Multimedia Systems
https://doi.org/10.1007/s00530-020-00662-9

SPECIAL ISSUE PAPER

Fast stereo visual odometry based on LK optical flow and ORB-SLAM2

Chuanye Tang[1] · **Xinwen Zhao**[1] · **Jianfeng Chen**[1] · **Long Chen**[1] · **Yazhou Zhou**[1]

Abstract
A stereo visual odometry algorithm based on the fusion of optical flow tracking and feature matching called LK-ORB-SLAM2 was proposed. In LK-ORB-SLAM2, the operation of optical flow tracking is introduced to adjust the intensive and time-consuming operation of feature matching. This requires solving a key issue: how to solve the problem of losing feature points during optical flow tracking. For this reason, an adaptive matching-frame insertion scheme is proposed to stop optical flow tracking in time and inserts matching-frames and detect new feature points at the right time to keep LK-ORB-SLAM2 running. The experiment on the KITTI and EuRoC data set showed that LK-ORB-SLAM2 reduced the average processing time per frame of ORB-SLAM2 by about 70%, with the change of less than 2% in its accuracy.

Keywords Visual odometry · Stereo vision · LK optical flow · ORB-SLAM2 · Motion estimation

1 Introduction

Motion estimation is the basis of vehicle intelligence. Common positioning methods have some shortcomings, for example, the system error of inertial measurement unit (IMU) diverges rapidly, while global positioning system (GPS) is often unreliable under tunnels, overpasses and dense trees. Visual odometry (VO) with the camera provides another idea for estimating vehicle motion. Cameras are low-cost and easy to integrate into vehicles [1], so VO is popular currently [2].

VO recovers the camera motion according to the sequence image of the surrounding environment. Depending on the type of sensor, VO can be divided into three types: Monocular VO using only one camera; stereo VO using multiple cameras, of which binocular VO is the most; RGB-D VO using RGB-D cameras [3]. In addition, VO can also be divided into feature-based VO and direct VO according to its workflow [4]. Feature-based VO obtains the camera motion based on feature correspondences acquired by feature matching, while direct VO performs motion estimation by minimizing the photometric error. In [5], an open-source feature-based Simultaneous Localization and Mapping (SLAM) system called ORB-SLAM2 was proposed. According to [6], feature-based methods are more accurate than direct methods, but the processing speed of feature-based methods are not fast enough (e.g. when processing 1000 feature points per frame, it is common to process only ten frames per second). The main reason is that the feature matching process is time-consuming [7].

We proposed a stereo Visual Odometry algorithm called Improved ORB-SLAM2 based on LK Optical flow (LK-ORB-SLAM2) that combines feature matching and tracking, which is faster than feature-based visual odometry while maintaining the accuracy. LK-ORB-SLAM2 performs optical flow tracking in tracking-frames and it can adaptively choose when to stop tracking feature points and insert matching-frames, and re-extracts feature points in the matching-frame to maintain a sufficient number of feature points to ensure the operation of the system. LK-ORB-SLAM2 presents two main contributions:

✉ Xinwen Zhao
2211704010@stmail.ujs.edu.cn

Chuanye Tang
tangchuanye@ujs.edu.cn

Jianfeng Chen
ccjf@ujs.edu.cn

Long Chen
chenlong@ujs.edu.cn

Yazhou Zhou
2221804098@stmail.ujs.edu.cn

[1] Automotive Engineering Research Institute, Jiangsu University, Zhenjiang 212013, China

Published online: 11 June 2020

1. We proposed a novel visual odometry method with a complete workflow based on sparse optical flow (Sect. 3) and an adaptive matching-frame insertion scheme (Sect. 4.4) that stops optical flow tracking in time and inserts matching-frames at the right time to keep the VO system running.
2. The efficiency of LK-ORB-SLAM2 is higher than that of ORB-SLAM2, and the processing time of each frame is about 70% less than that of ORB-SLAM2.

The remainder of this paper is organized as follows. In Sect. 2, the related work of VO is introduced. Section 3 gives a basic description of LK-ORB-SLAM2, while Sect. 4 focus on the details of the proposed visual odometry algorithm. Section 5 mainly shows the experimental results of LK-ORB-SLAM2 on the KITTI [8] data set and the EuRoC [9] data set, Sect. 6 summarizes the paper.

2 Related work

Visual odometry (VO) is a popular research field with a large number of literatures. We will review some of the most relevant work with LK-ORB-SLAM2. Both VO and V-SLAM can estimate camera motion, so this article does not distinguish between the concepts of VO and V-SLAM.

2.1 Feature-based VO

In [10], the author created the term VO and proposed the general workflow of VO. Many researchers proposed SLAM algorithms such as [11–14], which are implemented by Extended Kalman Filter (EKF). The use of EKF results in insufficient real-time performance. In response to this problem, in [15], the author proposed a creative solution: separate localization and mapping, which effectively enhanced the speed of the SLAM algorithm.

Feature point has a great influence on VOs. The accuracy of Scale Invariant Feature Transform (SIFT) feature is extremely high [16]. However, SIFT is not fast enough (when processing 1000 feature points per frame, it is common 5000 ms per frame). Oriented FAST and Rotated BRIEF (ORB) were proposed in [17], which turned out to be 100 times faster than SIFT.

Based on [15] and [16], ORB-SLAM2 was presented in [5], ORB-SLAM2 is one of the state-of-art SLAM systems, it employs three threads, all of which use ORB feature.

2.2 Direct VO

Let's first introduce Semi-direct VO, it refers to VO that combines feature tracking and matching. According to [7], the feature matching process in feature-based VO is time-consuming. To accelerate feature-based VOs, in [18], the author proposed a monocular VO algorithm based on the fusion of optical flow tracking and feature matching. In [19], a stereo VO based on the optical flow tracking and trifocal tensor constraint [20] was proposed and turned out to be 6.2 times faster than the VO in [21]. In [22], the author formally created the term Semi-direct VO.

Direct VO refers to VO that estimates camera motion by minimizing photometric errors. Stereo Direct Sparse Odometry (Stereo DSO) [23] is a typical kind of direct VO, which is fast and with high precision.

2.3 VO based on deep learning

Nowadays, many researchers are exploring ways to integrate deep learning into VO. The first achievement in this field is PoseNet [24], which used a Convolutional Neural Network (CNN) to estimate camera motion. In [25], the author proposed Flowdometry which do not need semi-supervised pretraining. In [26], the author presented a novel method for unsupervised learning of depth and estimate the camera motion.

3 System overview

This section first introduces the workflow of ORB-SLAM2, it contains four steps: detecting feature points, stereo matching, feature matching, and motion estimation. When ORB-SLAM2 receives a frame, it first extracts feature points. After feature detection is the stereo matching process. The scene points created in the previous frame are then projected into the current frame, and matched with the current frame to get several feature correspondences. At last, the motion can be calculated using a common 3D-2D algorithm.

Different from ORB-SLAM2, as illustrated in Fig. 1, when LK-ORB-SLAM2 loads a frame, it will determine whether the frame is the first frame. If it is, LK-ORB-SLAM2 will perform initialization: detect feature points in the frame, then some scene points will be acquired through stereo matching (the method of stereo matching will be described in Sect. 4.2), and the first frame will be set as a matching-frame. After initialization, using optical flow to track feature points to acquire feature correspondences. Feature points keep losing during the camera motion, then the matching-frame insert scheme proposed by us will insert matching-frames at the right time. (The matching-frame insert scheme will be described in Sect. 4.4). LK-ORB-SLAM2 will detect feature points in the matching-frame and perform feature matching to acquire feature correspondences. When acquired enough feature correspondences, the

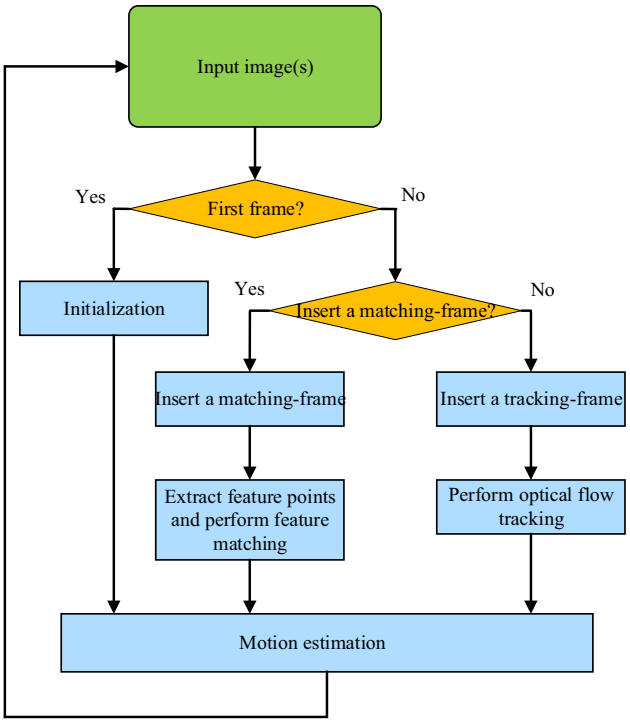


Fig. 1 Architecture of LK-ORB-SLAM2

motion can be calculated using a common PnP algorithm [3] (the principle of the PnP algorithm is in this section of Reference [3]: 3-D-to-2-D: Motion from 3-D Structure and Image Feature Correspondences).

4 Detailed description of the system

In this part, we will introduce details of each module of LK-ORB-SLAM2, including the improved feature point extraction algorithm, stereo matching algorithm, optical flow tracking algorithm and the matching-frame insert scheme.

4.1 Feature extraction algorithm

The performance of the feature point has an exponential impact on visual odometry. According to [17], ORB is much more efficient than SIFT and speeded up robust features (SURF) [27]. ORB builds on the features from accelerated segment test (FAST) [28] keypoint detector and binary robust independent elementary features (BRIEF) [29] descriptor.

The disadvantage of ORB is that the distribution of the feature points is relatively centralized, which is not conducive to finding these points in subsequent frames. ORB-SLAM2 divided the image into several rectangular blocks and extracted key points separately from each rectangular

Fig. 2 Feature points detected by LK-ORB-SLAM2 and traditional ORB algorithm

block to solve this problem. LK-ORB-SLAM2 used this method to optimize ORB.

The color circles in Fig. 2 represent feature points. The upper half of Fig. 2 shows the feature points detected by LK-ORB-SLAM2, and the points in the lower half were extracted by the original ORB method. It could be seen that the points in the upper half were more widely and evenly distributed in the image.

4.2 Stereo matching

After feature detection, the stereo matching algorithm will be used to calculate the depth of scene points. Figure 3 illustrates the process of stereo matching, for the blue feature point in the left image, a banded search area is created in the same line in the right image, and the points in this area are matching candidates. For the two matching candidates in the right image,

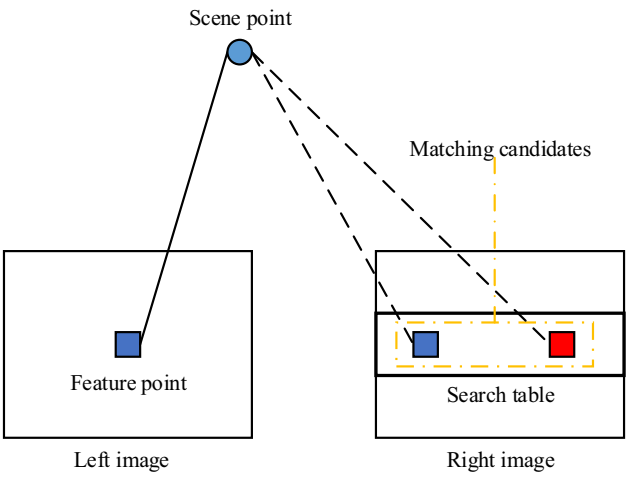


Fig. 3 Stereo matching process. The pixel with the smallest descriptor distance in the search area would be matched with the point in the left image

calculate their BRIEF descriptors respectively and then compare the descriptor distance respectively (for more information on feature descriptors and distance comparisons, refer to the literature [3]). The point with smaller distance is the matching point of the blue point in the left image.

After stereo matching, the depth of the scene point will be calculated by the following formula:

$$z = fb/d \tag{1}$$

where z is the depth of the scene point, f is the focal length of the camera, b denotes the baseline between two cameras and d is the parallax between two feature points.

4.3 Optical flow tracking

This subsection first introduces what is a feature correspondence. A correspondence means a pair of feature points corresponding to the same scene point while distributed over two frames. The feature-based VO obtains feature correspondences through the feature matching process. Feature matching finds feature correspondences between two frames by comparing all feature descriptors in the current frame with all feature descriptors in the previous frame, which is time-consuming. According to [7], feature matching takes about 64% of the entire VO process.

To solve the problem that feature matching takes too much time, LK-ORB-SLAM2 uses optical flow to calculate the movement of feature points in the previous frame so as to find these feature points directly in the current frame, thereby directly getting feature correspondences. Figure 4 shows the process of optical flow tracking, where two feature correspondences are found.

To use optical flow to track features, an assumption needs to be satisfied: the grayscale value of the feature point does not change when the movement is small. The grayscale value of a feature point in the previous frame at t-time is represented as $G(x, y, t)$. Where (x, y) is the localization of the feature point. After time $\mathrm{d}t$, the gray value of the same feature point in the current frame is represented by $G(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t)$. According to the above assumption, there is:

$$G(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t) = G(x, y, t). \tag{2}$$

Performing a Taylor expansion on the right side of Eq. (2) yields:

$$G(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t) = G(x, y, t) + \frac{\partial G}{\partial x}\mathrm{d}x + \frac{\partial G}{\partial y}\mathrm{d}y + \frac{\partial G}{\partial t}\mathrm{d}t + o(\mathrm{d}x, \mathrm{d}y, \mathrm{d}t). \tag{3}$$

According to Eqs. (2) and (3), there is:

$$\frac{\partial G}{\partial x} \cdot \mathrm{d}x + \frac{\partial G}{\partial y} \cdot \mathrm{d}y + \frac{\partial G}{\partial t} \cdot \mathrm{d}t + o(\mathrm{d}x, \mathrm{d}y, \mathrm{d}t) = 0. \tag{4}$$

Dividing both side of Eq. (4) by $\mathrm{d}t$ yields:

$$\frac{\partial G}{\partial x} \cdot \frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial G}{\partial y} \cdot \frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial G}{\partial t} = 0. \tag{5}$$

$\frac{\mathrm{d}x}{\mathrm{d}t}$ and $\frac{\mathrm{d}y}{\mathrm{d}t}$ represent the movement of the feature point in the previous frame, so that the position of the same feature point in the current frame can be found through the movement, then a feature correspondence will be found. After all feature correspondences are obtained, we use the reverse optical flow tracking method to remove wrong feature correspondences. The illustration of reverse optical flow tracking is in Fig. 5, the solid line in the figure represents the process of optical flow tracking, 5 corresponds to 1, and 6 corresponds to 2. Now perform reverse tracking. The dashed line in the figure represents the process of reverse tracking, from 1 to 3. If the distance between 5 and 3 is less than 0.6 pixels, the correspondence between 5 and 1 is correct. On the contrary, the correspondence between 5 and 1 is wrong and should be deleted. Do the same for 6 and 2. Reverse tracking can improve the accuracy of optical flow tracking.

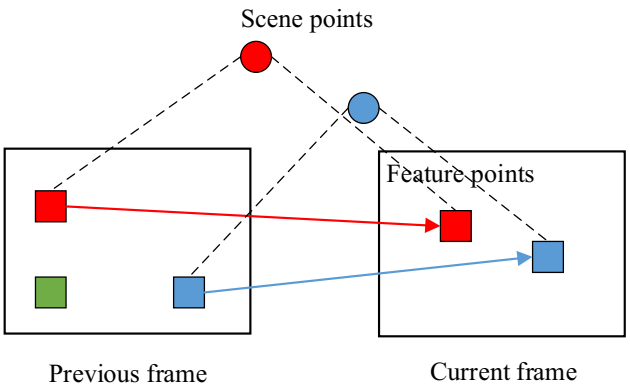


Fig. 4 Illustration of optical flow tracking

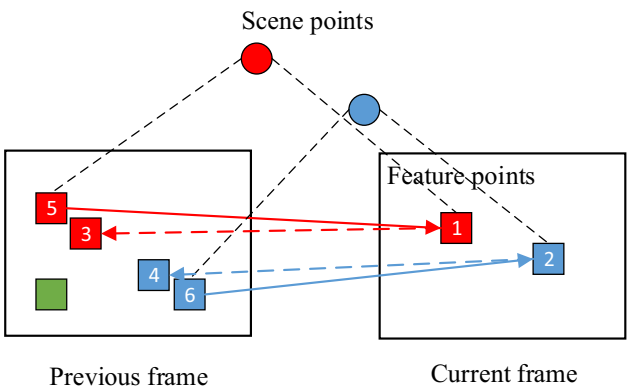


Fig. 5 Illustration of reverse optical flow tracking

Optical flow tracking has a problem: The number of feature points keeps losing during the tracking process. If the number of feature points is too small, the motion estimation of the current frame will fail [3].

To solve the problem, in the monocular VO method of [18], feature points are extracted on a fixed frame spacing (spacing of 20 basic frames). This can cause the problem that sometimes although there are adequate features, VO still performs feature point extraction, which leads to the waste of computing resources; while sometimes the available feature points in the image frame are not enough, which makes VO unable to work. In LK-ORB-SLAM2, a novel mechanism is proposed to adaptively set matching-frames and extract features at the right time.

4.4 Optical flow tracking

Let's first quantify the difficulty of optical flow tracking. The loss rate of feature points is directly proportional to the difficulty of optical flow tracking. This paper uses F to represent frames. Suppose there is an optical flow tracking process that continues for Q frames and $\{F_1, \ldots, F_Q\}$ is used to represent the process. Let $\xi_{F_i}, i = 1, \ldots, Q$ be the number of features in ith frame. We use tracking difficulty coefficient η to quantitatively describe the difficulty of the above tracking process, there is:

$$\eta = \frac{\xi_{F_1} - \xi_{F_Q}}{\xi_{F_1}}. \tag{6}$$

It can be known from the above formula that η is directly proportional to the difficulty of the tracking process.

The proposed adaptive matching-frame insertion scheme performs inserting matching-frames in advance when tracking is difficult, while delays the insertion when tracking is easy. The scheme is implemented as follows:

1. Set the initial frame (F_1) as a matching-frame and perform feature detection;
2. Set the next H frames (i.e. $F_2, \ldots, F_{H+1}$) as tracking-frames and perform optical flow tracking, where H is a preset value used to start the matching-frame insert scheme;
3. Include three sub-steps: 1) to set F_{H+2} as a matching-frame and perform feature detection (here we call the set of two adjacent matching-frames and all the tracking-frames between them as an optical flow tracking process), 2) to calculate η for the first optical flow tracking process ($\{F_1, \ldots, F_{H+2}\}$) according to Eq. (5), and 3) to set the next n frames as tracking-frames and perform feature tracking, where n represents adaptive tracking times and is given by:

$$n = (1 - \eta) \times m, \tag{7}$$

where m represents the average continuous tracking times without adding new feature points;

4. Repeat step 3 until the end.

The adaptive matching-frame insertion scheme is shown in Fig. 6.

According to our test, if LK optical flow is used to track feature points in the sequence images of the KITTI dataset, it is found that on average 20 frames can be continuously tracked without adding feature points. After 20 frames, because the feature points are too few, the tracking cannot continue. So we recommend that when using the KITTI dataset, m takes a number around 20. Of course, in different environments, m can take different values according to the actual situation. We recommend that the $H=3$ in KITTI data set. If $H>3$, then there are cases where the tracking fails before $_{F_{H+2}}$. So $H=3$. Of course, in different environments, H can take different values according to the actual situation.

The method of using LK optical flow and adaptively inserting matching-frames to quickly obtain feature correspondences is introduced above. The motion can be calculated using a common 3D-2D method after getting enough feature correspondences. Suppose there are n feature correspondences achieved by optical flow. Then the relationship between the scene points and the feature points is expressed as follows:

$$s_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = K \left(R \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + t \right), \quad i = 1, 2, \ldots, n, \tag{8}$$

where s_i is the depth of the scene point, $(u_i, v_i, 1)^{\mathrm{T}}$ is the feature point, $(X_i, Y_i, Z_i)^{\mathrm{T}}$ is the scene point, K is the camera intrinsic and R, t are the camera extrinsics.

Minimizing the reprojection error to estimate R and t:

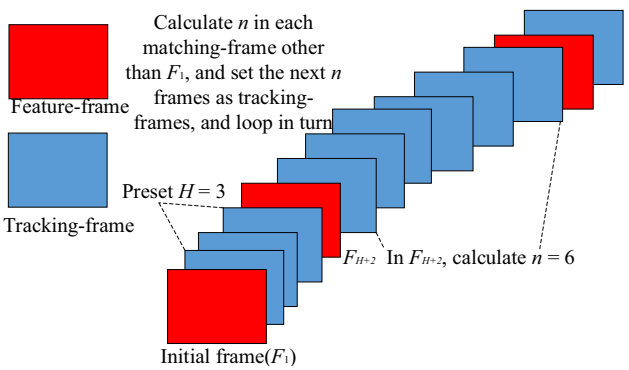


Fig. 6 Illustration of adaptive matching-frame insertion scheme

$$\{R,t\} = \arg\min_{R,t} \sum_{i=1}^{n} \rho\left(\left\|u_i - \frac{1}{s_i}K(RP_i + t)\right\|_{\Sigma}^{2}\right), \tag{9}$$

where u_i is a feature point, P_i is a scene point, ρ is the Huber cost function and $\sum$ is the covariance matrix.

5 Experimental results

To evaluate LK-ORB-SLAM2, we performed experiments on the famous KITTI and EuRoC data set. An Intel Core i5-2520 M CPU @ 2.50 GHz × 4 computer was used in all experiments and the operating system was Ubuntu 16.04.

5.1 KITTI data set

We compared our algorithm with the state-of-the-art method ORB-SLAM2. The main parameter that affects the performance of VO is the number of feature points extracted in the frame. The feature points extracted by the two algorithms in this experiment are all 1000, the number of layers of an image pyramid is 8, and the scaling factor is 1.2. $m = 20$ and $H = 3$.

Root mean square error of Absolute Trajectory Error (ATE) [30] was adopted to evaluate the precision of VO and the average processing time was utilized to evaluate the speed of VO. ATE is a commonly used measure for evaluating visual SLAM [31] systems. Assuming that the camera moved k frames in total, we use $Q_{1:k}$ for ground truth trajectory and $P_{1:k}$ for estimated trajectory. The ATE at time step i can be computed as:

$$ATE_i = Q_i^{-1} S P_i \tag{10}$$

where S is the rigid-body transformation between the estimated motion and the groundtruth at time step i.

Then the ATE of the whole k frames is:

$$\mathrm{ATE} = \left(\frac{1}{k}\sum_{i=1}^{k}\left\|\mathrm{trans}(\mathrm{ATE}_i)\right\|^2\right)^{1/2}, \tag{12}$$

where $\mathrm{trans}(\mathrm{ATE}_i)$ is the translation error of ATE_i.

KITTI is an open-source data set for evaluating the performance of computer vision algorithms. The data set consists of 22 stereo sequences. The resolution of the image is 1240 × 376.

The average processing time of these two methods for sequence 00–10 are shown in Table 1. Figures 7,8,9,10 shows the trajectories of sequence 01,03,08,10 estimated by LK-ORB-SLAM2 and ORB-SLAM2.

Table 1 Processing time per frame of TWO algorithms (KITTI)

KITTI	Processing time per frame (ms)			
	LK-ORB-SLAM2		ORB-SLAM2	
	Median	Mean	Median	Mean
00	15.25	25.41	105.67	107.45
01	61.93	47.62	84.55	80.74
02	68.91	54.04	90.87	147.92
03	15.55	27.21	101.4	105.45
04	18.36	33.26	99.16	105.85
05	15.58	28.44	110.69	112.50
06	16.89	34.91	95.36	98.75
07	15.42	27.09	110.86	114.62
08	14.51	33.47	112.24	120.00
09	16.11	31.38	104.86	105.38
10	15.23	27.66	103.56	105.76
Average	24.89	33.68	101.75	109.49

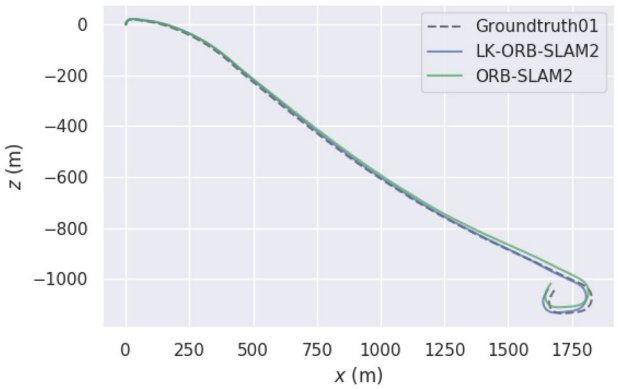


Fig. 7 The trajectory of KITTI sequence 01 achieved by LK-ORB-SLAM2 and ORB-SLAM2

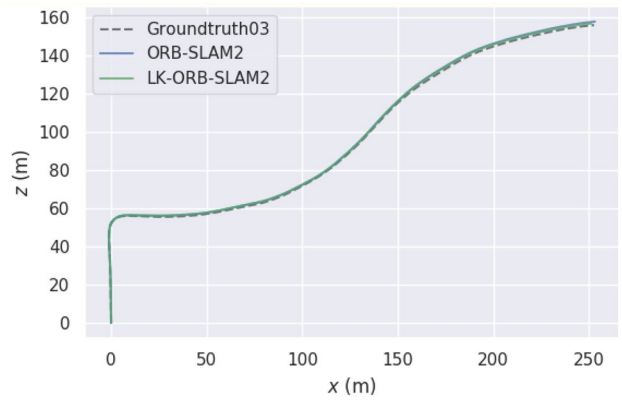


Fig. 8 The trajectory of KITTI sequence 03 achieved by LK-ORB-SLAM2 and ORB-SLAM2

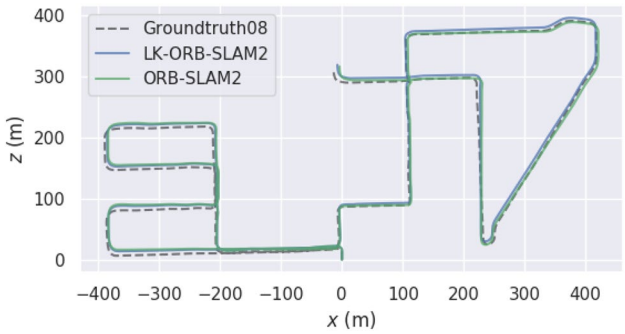


Fig. 9 The trajectory of KITTI sequence 08 achieved by LK-ORB-SLAM2 and ORB-SLAM2

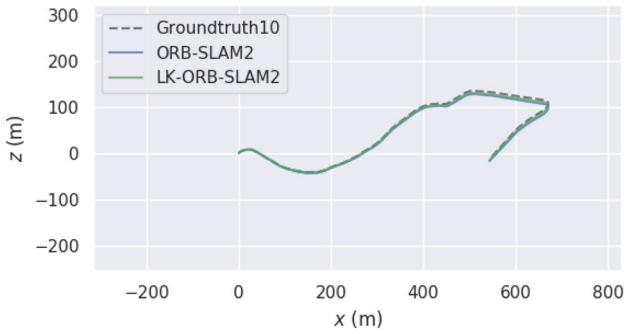


Fig. 10 The trajectory of KITTI sequence 10 achieved by LK-ORB-SLAM2 and ORB-SLAM2

Because the efficiency of optical flow tracking is much higher than the feature matching process, LK-ORB-SLAM2 is much faster than ORB-SLAM2 in all sequences. According to Table 1, the mean processing time per frame of LK-ORB-SLAM2 is about 75% less than that of ORB-SLAM2 except sequences 01 and 02. The mean processing time per frame of LK-ORB-SLAM2 is about 41% less than that of ORB-SLAM2 in sequences 01, and 64% in sequence 02. The reason for this phenomenon is that the vehicle movements in sequences 1 and 2 are more intense than that in other sequences, so the tracking difficulty coefficient η is large, and the adaptive tracking times n is small, which leads to the need to frequently re-extract feature points and perform feature matching. In all sequences, the mean processing time per frame of LK-ORB-SLAM2 is about 69.2% less than that of ORB-SLAM2.

The RMSE of ATE are shown in Table 2. The smaller the ATE, the higher the accuracy of the algorithm. It can be seen from Table 2 that the ATEs of the two algorithms are close in all sequences. From Figures 7,8,9,10 we can also find that the trajectories recorded by LK-ORB-SLAM2 and ORB-SLAM2 are very close. In all sequences, according to average ATE, LK-ORB-SLAM2 is 1.4% more accurate than ORB-SLAM2, therefore it came to the conclusion that LK-ORB-SLAM2 is as precise as ORB-SLAM2.

From Tables 1, 2 and Figures 7,8,9,10, LK-ORB-SLAM2 is faster than ORB-SLAM2 while maintaining the accuracy.

Table 2 ATE of TWO algorithms (KITTI)

KITTI	Absolute trajectory error RMSE in meters	
	LK-ORB-SLAM2	ORB-SLAM2
0	0.57	0.72
1	8.83	9.58
2	5.21	4.54
3	0.35	0.58
4	0.04	0.02
5	2.09	1.91
6	0.67	0.66
7	0.97	0.79
8	3.4	3.23
9	0.27	0.45
10	1.18	1.35
Average	2.14	2.17

Table 3 Processing time per frame of TWO algorithms (EuRoC)

EuRoC	Processing time per frame(ms)			
	LK-ORB-SLAM2		ORB-SLAM2	
	Median	Mean	Median	Mean
V1_01_easy	15.38	35.41	85.07	88.34
V1_02_medium	16.93	40.62	91.99	91.56
V1_03_difficult	18.41	39.05	89.03	91.04
V2_01_easy	15.55	27.21	80.37	82
V2_02_medium	18.31	33.26	103.07	112.26
V2_03_diffucult	14.58	28.4	74.96	77.97
MH_01_easy	26.89	34.91	110.85	127.48
MH_02_easy	18.41	27.91	100.3	119.92
MH_03_medium	14.21	33.47	110.86	118.32
MH_04_difficult	16.14	31.38	95.7	97.93
MH_05_difficult	19.34	37.29	105.94	113.73
Average	17.65	33.54	95.29	101.87

5.2 EuRoC data set

The EuRoC data set contains 11 stereo sequences captured from two rooms and an industrial environment by a micro aerial vehicle. These sequences are divided into easy, medium, difficult according to different speed and environment texture.

We compared LK-ORB-SLAM2 with ORB-SLAM2 on the EuRoC data set. To ensure fairness, LK-ORB-SLAM2 and ORB-SLAM2 used exactly the same configuration file

Table 4 ATE of TWO algorithms(EuRoC)

EuRoC	Absolute trajectory error RMSE in meters	
	LK-ORB-SLAM2	ORB-SLAM2
V1_01_easy	0.062	0.06
V1_02_medium	0.058	0.063
V1_03_difficult	0.069	0.059
V2_01_easy	0.068	0.059
V2_02_medium	0.065	0.059
V2_03_diffucult	0.156	0.173
MH_01_easy	0.046	0.035
MH_02_easy	0.041	0.046
MH_03_medium	0.063	0.058
MH_04_difficult	0.173	0.093
MH_05_difficult	0.057	0.06
Average	0.078	0.07

(1200 feature points, the number of layers of an image pyramid is 8, and the scaling factor is 1.2.). The average calculation time per frame for all sequences are listed in Table 3. LK-ORB-SLAM2 took the least time in all sequences. On average, the mean processing time per frame of LK-ORB-SLAM2 is about 67% less than that of ORB-SLAM2. The RMSE of ATE for all sequences of the EuRoC dataset are listed in Table 4, In all 11 sequences, ORB-SLAM2 achieved the highest accuracy in 7 sequences and LK-ORB-SLAM2 achieved the highest accuracy in 4 sequences. On average, the RMSE of LK-ORB-SLAM2 is 1.02% higher than that of ORB-SLAM2 so the accuracy of LK-ORB-SLAM2, ORB-SLAM2 is at the same level.

From Tables 3 and 4, LK-ORB-SLAM2 also is faster than the ORB-SLAM2 while maintaining accuracy.

6 Conclusion

The proposed stereo visual odometry algorithm LK-ORB-SLAM2 is highly efficient. This is verified by the experimental results on the KITTI and EuRoC data set. The experimental results showed that LK-ORB-SLAM2 reduces the average processing time per frame of ORB-SLAM2 by about 70%, with almost no change of accuracy.

Funding This work was supported by the National Natural Science Foundation of China under Grant #61603158, the Senior Talent Fund Project of Jiangsu University under Grant #16JDG067 and the Six Talent Peaks Project in Jiangsu Province under Grant #2016-JXQC-007.

References

1. Wu, T., Zhao, J.Y., Zhang, Z.L., Lu, Z.Y., Chang, Z.J.: On current status and development tendency of vehicle visual odometer. Electron Optics Control **24**(10), 69–74 (2017)
2. Lu, W., Xiang, Z., Liu, J.: High-performance visual odometry with two-stage local binocular BA and GPU. In: Presented at the 2013 IEEE Intelligent Vehicles Symposium (IV), 23–26 June 2013
3. Scaramuzza, D., Fraundorfer, F.: Visual Odometry [Tutorial] Part I: The first 30 Years and Fundamentals. IEEE Robot. Automat. Mag. **18**, 80–92 (2011)
4. Nicola, K., David, D., Sebastian, H., Sven, B.: Feature-based visual odometry prior for real-time semi-dense stereo SLAM. Robot. Auton. Syst. **109**, 38–58 (2018)
5. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D Cameras. IEEE Trans. Rob. **33**(5), 1255–1262 (2017)
6. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: "ORB-SLAM: a Versatile and Accurate Monocular SLAM System. IEEE Trans. Rob. **31**(5), 1147–1163 (2015)
7. Gao, X., Zhang, T., Liu, Y., Yan, Q.: 14 Lectures on Visual SLAM: From Theory to Practice. Publishing House of Electronics Industry (2017)
8. Geiger A.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Presented at the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 16–21 June 2012
9. Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., Siegwart, R.: The EuRoC micro aerial vehicle datasets. Int. J. Robot. Res. **35**(10), 1157–1163 (2016)
10. Nister, D., Naroditsky, O., Bergen, J.R.: Visual odometry. In: Presented at the 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1 Jan 2004
11. Davison, A.J.: SLAM with a single camera. In: Presented at the Proceedings of Workshop on Concurrent Mapping an Localization for Autonomous Mobile Robots in Conjunction with ICRA (2002)
12. Davison, A.J.: Real-time simultaneous localization and mapping with a single camera. In: presented at the Proceedings of the Ninth IEEE International Conference on Computer Vision (2003)
13. Davison, A.J., Reid, I.D., Molton, N.D.: Mono-SLAM: real-time single camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)
14. Civera, J., Davison, A.J., Montiel, J.M.M.: Inversedepth parametrization for monocular SLAM. IEEE Trans. Rob. **24**(5), 932–945 (2008)
15. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: IEEE and ACM International Symposium on Mixed and Augmented Reality (2007)
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
17. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: Presented at the International Conference on Computer Vision, 1 Nov. 2012
18. Zheng, C., Xiang, Z., Liu, J.: Monocular vision odometry based on the fusion of optical flow and feature points matching. J. Zhejiang Univ. (Eng. Sci.) **48**(2), 279–284 (2014)
19. Cheng, C., Hao, X., Zhang, Z., Zhao, M.: Stereo visual odometry based on Kalman fusion of optical flow tracking and trifocal tensor constraint. J. Chin. Inertial Technol. **24**(2), 473–479 (2016)
20. Chen, Y.J., Yang, G.L., Jiang, Y.X., Liu, X.Y.: Monocular visual odometry based on trifocal tensor constraint. J Phys Conf Ser **976**, 012002 (2018)
21. Kitt, B., Geiger, A., Lategahn, H.: Visual odometry based on stereo image sequences with RANSAC-based outlier rejection

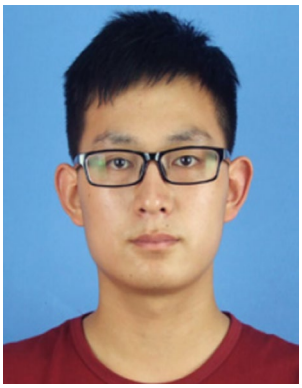
scheme. In: Presented at IEEE Intelligent Vehicles Symposium, pp. 486–492 (2010)
22. Forster, C., Zhang, Z., Gassner, M., Werlberger, M., Scaramuzza, D.: Svo: semidirect visual odometry for monocular and multicamera systems. IEEE Trans. Robot. **33**(2), 249–265 (2017)
23. Wang, R., Schwörer, M., Cremers, D.: Stereo dso: large-scale direct sparse Visual Odometry with stereo cameras. In: Presented at the 2017 IEEE International Conference on Computer Vision (ICCV) (2017)
24. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a convolutional net-work for real-time 6-DOF camera relocalization. In: IEEE International Conference on Computer Vision (ICCV), pp. 2938–2946 (2015)
25. Muller, P., Savakis, A.: Flowdometry: an optical flow and deep learning based approach to visual odometry. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 624-631 (2017)
26. Zhang, J., Su, Q., Liu, P., Xu, C., Chen, Y.: Unsupervised learning of monocular depth and large-ego-motion with multiple loop consistency losses. IEEE Access (2019). https://doi.org/10.1109/ACCESS.2019.2920301
27. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Comput. Vis. Image Underst **110**(3), 346–359 (2008)
28. Rosten, E., Drummond, T.: Machine Learning for High-Speed Corner Detection. Springer, Berlin (2006)
29. Calonder, M, Lepetit, V., Strecha, C., Pascal, F.: BRIEF: binary robust independent elementary features. In: Presented at 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 Sep 2010
30. Jürgen, S., et al.: A benchmark for the evaluation of RGB-D SLAM systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems IEEE (2012)
31. Strasdat, H., Montiel, J.M.M., Davison, A.J.: Visual SLAM: why filter? ☆. Image Vis. Comput. **30**(2), 65–77 (2012)

Chuanye Tang was born in Anhui, People's Republic of China, in 1982. He received the B.S. degree in Chang'an University in 2004, the M.S. degree in Northwestern Polytechnical University in 2007, and the Ph.D. degree in Southeast University in 2016. Currently, he is an associate professor at the School of Automotive and Traffic Engineering, Jiangsu University. His main research interests include inertial navigation, visual navigation and multisensor integrated navigation, as well as the environmental perception of intelligent vehicles.

Xinwen Zhao was born in Yancheng City, Jiangsu Province, China, in 1994. He received the B.S. degree in vehicle engineering from the Nanjing Institute of Technology in 2017. He is currently pursuing the M.S. degree with the Automotive Engineering Research Institute, Jiangsu University, Zhenjiang, China. His main research interests include computer vision, Visual Odometry and vehicle positioning.

Jianfeng Chen received the bachelor's degree in industry automation and the master's degree in power electronics and power drives from Jiangsu University in 2002 and 2005 respectively, and the Ph.D. degree in instrument science and technology from Southeast University in 2015. He is now an associate professor with the Automotive Engineering Research Institute, Jiangsu University, China. His current research interests include vehicle dynamic and control, strapdown inertial navigation system (SINS), and integrated measurement. His professional activities mainly span data fusion algorithms, and the state estimation, SINS algorithm design and evaluation, performance analysis and control optimization of vehicle motion.

Long Chen received the B.Sc. and Ph.D. degrees in mechanical engineering from Jiangsu University, Zhenjiang, China, in 1982 and 2006, respectively. He is currently a Professor with the Automotive Engineering Research Institute, Jiangsu University. His areas of interest include electric vehicles, electric drives, simulation and control of vehicle dynamic performance, vehicle operation, and transport planning.



Yazhou Zhou received his B.S. degree in vehicle engineering from the Nanjing University of Technology in 2018. he is currently pursuing for a master degree in vehicle engineering from Jiangsu University, Zhenjiang, China. His current research direction is the fusion of vision and inertial sensors for intelligent vehicle positioning.