qingdao, Shandong, China
xinshiqing@sdu.edu.cn

Ze Ji
School of Engineering
Cardiff University
Cardiff, CF24 3AA, UK
jiz1@cardiff.ac.uk

***Abstract*— Visual simultaneous localization and mapping (SLAM) is the core of intelligent robot navigation system. Many traditional SLAM algorithms assume that the scene is static. When a dynamic object appears in the environment, the accuracy of visual SLAM can degrade due to the interference of dynamic features of moving objects. This strong hypothesis limits the SLAM applications for service robot or driverless car in the real dynamic environment. In this paper, a dynamic object removal algorithm that combines object recognition and optical flow techniques is proposed in the visual SLAM framework for dynamic scenes. The experimental results show that our new method can detect moving object effectively and improve the SLAM performance compared to the state of the art methods.**

***Index Terms*— Visual SLAM; dynamic scenes; optical flow method; object detection.**

I. Introduction

Visual simultaneous localization and mapping (SLAM) refers to the synchronization construction of structural map and self-localization of the robot by using the visual sensor to sense the surrounding environment. In recent years, with the wide application of robots in social service, public security, disaster relief, etc., the traditional geometric visual SLAM algorithm can no longer meet the high-level task requirements of robots, which need to perform interactive and cooperative tasks from the semantic level. Especially when there are dynamic objects in the environment, the traditional SLAM algorithm based on geometric vision has large error in estimating robot pose.

Dynamic objects can provide dynamic visual feature points, while the current mainstream SLAM algorithm uses static feature point to estimate pose and map reconstruction, so dynamic targets need to be removed to reduce its impact on SLAM algorithm. Li [1] et al. proposed to use a static weight of visual feature point to depict whether the feature belongs to the dynamic objects, which can be updated according to the estimated relative pose transformation between consequence images. Wang et al. [2] proposed a motion segmentation based SLAM algorithm, which uses optical flow to classify the matching features in adjacent frames. Sun et al. [3] proposed an image difference method based on self-motion compensation to detect and eliminate moving objects. Lee [4] et al. proposed a node grouping method to prune the false connected nodes in the pose graph according to the grouping rules with noise covariances, which can reduce the chance of false loop closing. Wang [5] et al. use mathematical models and geometric constraints to detect moving objects which is then incorporated into SLAM process as a data filtering process. Zou et al. [6] introduced inter-camera pose estimation and inter-camera mapping to deal with dynamic objects in the localization and mapping process. The dynamic points are recognized according to reprojection error. Fang et al. [7] used the improved optical flow method and Kalman filter for dynamic object tracking. Wang et al. [8] proposed a dense moving object segmentation method for robust dense SLAM.

To solve the problem of disturbance caused by dynamic object to visual SLAM, this paper designs a front-end visual odometer method for dynamic object removal. Our method integrates dynamic object detection and identification module into the visual SLAM system [9], which can reduce the impact of dynamic objects. The motion characteristic of the object is first recognized by the state of the art deep learning based object detection and recognition method, and then the optical flow is employed to valid the moving state of the object. Experimental results show that the proposed method in this paper has better pose estimation performance compared with the state of the art visual SLAM algorithms in case of dynamic scenarios.

II. The Proposed Method

Normally, visual SLAM consists of the following four modules: visual odometer, back-end optimization, loop detection and mapping. Visual odometer is mainly responsible for motion prediction between images. Back-end optimization optimizes the prediction of visual odometer to obtain relatively accurate transformation between image frames. The closed-loop detection detects whether the camera has ever been to the current position before. If it has been to the current position, it can optimize the posture again through

*This work is partially supported by the National Key R&D Program of China (#2018YFB1306504), National Natural Science Foundation of China (#61603213, #91748115), Young Scholars Program of Shandong University (#2018WLJH71), the Fundamental Research Funds of Shandong University, and the Taishan Scholars Program of Shandong Province. (Corresponding author: Guoliang Liu).

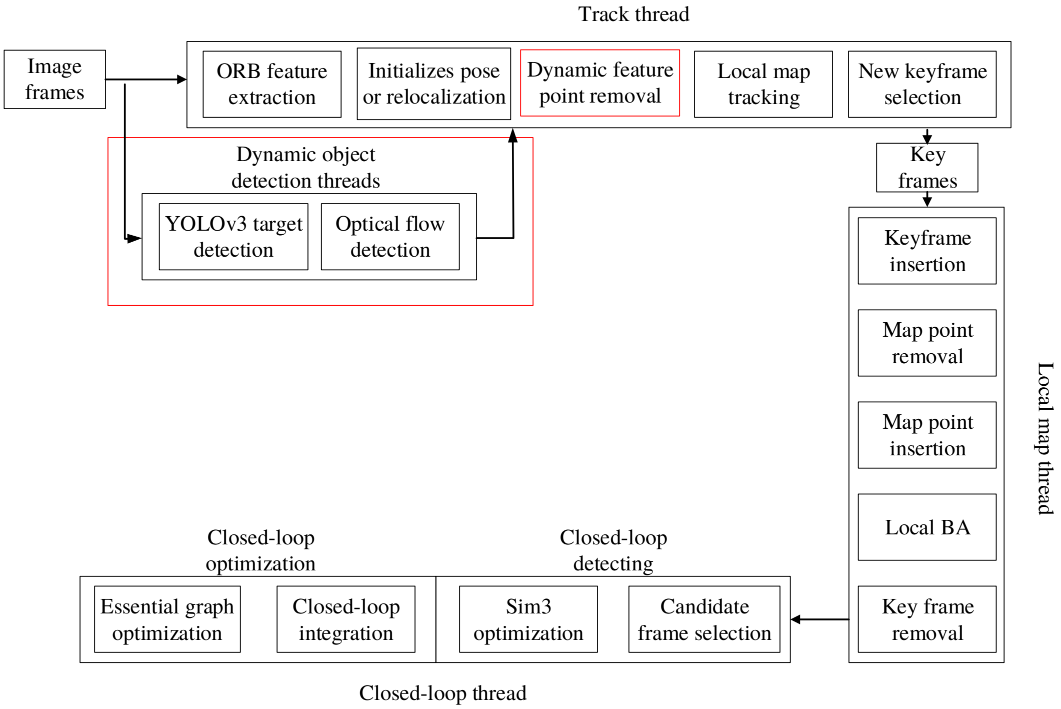


Fig. 1. The main modules of proposed visual SLAM for dynamic environment using object detection/recognition and optical flow method.

the back-end optimization to reduce the motion error. Map construction is to generate a specific environment map based on the estimated camera positions. The above four modules constitute the basic framework of visual SLAM.

In order to reduce the impact of the dynamic objects for visual SLAM, a robust moving object detection method is required. Here we propose to use a deep learning method to recognize the motion characteristics of the objects first and then use an optical flow method to check whether the object is moving. The overall work flow of the proposed method is shown in Fig.1.

A. *Object detection and recognition based on YOLO*

Object detection and recognition is to recognize object category in the image and determine their positions. If the category of the object has movable characteristics, e.g., bicycle, car, human, etc., we can have an initial guess about the moving objects in the current frame. Recently, deep learning has made rapid progress in the direction of object detection and recognition. Compared with traditional methods, object detection methods based on deep learning have stronger robustness to complex environment conditions, such as illumination changes and occlusion. Currently, the object detection methods related to deep learning mainly have two directions: two-stage method and one-stage method. For the two-stage method, the first step is to generate candidate proposals, and the second step is to adjust and classify the proposals, such as the R-CNN[18], Fast R-CNN[19], Faster R-CNN [20]. The one-stage method omits the selection of candidate proposals and directly predicts the category and location of the target, such as YOLOv3 and SSD [21]. Compared with the two-stage methods, the one-stage methods are more efficient. Therefore, we here use YOLOv3 for object detection and recognition. YOLOv3 is an end-to-end target detection algorithm based on the darknet network architecture. By modeling the detection situation into regression, the position and attribution of the rectangular box of the object can be easily predicted as shown in Fig.2.

The detection process for YOLOv3 consists of the following steps. First, to meet the requirements of the network architecture, the input image is adjusted to the specified scale, which is then divided into $n \times n$ grids. Each grid is responsible for detecting the object that falls on the central point of this grid. Finally, in order to prevent multiple grids responding to the same object, YOLOv3 uses nonmaximal suppression to eliminate unwanted results. Non-maximum suppression first obtains the object bounding box with the highest confidence, and then calculates the IOU between other object boxes and this object box. When the IOU is larger than a certain threshold, the object box has lower confidence is eliminated. Finally, the object box has the highest confidence and no overlap with others is obtained.

B. *Optical flow based moving object detection*

Optical flow describes the motion relationship between two adjacent frames by the correlation of pixels. Optical

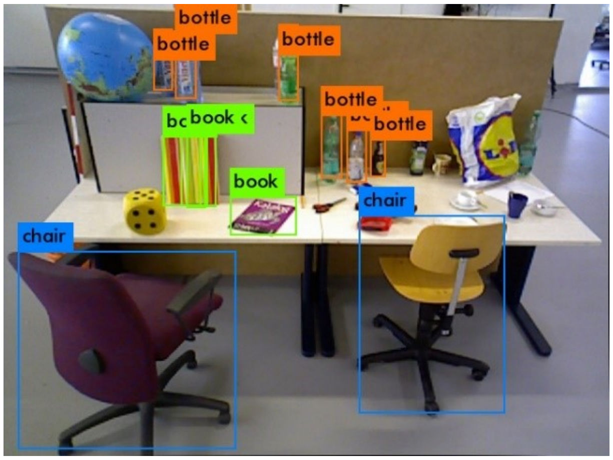


Fig. 2. YOLOv3 for object detection and recognition.

flow method does not need feature descriptor calculation and feature matching for pixel tracking, which has high real-time performance. In this paper, a motion consistency detection algorithm based on Lucas-Kanade optical flow is proposed to further classify dynamic feature points. Lucas-Kanade optical flow algorithm [22] first assumes that the image obtained by the camera changes with time, and then the image can be regarded as a function of time $I(t)$. For a pixel with (x, y) coordinates, its grayscale value is $I(x, y, t)$. Assume that the horizontal and vertical coordinates of a fixed point in the 2D space are x and y respectively at time on the image, and their coordinates also change with time. The purpose of optical flow method is to predict the position of the 2D fixed point in the image at different time.

Feature point tracking based on optical flow method first assumes that the gray of the pixel does not change for adjacent frames, which means the gray value of the same spatial point always remains consistent on the image plane during a short time period. If a pixel is at the position of (x, y) at t and $(x+dx, y+dy)$ at $t+dt$, then the motion of the pixel satisfies:

$$I(x, y, t) = I(x+dx, y+dy, t+dt) \tag{1}$$

Assuming that the motion between two image frames is relatively small, we can get

$$I(x+dx, y+dy, t+dt) \approx I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt \tag{2}$$

Combining (1) and (2), we can get:

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0 \tag{3}$$

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} = -\frac{\partial I}{\partial t} \tag{4}$$

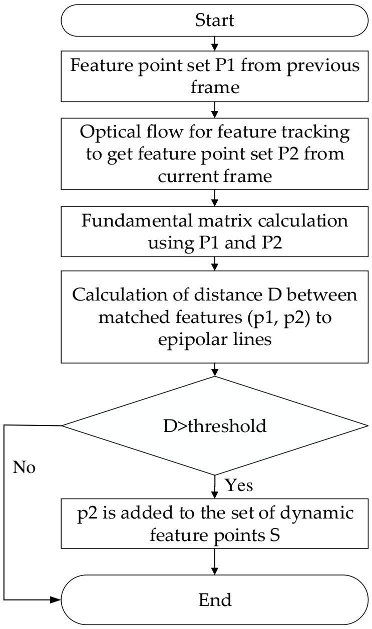

Start
Feature point set P1 from previous frame
Optical flow for feature tracking to get feature point set P2 from current frame
Fundamental matrix calculation using P1 and P2
Calculation of distance D between matched features (p1, p2) to epipolar lines
D>threshold
No
Yes
p2 is added to the set of dynamic feature points S
End

Fig. 3. Flow chart of motion consistency detection algorithm.

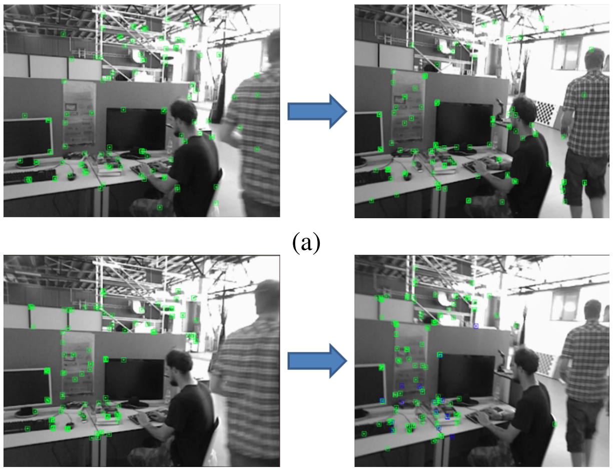
where $\partial I/\partial x$ is the gradient of the gray value of this point along the x axis, $\partial I/\partial y$ is the gradient of the gray value of this point along the y axis, dx/dt is the velocity of this point along the x axis, and dy/dt is the velocity of this point along the y axis. If dx/dt is u, dy/dt is v, $\partial I/\partial x$ is I_x, $\partial I/\partial y$ is I_y, and $\partial I/\partial t$ is I_t, then (3) and (4) can be written as:

$$\begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t \tag{5}$$

To calculate u and v, the lucas-kanade optical flow algorithm assumes the same pixel motion within the image block. Finally, through multiple iterations, the motion of pixels in the image can be obtained, so as to realize the tracking of pixel points.

C. Motion consistency detection algorithm

In order to detect dynamic feature points in images, a motion detection method based on optical flow method is proposed. The algorithm first obtains the matching feature point pairs by optical flow method and calculates the fundamental matrix with the matching feature point pairs. Then, the corresponding polar line of the feature point is calculated using the fundamental matrix and the position of feature point. When the distance between the feature point and the polar line is greater than a certain value, it is classified as a dynamic feature point. The detailed algorithm process is shown in Fig.3. Supposing p_1 and p_2 is a pair of matching feature point, their homogeneous coordinates are shown as follows:



(a)

(b)

Fig. 4. (a) the tracked original ORB features and (b) tracked features after dynamic feature point removal.

$$\begin{cases} p_1 = [u_1, v_1, 1] \\ p_2 = [u_2, v_2, 2] \end{cases} \tag{6}$$

where u and v are the corresponding horizontal and vertical coordinates of pixels. Then the epipolar line I_1 corresponding to p_1 is:

$$I_1 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Fp_1 \tag{7}$$

where F represents the corresponding fundamental matrix, so the distance d from point p_2 to the epipolar line I_1 is:

$$d = \frac{|p_2^T F p_1|}{\sqrt{\|X\|^2 + \|Y\|^2}} \tag{8}$$

If the distance d is greater than a certain threshold, p_2 is classified to be a dynamic feature point.

According to the proposed method, dynamic feature points in the images can be removed effectively, and one example is shown in Fig.4, where the feature points on the moving person have been removed.

III. Results

To verify the visual SLAM effect based on dynamic object removal proposed in this paper, TUM rgb-d data set [23] was used to test. TUM rgb-d data set contains rgb-d image sequences of some dynamic scenes, and has the accurate position and posture reference information corresponding to each image, which is very suitable for verifying the effect of the algorithm in this paper.

There are two main evaluation criteria for visual SLAM front-end odometer: relative pose error (RPE) and absolute trajectory error (ATE), where the relative posture error represents the local accuracy of the measured trajectory within a certain time interval, and the absolute trajectory error directly calculates the difference between the real coordinates and the estimated coordinates. Here we define the predicted trajectory and the real trajectory are P and Q respectively. The transformation matrix T is obtained through singular value decomposition to align the predicted trajectory and the real trajectory, and then the pose error is calculated. Let $E_i = Q_i^{-1} T P_i$, where Q_i represents the real trajectory of the i_{th} key frame, P_i represents the predicted trajectory of the i_{th} key frame, and then the ATE is defined as:

$$RMSE(E_{1:n}) = \sqrt{\frac{\sum_i^n \|trans(E_i)\|^2}{n}} \tag{9}$$

The accuracy performance of the algorithm can be given by the ATE. However, ATE can only evaluate the translation error of the algorithm. In order to find the rotation error of the algorithm and to evaluate the drift error of the visual odometer in a period of time, the measurement criterion of RPE can be adopted. Assuming that the relative pose between real postulates Q_i and $Q_{i+\Delta t}$ is $\Delta Q_{i,\Delta t} = Q_i^{-1} Q_{i+\Delta t}$, and the relative pose between predicted poses P_i and $P_{i+\Delta t}$ is $\Delta P_{i,\Delta t} = P_i^{-1} P_{i+\Delta t}$, let $F_i = \Delta Q_{i,\Delta t}^{-1} \times \Delta P_{i,\Delta t}$, then the RPE is defined as

$$RMSE(E_{1:n}, \Delta t) = \sqrt{\frac{\sum_i^m \|trans(F_i)\|^2}{m}} \tag{10}$$

This paper selected dynamic scenarios in TUM RGB-d to evaluate the proposed algorithm. The comparison to the original ORB-SLAM2 is shown in Table.I to Table.III. It can be seen that the performance of our algorithm in this paper outperforms ORB-SLAM2 in the dynamic environments, which indicates that our visual odometer based on dynamic object removal is effective. In low motion scenarios, such as fr3_sitting_static sequence, because there is no obvious object motion, ORB-SLAM2 has similar performance to ours.

In order to visualize the comparison results between the proposed algorithm and ORB-SLAM2, the fr3_walking_half sequence is taken as an example to draw absolute trajectory error curve and relative posture error curve, as shown in Fig.5 and 6. It can be seen that the absolute trajectory error and relative pose error of our method are much smaller than ORB-SLAM2, which verifies that the visual odometer based on dynamic object removal in this paper has better performance compared with ORB-SLAM2 in dynamic environments.

Meanwhile, this paper compares our visual SLAM algorithm with the current state of the art methods in dynamic scenarios as shown in Table IV. The experimental results show that our method has higher pose estimation accuracy compared to other methods, which prove our claims that the combination of object recognition and optical flow can

TABLE I

COMPARISON OF ABSOLUTE TRAJECTORY ERRORS

Sequences	$OURS$			$ORB-SLAM2$		
	$RMSE$	$Mean$	$Median$	$RMSE$	$Mean$	$Median$
$Fr3\_walking\_xyz$	0.0163m	0.014m	0.0122m	0.5771m	0.5162m	0.4602m
$Fr3\_walking\_static$	0.0105m	0.0073m	0.0058m	0.0452m	0.027m	0.0136m
$Fr3\_walking\_rpy$	0.0417m	0.0298m	0.0217m	0.8678m	0.7311m	0.841m
$Fr3\_walking\_half$	0.0311m	0.0261m	0.0224m	0.5166m	0.4555m	0.3904m
$Fr3\_sitting\_static$	0.0059m	0.0051m	0.0046m	0.0086m	0.0076m	0.007m

TABLE II

COMPARISON OF RELATIVE ERRORS OF ROTATION

Sequences	$OURS$			$ORB-SLAM2$		
	$RMSE$	$Mean$	$Median$	$RMSE$	$Mean$	$Median$
$Fr3\_walking\_xyz$	0.6299°	0.4974°	0.4102°	6.0955°	3.6146°	1.1380°
$Fr3\_walking\_static$	0.3106°	0.2512°	0.2169°	1.0154°	0.6195°	0.3143°
$Fr3\_walking\_rpy$	1.4067°	1.0363°	0.7286°	7.5397°	5.3755°	3.1801°
$Fr3\_walking\_half$	0.7872°	0.6872°	0.6027°	6.0396°	3.2201°	1.0818°
$Fr3\_sitting\_static$	0.2618°	0.2354°	0.2207°	0.2859°	0.2565°	0.2463°

TABLE III

COMPARISON OF RELATIVE TRANSLATION ERRORS

Sequences	$OURS$			$ORB-SLAM2$		
	$RMSE$	$Mean$	$Median$	$RMSE$	$Mean$	$Median$
$Fr3\_walking\_xyz$	0.0205m	0.0177m	0.016m	0.3189m	0.1883m	0.0585m
$Fr3\_walking\_static$	0.015m	0.0105m	0.0083m	0.0562m	0.0316m	0.0128m
$Fr3\_walking\_rpy$	0.0639m	0.0461m	0.0322m	0.3817m	0.2681m	0.1464m
$Fr3\_walking\_half$	0.033m	0.0277m	0.0241m	0.2908m	0.1482m	0.0417m
$Fr3\_sitting\_static$	0.0073m	0.0064m	0.0058m	0.0095m	0.0085m	0.0077m

TABLE IV

COMPARISON RESULTS OF ABSOLUTE TRAJECTORY ERROR OF VISUAL SLAM METHODS

Sequences	Li[1]	Wang[2]	Sun[3]	Ours
$Fr3\_walking\_xyz$	0.0600m	0.0400m	0.0930m	0.0163m
$Fr3\_walking\_static$	0.0260m	0.0240m	0.0660m	0.0105m
$Fr3\_walking\_rpy$	0.1790m	0.0760m	0.1330m	0.0417m
$Fr3\_walking\_half$	0.0490m	0.0550m	0.1250m	0.0311m

remove the features that belong to dynamic objects effectively and hence improve the performance of visual SLAM.

IV. CONCLUSION

To improve the performance of visual SLAM for mapping a dynamic environment, this paper proposes a dynamic object removal method combining deep learning based object detection and recognition with the optical flow based motion consistence checking. The proposed method can detect the moving features effectively, and achieve higher pose estimation accuracy compared to the state of the art methods on the public datasets, e.g., original ORB-SLAM2.

REFERENCES

[1] 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.

[2] Y. Wang and S. Huang, ”Motion segmentation based robust RGB-D SLAM,” Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, pp.3122-3127, 2014.

[3] Y. Sun, M. Liu, and M.Q.-H.Meng, ”Improving RGB-D slam in dynamic environments: A motion removal approach”, Robotics and Autonomous Systems, pp.110-122, 2017.

[4] D. Lee and H. Myung, ”Solution to the SLAM problem in low dynamic environments using a pose graph and an RGB-D sensor”, Sensors,vol.14, no.7, pp.12467-12496, 2014.

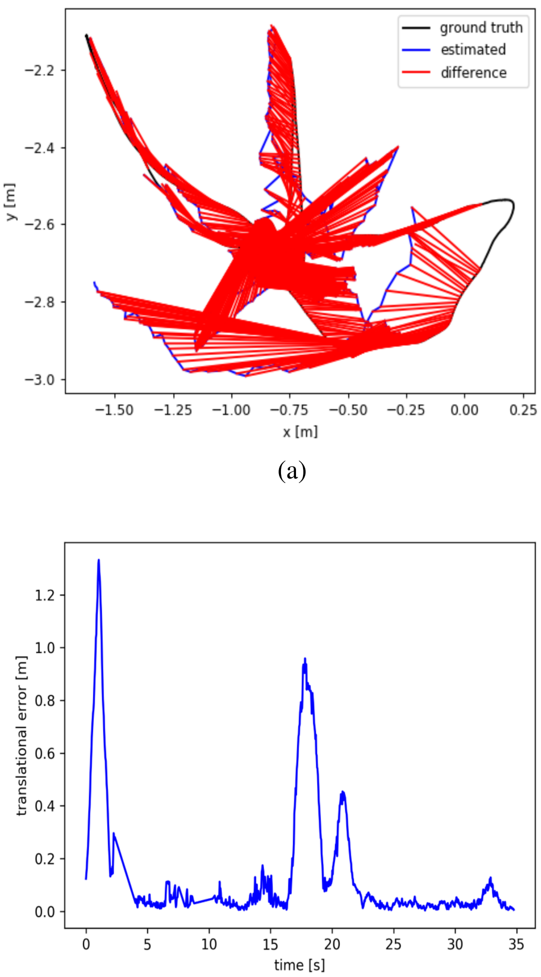
[5] R. Wang, W. Wan, Y. Wang, and K. Di, ”A New RGB-D SLAM Method with Moving Object Detection for Dynamic Indoor Scenes”, Remote Sensing, vol.11, no.10, p.1143, 2019.

[6] D. Zou and P. Tan, ”CoSLAM: Collaborative Visual SLAM in Dynamic Environments,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.35, no.2, pp.354-366, 2013.

[7] Y. Fang and B. Dai, ”An improved moving target detecting and tracking based on Optical Flow technique and Kalman filter,” 4th International Conference on Computer Science & Education, Nanning, pp.1197-1202, 2009.

[8] Y. Wang and S. Huang, ”Towards dense moving object segmentation based robust dense RGB-D SLAM in dynamic scenarios,” 13th International Conference on Control Automation Robotics & Vision, Singapore, pp.1841-1846, 2014.

[9] R. Mur-Artal and J. D. Tards, ”ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras,” in IEEE Transactions on Robotics, vol.33, no.5, pp.1255-1262, 2017.



(a)

(b)

Fig. 5. The (a) absolute trajectory error (ATE) and (b) relative pose error (RPE) of the original ORB-SLAM2 algorithm.

[10] J. Redmonand, A. Farhadi, "Yolov3: An incremental improvement," arXiv:1804.02767. [Online]. Available: https://arxiv.org/abs/1804.02767, 2018.

[11] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, vol.60, no.2, pp.91-110, 2004.

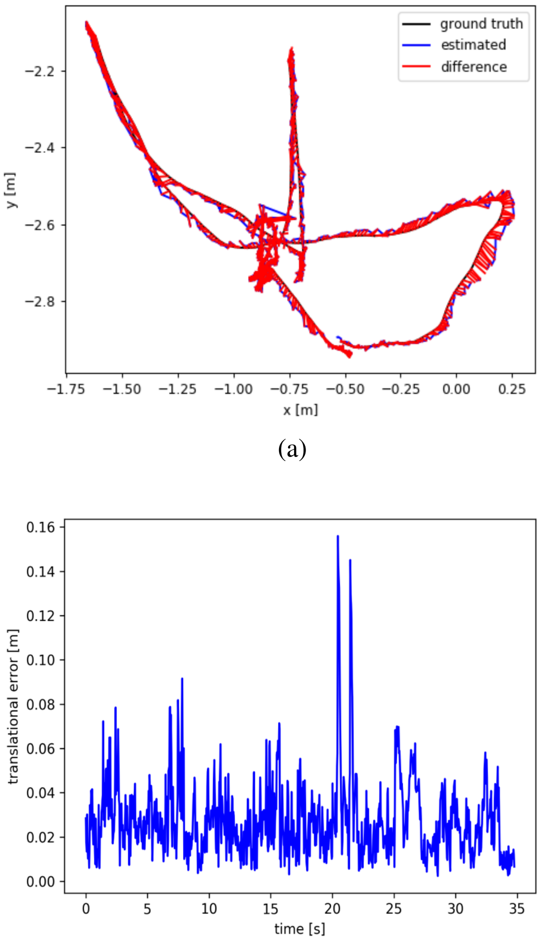
[12] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," Computer Vision and Image Understanding, vol.110, no.3, pp.346-359, 2008.

[13] R. K. Ummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in IEEE International Conference on Robotics and Automation, Shanghai, pp. 3607-3613. 2011.

[14] D. Galvez-Lpez and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," in IEEE Transactions on Robotics, vol.28, no.5, pp.1188-1197, 2012.

[15] Sivic and Zisserman, "Video Google: a text retrieval approach to object matching in videos," Proceedings Ninth IEEE International Conference on Computer Vision, Nice, France, vol.2, pp. 1470-1477, 2003.

[16] S.E. Robertson, "Understanding Inverse Document Frequency: On Theoretical Arguments for IDF," Journal of Documentation, vol.60, no.5, pp.503-520, 2004.



(a)

(b)

Fig. 6. The (a) absolute trajectory error (ATE) and (b) relative posture error (RPE) of our method.

[17] K. Mikolajczyk, and C. Schmid, "Scale & Affine Invariant Interest Point Detectors," International Journal of Computer Vision, vol.1, no.60, pp.63-86, 2004.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp.580-587, 2014.

[19] R. Girshick. "Fast R-CNN". In IEEE International Conference on Computer Vision, 2015, pp. 1440-1448.

[20] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks," In NIPS, pp.91-99, 2015.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: Single shot multibox detector," arXiv:1512.02325, 2015.

[22] B. D. Lucas, T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", Proceeding IJCAI'81 Proceedings of the 7th international joint conference on Artificial intelligence, pp.674-679, 1981.

[23] A. Handa, T. Whelan, J. McDonald and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," IEEE International Conference on Robotics and Automation, Hong Kong, pp.1524-1531, 2014.