Proceedings of the 2018 IEEE
International Conference on Robotics and Biomimetics
December 12-15, 2018, Kuala Lumpur, Malaysia

机器人使用optical flow在动态环境中的accurate localization

An Accurate Localization Scheme for Mobile Robots Using Optical Flow in Dynamic Environments

Jiyu Cheng, Yuxiang Sun, Wenzheng Chi, Chaoqun Wang, Hu Cheng and Max Q.-H. Meng, *Fellow*, *IEEE*

***Abstract*— Visual Simultaneous Localization and Mapping (Visual SLAM) has been studied for the past years and many state-of-the-art algorithms have been proposed with rather satisfactory performance in static scenarios. However, in dynamic scenarios, off-the-shelf Visual SLAM algorithms cannot localize the robot very accurately. To address this problem, we propose a novel method that uses optical flow to distinguish and eliminate dynamic feature points from extracted ones by using the RGB images as the only input. The static feature points are fed into the Visual SLAM algorithm for camera pose estimation. We integrate our method with the ORB-SLAM system and validate the proposed method with challenging dynamic sequences from the TUM dataset. The entire system can run in real time. Qualitative and quantitative evaluations demonstrate that our method significantly improves the performance of the Visual SLAM in dynamic scenarios.**

I. Introduction

Recent decades have witnessed the great development of Visual SLAM. As a fundamental technology of robotics, Visual SLAM is widely used in many fields. However, Visual SLAM can achieve impressive performance only with the assumption that the environments are static, which limits the application of Visual SLAM. Dynamic objects will degrade the performance of the localization and even induce the system into a failure. One reason is that for visual odometry, the trajectory of the camera is estimated through correspondences between two consecutive frames. Therefore, the motion of the objects will cause errors in the computation of the camera motion.

One solution to solve the dynamic SLAM problem is the information fusion from different sensors [1]. However, information fusion requires additional sensors, which is not a cost-effective way. Another popular solution is to use the depth information [2], [3]. The disadvantage is that the RGB-D cameras can hardly work in outdoor scenarios. In most cases, a monocular camera is more favorable. Therefore, in this paper, we focus on the problem on how to eliminate the dynamic factors using only the RGB images.

Low cost, easy calibration and portability make the monocular camera a very popular sensor for Visual SLAM [4]–[6]. Many Visual SLAM systems use features such as Scale-invariant Feature Transform (SIFT) [7], Oriented FAST and Rotated BRIEF (ORB) [8] as intermediate representation for the raw sensor measurements [9]. This kind of system is known as feature based Visual SLAM. In a typical feature-based Visual SLAM system, a feature matcher firstly finds feature correspondences between two consecutive frames. Then, transformation estimation algorithms compute the transformation matrix from the adjacent poses. If the environment is static, the transformation matrix between the two frames represents the camera motion. However, in dynamic scenarios, if feature points are extracted from the dynamic objects, object motion will bring noise into the correspondence. As a result, the transformation matrix will fail to represent the camera motion. To address this problem, we propose to distinguish and eliminate dynamic feature points from input frames using the optical flow in a feature-based monocular SLAM system. The inputs of our system are only RGB images, and only a monocular camera is required, which is available for almost all robot platforms. A novel method to distinguish dynamic points using optical flow is introduced first, and an efficient dynamic point elimination strategy is then applied. The correspondences between the two consecutive frames are reliably selected to estimate the camera motion. We validate our algorithms in public TUM dataset. Experimental results demonstrate the impressive performance of our method. The entire system can run in real time, which is a very important criterion in robotics.

The novelty of our work is summarized as follows:

1. We proposed a novel method to distinguish dynamic points using the optical flow in real time; and
2. We integrated the proposed method into a feature-based monocular SLAM system. The performance has been significantly improved in dynamic scenarios.

The remainder of this paper is organized as follows. Section II presents a review of related work. Section III explains the details of our method. Section IV discusses the experimental results. We draw some conclusions and state the future work in the last section.

II. Related Work

To solve the Visual SLAM problems in dynamic scenarios, researchers have done many state-of-the-art works, which can be roughly divided into two categories.

The first category is information fusion [10]–[12]. With more sensor data, camera poses can be estimated more robustly and accurately. In [10], Bloesch et al. combine complementary information from vision and inertial sensors to enable robust performance for high dynamic scenarios.

This project is partially supported by RGC GRF grants CUHK 415512 and CUHK 415613, CRF grant CUHK 6CRF13G, and CUHK VC discretional fund #4930765, awarded to Prof. Max Q.-H. Meng.

Jiyu Cheng, Yuxiang Sun, Wenzheng Chi, Chaoqun Wang, Hu Cheng and Max Q.-H. Meng are with the Robotics, Perception and Artificial Intelligence Lab, Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T. Hong Kong SAR, China. email:{jycheng, yxsun, wzchi, cqwang, hcheng, qhmeng}@ee.cuhk.edu.hk

978-1-7281-0377-8/18/$31.00 ©

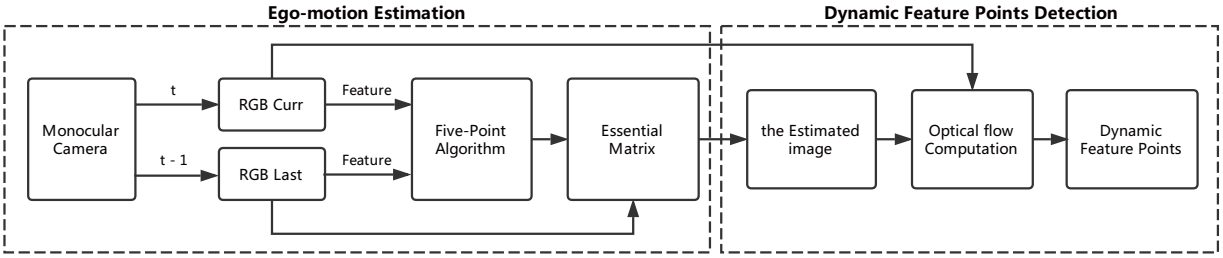


Fig. 1: The overview of the proposed method. It consists two modules: `Ego-motion Estimation` and `Dynamic Feature Points Detection`. The arrows represent the directions of data flow. The `Estimation` module is to estimate the camera ego-motion between two consecutive frames. The `Detection` module is to distinguish dynamic feature points based on the optical flow value for each feature point.

The inertial measurem re used to propagate the state of the filter, and the visual information is taken for the filter update. Usenko et.al [11] also use Inertial Measurement Unit (IMU) as an additional sensor. They use an energy function to combine photometric and inertial information. Through minimizing the energy function, camera pose, velocity and IMU bias are simultaneously estimated. Kim et al. [12] use an IMU sensor to compensate rotation of feature points. Then based on motion vectors of these points, static points can be selected and fed into the visual odometry process. Finally, the camera trajectory is precisely estimated. Information fusion is a reasonable way to solve the proposed problem, while in some cases additional sensors are not available and processing fusing data may be time-consuming.

The second category is based on o rejection. Researchers [2], [3], [13] often use depth information to detect eliminate dynamic elements. Wang et al. [13] cluster all points from each RGB image based on the optical flow values. Then dynamic objects are excluded through energy function minimization. After incorporating this method, a dense SLA tem works better in dynamic scenarios. Sun et al. [2] u -motion compensation to get a differencing frame. Then a particle filter-based tracking helps to segment dynamic objects based on vector quantization. Li et al. [3] e only depth edge points to conduct visual odometry. They give each point a static weight which indicates the likelihood of one point being static. Then the static weight is added into an intensity assisted iterative closest point method to perform the registration task. The proposed method can work in real time. Using depth information can efficiently detect the dynamic object, however, RGB-D sensors cannot work in outdoor scenarios.

All the methods above need either an additional sensor or depth information. However, in real scenes, usually one monocular camera is available. To the best of our knowledge, our method is the first one that improves the performance of localization in dynamic scenarios based on only RGB images.

III. Method

A. Problem Statement

Bundle adjustment (BA) [14] is an efficient method to accurately estimate camera poses as well as a sparse reconstruction, given a strong work of matches and good initial guesses. The problem to be solved in BA can be formulated as follows:

$$\arg\min_{a_j,\ b_i} \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}(X_{ij} - Q(a_j, b_i))^2, \tag{1}$$

where we assume that n 3D points can be observed in m views and X_{ij} is projection of point i in image j. If point i can be projected on image i, w_{ij} is 1 otherwise w_{ij} is set to 0. $Q(a_j, b_i)$ is predicted projection of point i on image j.

In ORB-SLAM, a local map is used to enhance the robustness and accuracy of camera pose estimates by providing 3D points for tracking. Current camera pose is initialized by motion model or feature matching with a key frame. Then bundle adjustment is applied to optimize the camera pose based on a local map. In dynamic scenarios, dynamic points can change the locations in each frame compared with those in static scenarios. Let vector V_{ij} denote changes of the location, and we modify (1) as follows:

$$\arg\min_{a_j,\ b_i} \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}(X_{ij} + \beta_{ij}V_{ij} - Q(a_j, b_i))^2, \tag{2}$$

where if corresponding point of i on image j is dynamic, β_{ij} is set to 1, otherwise, it is set to 0.

For this problem, if there are no motion model priors, the random value of V_{ij} will cause uncertainty for optimization of camera pose. As a result, camera poses cannot be estimated robustly and accurately.

B. Overview

Fig. 1 shows an overview of our proposed method. There are two modules in our method. The first module called `Ego-motion Estimation` is to estimate the camera

ego-motion between two consecutive frames. At the beginning of our method, the two consecutive images are captured and denoted as RGB Curr and RGB Last, as shown in Fig. 1. Then we adopt Five-Point Algorithm to estimate the motion of the camera from the last image to the current image by computing the essential matrix. We then multiply the last image with the estimated transformation matrix to get a new image which we call the estimated image. In this case, points in the estimated image are converted to the same coordinate system as those in the current image. The second module, `Dynamic Feature Points Detection`, calculates optical flow value for each feature point extracted from the current image between the current image and the estimated image and detects dynamic feature points for the current image based on optical flow values. Then static points are used for further camera pose estimation. Fig. 2 shows an experimental result of our method.

C. Ego-motion Estimation

The `Estimation` module takes two consecutive frames as the input: RGB Curr and RGB Last. Then Five-Point Algorithm [15] is used to compute the essential matrix E. Once the essential matrix E is known, R, t, and the camera matrices can be recovered from it. We use T to denote the transformation matrix, and use u and v to denote two pixel coordinates of the matching correspondence. Warp the point u with T, and we compute the reprojection error:

$$\xi = |Tu - v|, \tag{3}$$

where $|\cdot|$ represents the Euclidean distance.

In the feature extraction stage, we use the original extraction process in ORB-SLAM. The time of ORB feature points is far less than that of SIFT or SURF, which is very meaningful in real time implementations.

D. Optical flow-based Detection

Optical flow is an algorithm to detect object motion which has been extensively studied over the past decades.

In `Detection` module, we use Lucas-Kanade [16] to compute optical flow values of feature points extracted from the current image. A predefined tolerance τ is used to determine which point is dynamic by the following inequalities:

$$\begin{cases} d > \tau, \text{ if } f_i \in F_{dynamic}, \\ d < \tau, \text{ if } f_i \in F_{static}, \end{cases} \tag{4}$$

where d = $\sqrt{d_x^2 + d_y^2}$ is 2-norm of the flow vector for feature point f_i, and $F_{dynamic}$ and F_{static} are dynamic and static feature points sets, respectively. Through `estimation`, we get the perspective transformation matrix between the last image and the current image. Then we multiply the last image with the estimated transformation matrix to get the estimated image. And points in the estimated image are converted to the same coordinate system as those in the current image. We compute the optical flow values of feature points extracted from the current frame. Then we distinguish dynamic points based on the optical flow values and use a selection strategy to determine which point is used for camera pose estimation. For feature-based Visual SLAM, one big challenge is that when the environment is featureless for instance there is a white wall in the environment, which can lead to track-lost of camera pose. This is very common in Visual SLAM. Fortunately, these issues are not critical in most man-made environments. We sort the points by the optical flow values and remove the points with a higher value (up to a maximum of N points).

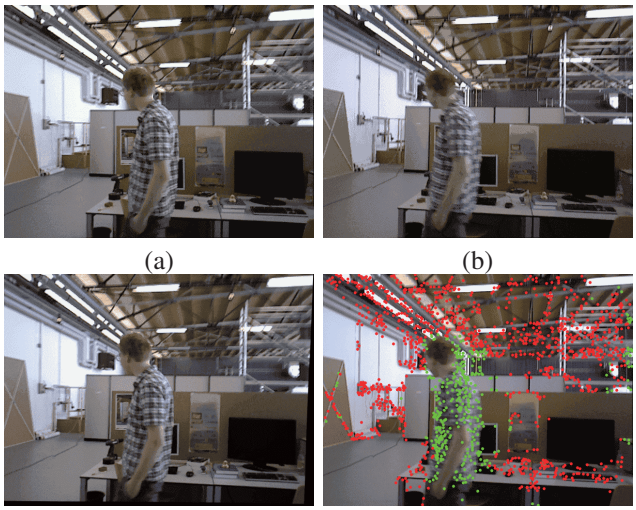


Fig. 2: An experimental result of our method. (a)-(c) represent the last RGB image, the current RGB image and the estimated image. (d) is the result of our method. Red points are static and green points are dynamic.

IV. Experiments

In this section, we demonstrate the feasibility and effectiveness of our method by using the public TUM RGB-D dataset. In the experiments, we integrate our proposed method inside the Visual SLAM system and the main function is to eliminate dynamic feature points which may degrade the performance of Visual SLAM. In this study, we adopt the ORB-SLAM [17] as the Visual SLAM scheme, which is a state-of-the-art feature-based monocular SLAM system. For ORB-SLAM, we only changed the feature extraction part and our algorithm preprocesses the input data of the feature matching module. Our method can process 28 frames per second, which can work in real time.

A. Experiment setup

We verify our method on the base of the public TUM RGB-D dataset [18]. For each video sequence in the dataset, we carry out two kinds of experiments. One is the evaluation of the performance of the proposed method. The other is the comparison of the performance of ORB-SLAM before and after integrating our method. Note that in experiments, we set τ in `Optical flow-based Detection` to 2 empirically.

Fig. 3: Selected experimental results using the TUM Dynamic Objects dataset. For each column, the top one is the original RGB image and the bottom one is the result using proposed method. Colored points represent feature points extracted from the current image. Red points are static and green points are dynamic. As we can see, our method is able to effectively distinguish dynamic points in dynamic scenarios like these in figure.

We used three types of scenarios in TUM Dynamic Objects dataset for our experiments: `desk`, `sitting` and `walking` scenarios. And there are 4 types of camera ego-motions: `halfsphere`, `rpy`, `static` and `xyz`. For brevity, we use the words `fr`, `half`, `w`, `s`, `d`, `v` to denote `freiburg`, `halfsphere`, `walking`, `sitting`, `desk`, `validation` in the names of sequences.

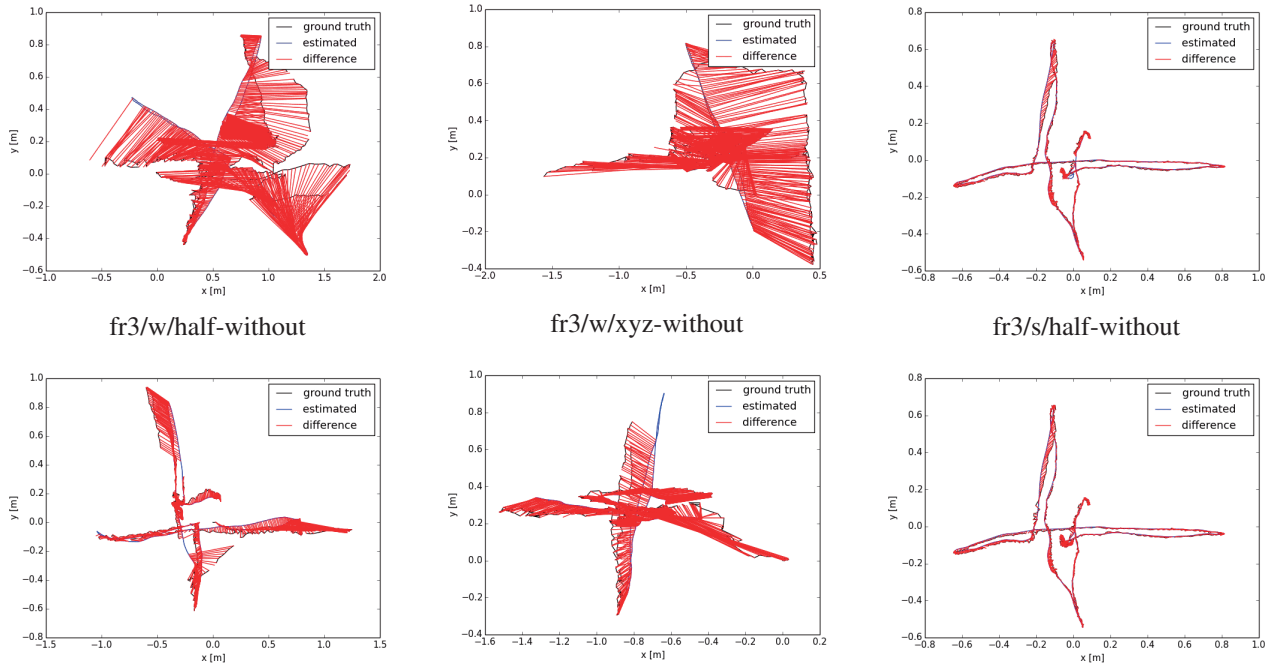
A PC with an Intel i7 CPU and 16GB memory is used in the experiments. The RGB images are processed with 640×480 resolution.

B. Evaluations of proposed method

In this section, we use TUM dataset to demonstrate the feasibility of our proposed method. Our goal is to distinguish dynamic feature points from all the feature points extracted.

Fig. 3 shows some selected experimental results using the TUM Dynamic Objects dataset. For each column, the top one is the original RGB image and the bottom one is the result using the proposed method. Colored points represent feature points extracted from the current image. Red points are static and green points are dynamic. As we can see, our method is able to effectively distinguish dynamic points in dynamic scenarios like these in the figure.

From the result images we can see that, some feature points on dynamic objects are regarded as static ones. This is because when an object is dynamic, not all the parts of it are dynamic. For instance, in fr3/w/static/v sequence, two persons are sitting in a chair, and chatting. In this case, some parts of their body are static, so our algorithm regarded feature points on these parts as static ones. In some images, some static points are taken for dynamic one, we think there are three reasons. First, it is ego-motion estimation. Ego-motion is based on the perspective transformation matrix. In highly dynamic scenarios, dynamic feature points usually



fr3/w/half-without fr3/w/xyz-without fr3/s/half-without

fr3/w/half-with fr3/w/xyz-with fr3/s/half-with

Fig. 4: Plots of ATE for the sequences fr3/w/half, fr3/w/xyz/, fr3/s/half. The words `with` and `without` represent the experiments performed with and without our method. This figure is best viewed in color.

bring noise into the computation of the transformation matrix which will degrade the performance of our method. The second reason is the noise arose from camera motion. Some images in selected sequences are ambiguous due to the camera motion. This ambiguity may have a negative effect on feature extraction or ego-motion estimation. The third reason is thresholding. We use a threshold to determine whether a point is dynamic, and the threshold is set to 2 in our experiments. However, for different sequences optimal threshold values may be different. We will try to find a way to update the threshold value based on the sequence conditions.

C. Evaluation of Visual SLAM

In this part, we evaluate the performance of Visual SLAM after the integration with our method. Both qualitative and quantitative results are given to demonstrate the feasibility of our method. We use the metrics Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) for the quantitative evaluation. The metric ATE measures the global consistency and RPE measures the odometry drift.

1) Qualitative Results: Fig. 4 consists of selected ATE plots which shows the qualitative results of ORB-SLAM after integration with our method. As we can see the ground truth is represented as the black line, and estimated trajectory as the blue line, differences as the red lines, respectively. In Fig. 4, We use `-with` and `-without` to represent that the experiments are performed with and without our method.

Both the sequence fr3/w/half and fr3/w/xyz belong to highly dynamic scenarios. ORB-SLAM cannot work in such cases. After using our method, the performance improved very significantly. The sequence fr3/s/half belongs to a low-dynamic scenario. ORB-SLAM performs very well in such a condition, while we can also see the improvement with our method. For instance, at the top of the trajectory, the difference between ground truth and the estimated trajectory is reduced after using our method. These experimental results show that our method can deal with the proposed problem in both high-dynamic and low-dynamic scenarios.

2) Quantitative Results: Table *I*-Table **??** demonstrate the quantitative results of our experiments. The first column of the tables shows the sequence names. The term `Without Our Approach` represents the original ORB-SLAM algorithm. The term `With Our Approach` represents the ORB-SLAM algorithm with our method integrated. We use the values of RMSE, Mean Error, Median Error and the Standard Deviation (S.D.) in this paper to facilitate our further comparisons. The improvement values in the tables are calculated using $\zeta = (1\text{-}\beta/\alpha)\times 100\%$, where ζ denotes the improvement value; α denotes the value without our method; and β denotes the value with our method. Also, we highlight the RMSE and S.D. values which can reflect the stability of the system.

Table *I* shows the global consistency performance. As we can see, our method brings significant improvements for all the sequences in terms of RMSE and S.D.. For high-dynamic scenarios, the improvements are more obvious and the highest improvement for RMSE is 95.86%. These experimental results demonstrated that our method can deal with the high-dynamic scenarios very effectively. And for the low-dynamic scenarios, our method provides improvements

TABLE I: ATE in meters for the experiments without and with our proposed method. Low dynamic sequences are denoted with a superscript star. Others are high-dynamic sequences. Our method effectively improves the ORB-SLAM performance in all scenarios in terms of ATE.

Sequences	Without Our Approach				With Our Approach				Improvements			
	RMSE	Mean	Median	S.D.	RMSE	Mean	Median	S.D.	RMSE	Mean	Median	S.D.
fr3/w/half	**0.4579**	0.3987	0.3774	**0.2252**	**0.1612**	0.1091	0.0637	**0.1187**	**64.80%**	72.64%	83.12%	**52.71%**
fr3/w/rpy	**0.9046**	0.7685	0.7092	**0.4772**	**0.1533**	0.1048	0.0635	**0.1119**	**83.05%**	86.36%	91.05%	**76.55%**
fr3/w/xyz	**0.4808**	0.4367	0.4276	**0.2011**	**0.1899**	0.1537	0.1148	**0.1115**	**60.50%**	64.80%	73.15%	**44.55%**
fr3/w/half/v	**0.5591**	0.4567	0.2934	**0.3226**	**0.0671**	0.0435	0.0283	**0.0506**	**88.00%**	90.48%	90.35%	**84.31%**
fr3/w/rpy/v	**0.5799**	0.3534	0.0556	**0.4599**	**0.0299**	0.0240	0.0186	**0.0178**	**95.86%**	93.21%	66.55%	**96.13%**
fr3/w/xyz/v	**1.4212**	1.2811	1.1664	**0.6153**	**0.1415**	0.0561	0.0305	**0.1299**	**90.04%**	95.62%	97.39%	**78.89%**
fr3/s/half*	**0.0198**	0.0158	0.0135	**0.0120**	**0.0179**	0.0147	0.0131	**0.0102**	**9.60%**	6.96%	2.96%	**15.00%**
fr3/s/xyz*	**0.0097**	0.0088	0.0083	**0.0042**	**0.0092**	0.0081	0.0075	**0.0043**	**5.15%**	7.95%	9.64%	**-2.38%**
fr2/d/person*	**0.0090**	0.0083	0.0082	**0.0036**	**0.0067**	0.0061	0.0057	**0.0029**	**25.56%**	26.51%	30.49%	**19.44%**

from 5.15% to 25.56%, which is less than that in the high-dynamic scenarios. The reason may be that in low-dynamic scenarios, the relatively less dynamic feature points can be easily distinguished, so the original ORB-SLAM can perform very well in such a situation.

V. Conclusion

In this paper, we propose a novel method to distinguish and eliminate dynamic points in one image as the only input, which is an RGB image and our method can work in real time. The proposed method can be divided into two modules: ego-motion estimation and optical flow-based detection. After integration with our method, the performance of the Visual SLAM in dynamic scenarios is significantly improved. We conducted our experiments on the TUM dataset. Qualitative and quantitative evaluations demonstrated that our method can deal with both high-dynamic and low-dynamic scenarios. However, our method still presents some limitations. For instance, the threshold that is used to distinguish dynamic points is set to a fixed value, which may not be an optimal value for some sequences. Also, when dynamic objects occupy much space of the image, the number of feature correspondences for camera pose estimation will be reduced, which may lead to the track-lost. To overcome these limitations, we would like to enhance our method with learning capability in the future. The threshold will be updated online for different motion modes. Also, when there exist many dynamic objects, the system will choose the static area to extract feature points.

References

[1] Thomas Moore and Daniel Stouch. A generalized extended kalman filter implementation for the robot operating system. In *Intelligent Autonomous Systems 13*, pages 335–348. Springer, 2016.

[2] Yuxiang Sun, Ming Liu, and Max Q-H Meng. Improving rgb-d slam in dynamic environments: A motion removal approach. *Robotics and Autonomous Systems*, 89:110–122, 2017.

[3] Shile Li and Dongheui Lee. Rgb-d slam in dynamic environments using static point weighting. *IEEE Robotics and Automation Letters*, 2(4):2263–2270, 2017.

[4] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[5] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[6] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[7] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[8] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[9] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[10] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct ekf-based approach. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 298–304. IEEE, 2015.

[11] Vladyslav Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers. Direct visual-inertial odometry with stereo cameras. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1885–1892. IEEE, 2016.

[12] Deok-Hwa Kim, Seung-Beom Han, and Jong-Hwan Kim. Visual odometry algorithm using an rgb-d sensor and imu in a highly dynamic environment. In *Proc. Int. Conf. Robot. Intell. Technol. Appl.*, pages 11–26, 2015.

[13] Youbing Wang and Shoudong Huang. Towards dense moving object segmentation based robust dense rgb-d slam in dynamic scenarios. In *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on*, pages 1841–1846. IEEE, 2014.

[14] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustmentâĂŤa modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999.

[15] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004.

[16] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004.

[17] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[18] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 573–580. IEEE, 2012.