

Neuromorphic Visual Odometry System For Intelligent Vehicle Application With Bio-inspired Vision Sensor

Dekai Zhu¹, Zhongcong Xu¹, Jinhu Dong¹, Canbo Ye¹, Yinbai Hu², Hang Su³, Zhengfa Liu¹, Guang Chen^{1,2}

¹College of Automotive Engineering, Tongji University, China

²Robotics and Embedded Systems, Technische Universität München, Germany

³Department of Electronics, Information, and Bioengineering, Politecnico di Milano, Italy

Abstract—The neuromorphic camera is a brand new vision sensor that has emerged in recent years. In contrast to the conventional frame-based camera, the neuromorphic camera only transmits local pixel-level changes at the time of its occurrence and provides an asynchronous event stream with low latency. It has the advantages of extremely low signal delay, low transmission bandwidth requirements, rich information of edges, high dynamic range etc., which make it a promising sensor in the application of in-vehicle visual odometry system. This paper proposes a neuromorphic in-vehicle visual odometry system using feature tracking algorithm. To the best of our knowledge, this is the first in-vehicle visual odometry system that only uses a neuromorphic camera, and its performance test is carried out on actual driving datasets. In addition, an in-depth analysis of the results of the experiment is provided. The work of this paper verifies the feasibility of in-vehicle visual odometry system using neuromorphic cameras.

I. Introduction

Neuromorphic vision sensors, also known as event cameras or dynamic vision sensors, capture pixel-level changes caused by movement in a scene (called "events"). Unlike conventional cameras, the output of a biological heuristic camera is not an intensity image but an event stream, which is continuous in time. An event can be represented by a tuple $[t, x, y, p]$, where x and y represent the pixel coordinates of the events in the projected scene, t represents the timestamp and p indicates the polarity of the event (When the brightness is changed from dark to light, the polarity is positive; otherwise it is negative). Since the events are only generated when the brightness of a pixel changes, these pixels are mostly the edge of objects. This characteristic is advantageous for object recognition problem in computer vision because it can effectively reduce the storage and computational requirement. Moreover, event cameras can achieve an extremely high output frequency with a negligible latency within tens of microseconds. In addition, event cameras have a very high dynamic range of 130 dB, while conventional cameras can only achieve 60 dB[1].

II. Related Work

Due to the characteristics of low latency, high dynamic range, sparse event stream and effective detection of the

edges, the event cameras seem to be an ideal sensor for in-vehicle visual odometry system. Firstly, low latency ensures that the sensor equipped on the vehicle can obtain real-time information when the vehicle drives extremely fast. Secondly, high dynamic range improves the robustness against the extreme illumination change. Thirdly, the sparse event stream reduces the requirement of data transmission bandwidth.

In spite of the aforementioned advantages, it is still a tremendous challenge to apply the event camera on visual odometry system for vehicle since the output of the neuromorphic vision sensor is event stream, which is absolutely different with the conventional intensity images. Over the past few years, many researchers have attempted to solve this problem in more and more complex scenarios[2]. The complexity of these scenarios can be measured by the following three metrics: the type of motion, the type of scene and the dimensionality.

The type of motion can be divided into rotational motion, planar motion (both are 3-DOF) and free motion in 3D space (6-DOF). Regarding the type of scene, it can be divided into artificial scenes and natural scenes. The artificial scene contains more texture information while the natural scene are much more complex because of larger illumination changes and more dynamic objects. In recent years, Some researchers introduce extra visual sensors such as conventional cameras or RGB-D cameras. In 2016, Rebecq et al.[2] has made a very detailed review about this area, so we optimize their review by adding some latest works on the basis of their review in Table 1.

A relatively complete event based 2D SLAM system was proposed by Weikersdorfer et al. in [5], which is limited to planar motion. In addition, this system requires a parallel plane with artificial black-and-white lines. A year later, Weikersdorfer et al.[7] extended their event based 2D SLAM system to 3D SLAM system with an extra RGB-D camera. The extra sensor provides depth information but it also slows down the SLAM system. In [4], a filter based system was proposed for estimating the pose of the event camera and generating high resolution panorama, but this system is only able to estimate the rotational

TABLE I
The relative works about the event camera based pose tracking and/or mapping.

Reference	2D/3D	Scene type	Event camera only	Depth	Remarks
Cook et al. 2011 [3]	2D	natural	✓	X	rotational motion only
Kim et al. 2014 [4]	2D	natural	✓	X	rotational motion only
Weikersdorfer et al. 2013 [5]	2D	artificial	✓	X	a scene parallel to the plane of motion
Censi et al. 2014 [6]	3D	artificial	X	X	requires depth from extra RGB-D camera
Weikersdorfer et al. 2014 [7]	3D	natural	X	✓	requires depth from extra RGB-D camera
Mueggler et al. 2014 [8]	3D	artificial	✓	X	requires a prior 3D map of lines
Gallego et al. 2016 [9]	3D	natural	X	X	requires a prior 3D map
Kueng et al. 2016 [10]	3D	natural	X	✓	requires intensity images
Kim et al. 2016 [11]	3D	natural	✓	✓	requires intensity images and GPU
Rebecq et al. 2016 [2]	3D	natural	✓	✓	utilizes EMVS [12] to reconstruct 3D sparse point cloud
Gallego et al. 2017 [13]	2D	natural	✓	X	estimates angular velocity with low latency and high precision
Zhu et al. 2017 [14]	3D	natural	X	X	fusion with IMU based on Kalman filter
Mueggler et al. 2018 [15]	3D	natural	X	✓	fusion with IMU based on EVO [2]
Rosinol et al. 2018 [16]	3D	natural	X	✓	fusion with grayscale camera and IMU
Our work	3D	natural	✓	✓	utilizes the feature tracking proposed in [14] for both pose and depth estimation

motion without transfer and depth. In [10] Kueng et al. proposed a visual odometry system for parallel tracking and mapping. The system estimates 6-DOF motion of camera in natural environment by tracking sparse features in event streams. This is the first event based visual odometry system that uses sparse feature points, but it still requires intensity images from conventional camera to detect feature points at first. Compared with the visual odometry system completely based on conventional camera, this system avoids a tremendous computational burden of feature tracking. In [11] Kim et al. proposed a system based on three probabilistic filters to achieve pose tracking and reconstruction of depth and intensity images in natural scenes. This system is so computational expensive that it requires GPU, that makes it inappropriate for the platforms with limited computational power. EVO presented in [2] is the first 3D SLAM system only relies on event camera. It achieves parallel 6-DOF estimation and 3D map construction with a combination of the depth estimation algorithm from [12] and a key-frame based visual odometry system. EVO matches the frames of short-term accumulation of events with the already built map to estimate the pose of camera. This matching procedure is computational efficient, so EVO can work in real time on CPU. Compared with all these works, our visual odometry system is most similar to EVO, because it utilizes only event camera to achieve parallel pose tracking and mapping, while it combines a novel feature tracking algorithm proposed by Zhu et al.[14].

In recent years many researches has focused on the

fusion of multiple sensors to achieve higher accuracy and robustness. In [14] and [15], the authors combined the event camera with IMU. In [16], Vidal et al. even combined three kind of sensors: the event camera, IMU and the grayscale camera, to provide a mature engineering application.

III. Event based visual odometry using feature tracking

The main contribution of this paper is to propose a event based in-vehicle visual odometry system that utilizing the feature tracking algorithm proposed by Zhu et al. in [14]. The framework of our visual odometry system is shown in Fig.1.

Our visual odometry system refers to the framework of some popular visual odometry system[17][18][19][20]. It use a event camera to obtain asynchronous and low-latency event stream. With the tracked feature points and the already built 3D map, we can estimate the current pose of camera by solving PnP (Perspective-n-Point) problem[21], which minimizes the reprojection error of feature points. After that, the depth of feature points can be calculated by triangulation[22]. Due to the inevitable error of the measurement, we introduce depth filter[17][23] to model the depth estimation of the feature points with a probabilistic distribution. When the variance of the depth distribution is lower than the threshold after several updates, the corresponding point will be inserted into the 3D map.

It is noticeable that these two subsystems (pose estimation and depth estimation) rely on each other since the pose estimation needs the global 3D map while the

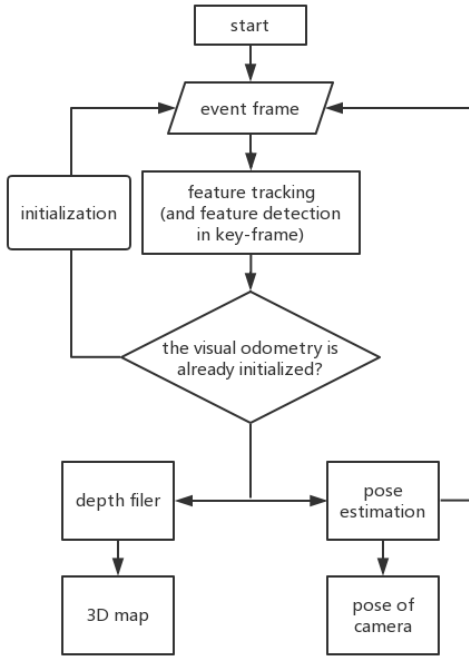


Fig. 1. The framework of our event based visual odometry system

update of depth filter needs the current camera pose. But there is no available map when our visual odometry system begins to work. For this reason, we estimate the camera pose by 2D-2D methods in initialization. The 2D-2D pose estimation method that are most widely used is based on Homography[24], but it is not appropriate for our target applicational scenario because most feature points in driving scenario are not on the same plane. So our visual odometry system utilizes Eight-Point-Algorithm[25] for the essential matrix, which can then be used to reconstruct the rotation and transfer between the first two event frames.

In order to improve the computational efficiency and decrease the storage space, our visual odometry system only regularly stores an event frame as a key frame. When dealing with these key frames, our visual odometry system detects feature points and resets the depth filter while it only tracks the previous detected feature points and updates depth filter on ordinary frames.

A. Feature tracking

The feature detection and feature tracking algorithm utilized in our system is proposed by Zhu et al. in [14]. Fig.2 presents the process of this algorithm. First, the event points in a adjustable time interval are aggregated into a event frame. Secondly, this frame is corrected by Expectation Maximization optical flow estimation to propagate the events within a spatiotemporal window to the initial time t_0 . The corrected image is similar to an edge map, then the feature points are detected by Harris corner detector on it. The final step is the feature alignment between the continuous corrected frames. Fig.2.(d)

shows two continuous corrected event frames before affine warping and Fig.2.(e) shows the result of affine warping.

B. Pose tracking

Our visual odometry system utilizes bundle adjustment to estimate the camera pose through the minimization of reprojection error $e(\xi)$, which is defined by

$$e(\xi) = u_i - \frac{1}{Z_i} K * \exp(\xi^\wedge) P_i \quad (1)$$

where ξ is the camera pose ($\xi \in se(3), T \in SE(3)$), u_i is the pixel coordinate of feature points in event frame, Z_i is depth, K is camera intrinsics and P_i is the coordinate of 3D point in the global map. The first order Taylor expansion of $e(\xi)$ is

$$e(\xi + \Delta\xi) \approx e(\xi) + J(\xi)\Delta\xi \quad (2)$$

where $J(\xi)$ is the derivative of $e(\xi)$ and a 2×6 Jacobian matrix. Hence, the target $\Delta\xi$ is

$$\Delta\xi = \arg \min_{\Delta\xi} \frac{1}{2} \|e(\xi) + J(\xi)\Delta\xi\|_2^2 \quad (3)$$

The result of Eq.3 is when the derivation of $e(\xi)$ to $\Delta\xi$ is 0

$$J(\xi)^T J(\xi)\Delta\xi = -J(\xi)e(\xi) \quad (4)$$

since there are n feature points in the event frames, Eq.4 is transformed as follow

$$\sum_{i=1}^n w_i J(\xi)^T J(\xi)\Delta\xi = -\sum_{i=1}^n w_i J(\xi)e(\xi) \quad (5)$$

where w_i is the weight based on the precise of reprojection[10]. Since $J(\xi)$ and $e(\xi)$ of each feature points are both known, we can calculate $\Delta\xi$ from Eq.5 in iterations until $\Delta\xi$ is small enough.

C. Mapping

In the last subsection we have estimated the pose of camera by bundle adjustment, then we can estimate depth of feature points by triangulation. Supposed that the homogeneous coordinates of a 3D point in two reference frames of camera are x_1 and x_2 respectively ($x_1 = (u_1, v_1, 1)^T$ and $x_2 = (u_2, v_2, 1)^T$). The geometry constraint between x_1 and x_2 is

$$Z_2 x_2 = Z_1 R x_1 + t \quad (6)$$

where Z_1 and Z_2 are depth, R and t are rotation and transfer between two event frames. After left multiplying x_2^\wedge on both sides of Eq.6, the equation is transformed to

$$Z_1 x_2^\wedge R x_1 + x_2^\wedge t = Z_2 x_2^\wedge x_2 = 0 \quad (7)$$

From Eq.7 we can calculate Z_1 and Z_2 for each feature point as a rough estimation. Because of the inevitable measurement error of depth estimation, we introduce depth filter to model the depth estimation of the feature points with a probability distribution. This depth filter was proposed in [23], which models the measurement \hat{d}_i^k (k-th

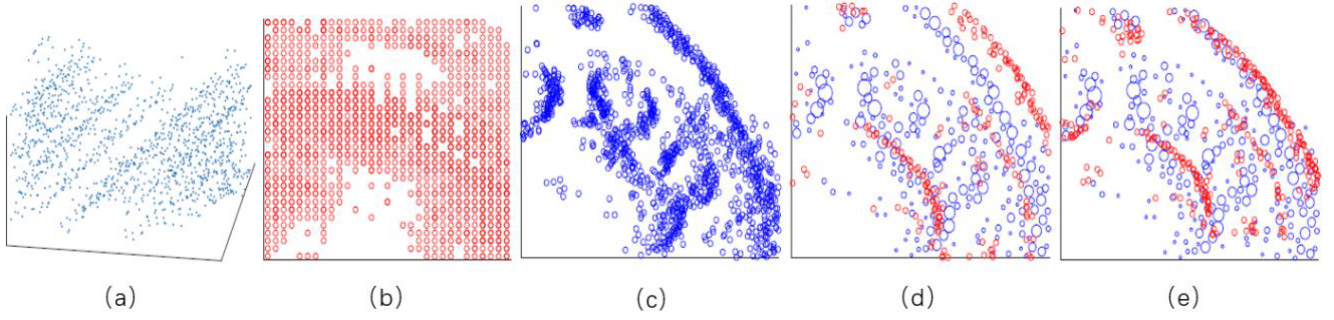


Fig. 2. The procedure of feature tracking: (a) the ordinary event stream in spatiotemporal coordinate system; (b) events aggregated in an event frame without flow correction; (c) the corrected event frame; (d) two corrected event frames before affine warping; (e) the result of affine warping (from [14])

observation of the i -th feature) using a Gaussian+Uniform mixture distribution:

$$p(\tilde{d}_i^k | d_i, \rho_i) = \rho_i N(\tilde{d}_i^k | d_i, \tau_i^2) + (1 - \rho_i) U(\tilde{d}_i^k | d_i^{min}, d_i^{max}) \quad (8)$$

where ρ_i is the inlier probability, which is close to 1 when the feature is well-tracked, τ_i^2 is the variance caused by one-pixel disparity in the image plane and can be computed by geometric constraint. $[d_i^{min}, d_i^{max}]$ is the known range of depth estimation. The update for a new measurement \tilde{d}_i^k is shown in Fig.3. And the detail of update of d_i and ρ_i is described in [23]. When the uncertainty of the depth distribution is smaller than the threshold, the corresponding feature point will be inserted into the global map for further camera pose estimation.

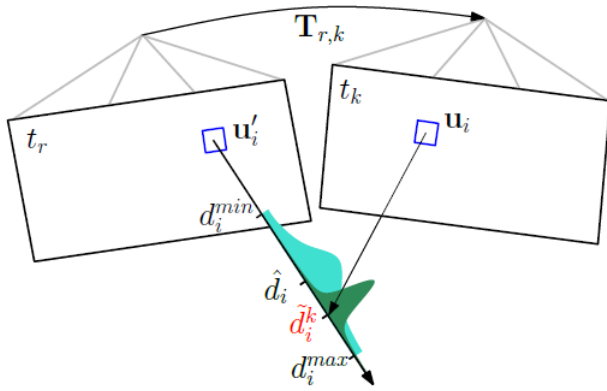


Fig. 3. Depth-filter update for a new measurement \tilde{d}_i^k at time t_k . The uncertainty of depth distribution becomes narrower. (from [17])

D. Parallel tracking and mapping

The framework of our event based visual odometry system has been shown in Fig.1. The input is event stream consisted of series of event tuple $[t, x, y, p]$. In order to improve the computational performance, the feature points are only detected regular in the key frames, and the corresponding depth filter are reset. Regarding the ordinary event frames, our visual odometry system tracks

the feature points to estimate the pose of camera and update the depth filter.

The pose estimation and the update of depth filter are handled by two main threads in our system. In pose estimation main thread, our visual odometry system utilizes bundle adjustment to estimate camera pose based on the tracked feature points in the current event frame and a already built map. In depth estimation main thread, our visual odometry system updates the depth filter with rotation and transfer between two consecutive frames. If the depth uncertainty of a feature point is lower than the threshold, this feature point will be inserted into the global map. The reason why our system works in two parallel threads is that pose tracking must be synchronous with the input of event camera while the real-time constraint of depth estimation main thread is not so strong relatively. Since the computational burden of depth estimation is relative greater, it's possible that a new event frame is received before the depth filters have been completely updated. In this case, our visual odometry system will store the pending event frames in a FIFO queue for the depth estimation.

E. Bootstrapping

Since two main threads of our visual odometry system (pose tracking and depth estimation) rely on each other but the visual odometry system has no available 3D global map for bundle adjustment when dealing with the first two event frames, our system utilizes Eight-point-algorithm[26] for essential matrix in bootstrapping. Supposed that the homogeneous coordinate of a 3D point in two reference frame of camera are x_1 and x_2 respectively ($x_1 = (u_1, v_1, 1)^T$ and $x_2 = (u_2, v_2, 1)^T$). The geometric constraint between x_1 and x_2 is

$$Z_2 x_2 = Z_1 R x_1 + t \quad (9)$$

After left multiplying $x_2^T t^\wedge$ on both sides of Eq.9, the equation is transformed to

$$x_2^T t^\wedge R x_1 = 0 \quad (10)$$

$$\begin{pmatrix} u_2 & v_2 & 1 \end{pmatrix} \begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = 0 \quad (11)$$

where $t^{\wedge}R$ is a essential matrix and e_i is the element of it (the depth can be neglected as a scalar multiplication). After converting the essential matrix to vector form, Eq.11 is transformed to

$$\begin{pmatrix} u_1u_2 & u_1v_2 & u_1 & v_1u_2 & v_1v_2 & v_1 & u_2 & v_2 & 1 \end{pmatrix} * e = 0 \quad (12)$$

where

$$e = (e_1 \ e_2 \ \dots \ e_9)^T \quad (13)$$

There are 9 elements in the essential matrix, but since the essential matrix is defined only up to scale, if we can detect 8 pairs of feature points in the first two event frames, then we can obtain essential matrix to compute the rotation and transfer between the first two event frames.

IV. Experiment

A. Dataset

Two datasets are used in this experiment. One is The Event-Camera Dataset and Simulator[27] provided by Robotics and Perception Group from the University of Zurich, the other one is Multi Vehicle Stereo Event Camera Dataset[28] from Pennsylvania University.

Event-Camera Dataset and Simulator dataset was recorded by DAVIS camera (the version is DAVIS240C). Most of the data are recorded indoors by event camera, conventional camera and IMU, and this dataset also provide the ground truth of camera pose. As for the outdoor data, a precise pose estimation is provided by IMU. This dataset is recorded on hand-held device or drone instead of vehicle, that is different from our target application scenario, but this dataset provides two file formats of recording: txt and rosbag, and the txt-format data provides dramatic convenience for the debug phase of our system. Fig.4 shows some scenarios in this dataset.



Fig. 4. Some scenarios in Event-Camera Dataset and Simulator dataset

The Multi Vehicle Stereo Event Camera Dataset was recorded by different sensors (event camera, conventional camera, 16-beam LiDAR, GPS, IMU) equipped on different platforms (car, motor cycle, drone). The data recorded by event camera equipped on vehicle are mostly from the urban roads at both day and night, and the pose of camera is calculated by localization algorithm based on LiDAR, IMU and GPS. This dataset includes several driving scenarios such as start-up acceleration, constant speed driving, deceleration at crossing and parking with

great changes of illumination and the noise introduced by dynamic objects.

Additionally, our experiment is carried out on ROS Melodic. Open source libraries such as Eigen, Sophus, OpenCV, g2o, Boost are also utilized in our visual odometry system.

B. Feature detection and tracking

The experiment of feature detection and tracking is carried out in scenarios of both daytime and night in Multi Vehicle Stereo Event Camera Dataset.

Some grayscale images and event frames of daytime scenario are shown in Fig.5 and Fig.6. Thanks to high dynamic range of event cameras, the event frames show the robustness against the dramatic illumination change and partial overexposure. The Fig.7 shows the feature tracking in four continuous event frames.



Fig. 5. The grayscale images of daytime in Multi Vehicle Stereo Event Camera Dataset

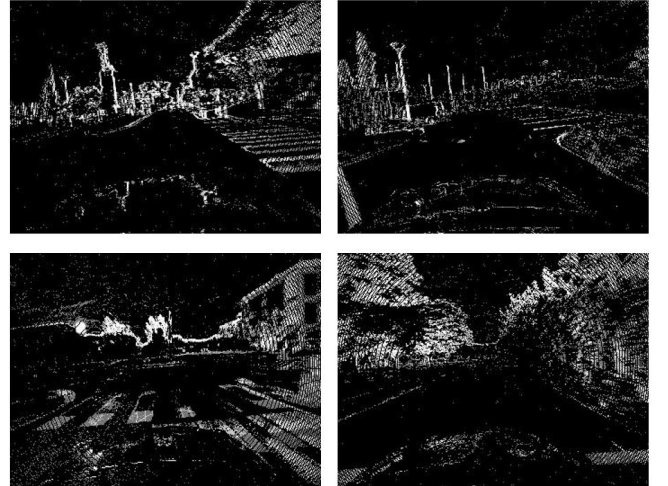


Fig. 6. The event frames of daytime in Multi Vehicle Stereo Event Camera Dataset

Some grayscale images and event frames of night scenario are shown in Fig.8 and Fig.9. Due to the poor

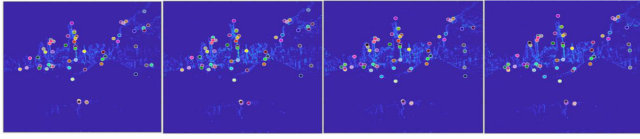


Fig. 7. The feature tracking in four continuous daytime event frames (the same feature points are represented by the same color)

illumination at night, it is quite hard to distinguish the vehicles and buildings in the dark. In addition, artificial lights cause partial overexposure. These problems are alleviated in the output of event camera, but the data recorded from the event camera contains relatively much more noise, which is a main drawback of the current event camera. Fig.10 shows the feature tracking in four continuous event frames.



Fig. 8. The grayscale images of night in Multi Vehicle Stereo Event Camera Dataset

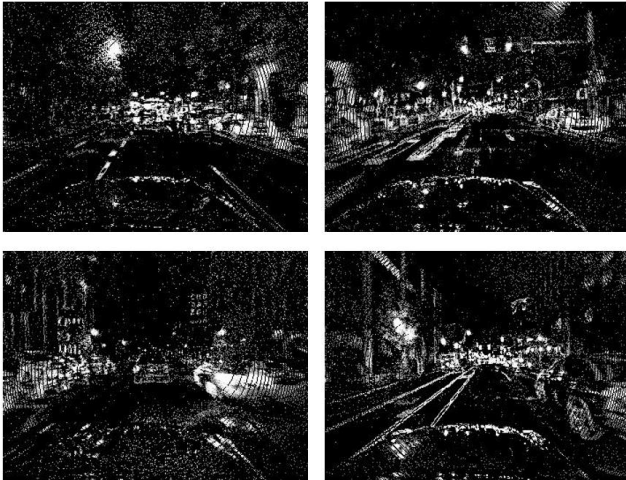


Fig. 9. The event frames of night in Multi Vehicle Stereo Event Camera Dataset

The stability of feature tracking is an important criterion in visual odometry system, because the variance of

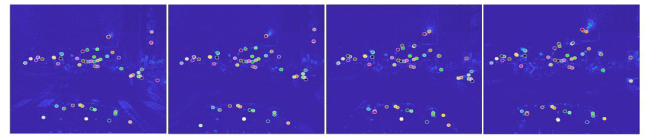


Fig. 10. The feature tracking in four continuous night event frames (the same feature points are represented by the same color)

TABLE II
The average lifetime of feature points in daytime and night scenarios

Item	Daytime	Night
Average lifetime	16.488 frames	14.321 frames
Standard variance	18.505 frames	15.687 frames

depth distribution can not converge unless feature points can be tracked in enough continuous event frames. And the feature detection is carried out only in key event frames, so a feature point can not be tracked once it is lost in an ordinary event frame. We adjust the stability of our system using 'lifetime'[29], which is the span of event frames from a feature point is detected until it is lost. The statistics result is shown in Table 2 (the feature points whose lifetime is less than 3 event frames will be ignored) and the distribution of lifetime in daytime and night scenarios are shown in Fig.11 and Fig.12 respectively.

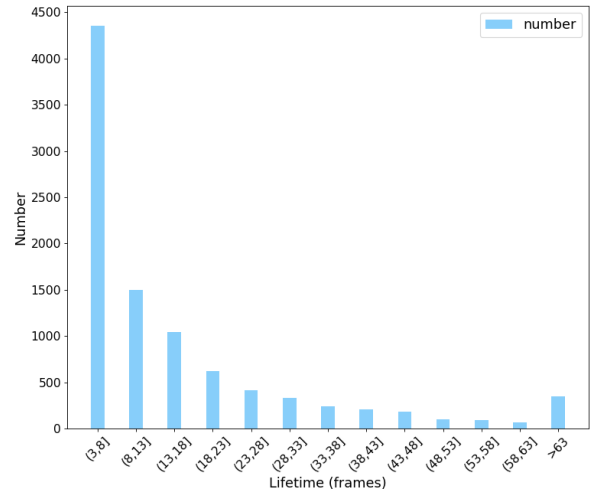


Fig. 11. The distribution of lifetime in daytime scenario

The average lifetime of feature points in daytime and night scenarios are 16.488 frames and 14.321 frames respectively. Considering that the variance of depth filter can converge after 5-7 iterations, the stability of feature tracking meets the requirement generally. It can also be seen that the difference between the tracking performance in daytime and night scenarios is much smaller than that of conventional visual odometry system, which proves that

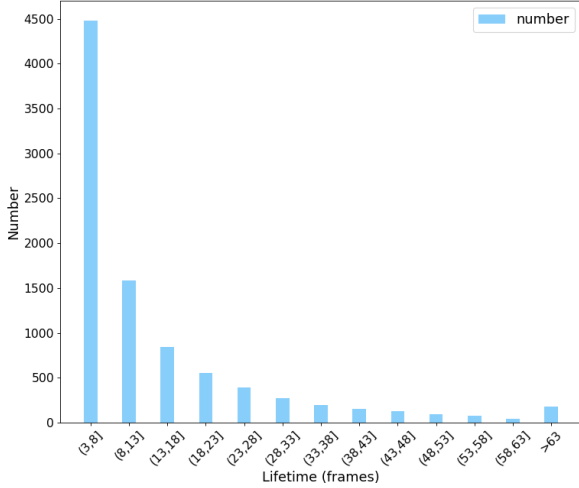


Fig. 12. The distribution of lifetime in night scenario

the event camera can detect the edges efficiently even under poor illumination condition. But the real-time performance of this feature tracking algorithm is quite poor. Using the source code provided by Zhu et al., our visual odometry system is far from meeting the requirement of tracking about 100 feature points synchronously, so until now our system can only run in offline mode. In [14], Zhu et al. claimed that they are improving the real-time performance of the algorithm to track more feature points synchronously, we will continuously follow their research project.

C. Event based visual odometry system

Since the real-time performance of the feature tracking has not been implemented, we first record the result of feature tracking (the identifier of feature points, the coordinate in event frames) in a txt-format file and then utilize this file in our odometry experiment. In Multi Vehicle Stereo Event Camera dataset, the localization is provided by GPS and other algorithms like LOAM and Cartographer, and we finally choose the localization provided by LOAM, which is based on IMU, as the ground truth. Fig.13 shows the visualization of the estimated trajectory in our experiment.

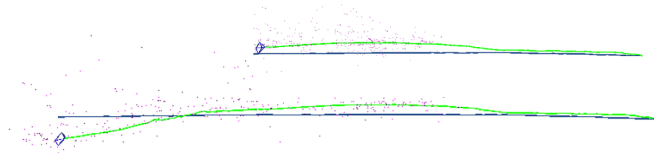


Fig. 13. The localization estimation from our event based visual odometry system and the ground truth (the blue line is the ground truth provided by LOAM and the green line is the trajectory estimated by our visual odometry system)

TABLE III
The statistics result of positional error

Item	Longitudinal	Lateral	Planar
Average error	0.488 m	0.275 m	0.581 m
Relative error	0.4399%	0.3464%	0.5578%

The timestamp of the event frames used in this experiment is from 12s to 72s and the driving distance in this interval is about 439 meters. The localization error in this interval is shown in Fig.14 and TABLE III. The

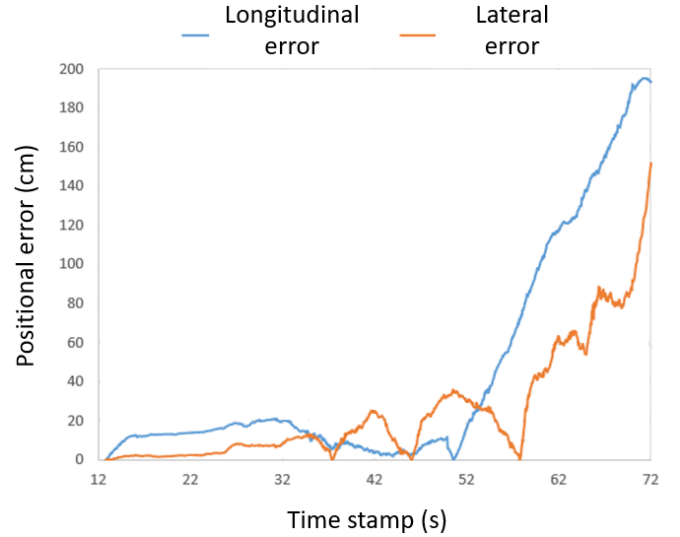


Fig. 14. The localization error of our event based visual odometry system in the interval (from 12s to 72s)

average error of planar localization is 0.581 m, which is larger than the most visual odometry system based on the conventional cameras. Additionally, the relative position error of our odometry system is 0.5578% while the relative position error of EVO[2] system is only 0.2%. In Fig.14 we can see that the longitudinal and lateral positional error are kept within 0.2 meter before 52s but the position error expands dramatically after 52s since the vehicle passed two bumps.

V. Conclusion

In this paper we design a event based visual odometry system that utilizes feature tracking to achieve parallel pose estimation and mapping. We test our system on dataset about urban road scenario and the result shows that our system has a good performance on feature tracking and robustness against extreme illumination changes and dark scenarios. But the real-time performance of our visual odometry system is poor because of the huge computational burden of feature tracking. In addition, the localization error of our system increase sharply when the vehicle passes a bump. But due to the low latency, the effective detection of edges and other aforementioned

advantages, event camera is still a promising sensor for autonomous driving. For this reason, we will keep improving the real-time performance and the robustness against bumpy road of our event based visual odometry system.

Acknowledgment

The research leading to these results has partially received funding from the Shanghai Automotive Industry Sci-Tech Development Program (Grant. No. 1838), from the Shanghai AI Innovative Development Project 2018, and from the Young Scientists Fund of the National Natural Science Foundation of China (Grant No. 6190021037).

References

- [1] G. Chen, H. Cao, M. Aafaque, J. Chen, and C. Ye, "Neuromorphic vision based multivehicle detection and tracking for intelligent transportation system," *Journal of Advanced Transportation*, vol. 2018, no. 4815383, p. 13, 2018.
- [2] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza, "Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real-time," *IEEE Robotics & Automation Letters*, vol. 2, no. 2, pp. 593–600, 2017.
- [3] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger, "Interacting maps for fast visual interpretation," in *International Joint Conference on Neural Networks*, 2011.
- [4] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. Davison, "Simultaneous mosaicing and tracking with an event camera," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.
- [5] D. Weikersdorfer, R. Hoffmann, and J. Conradt, *Simultaneous Localization and Mapping for Event-Based Vision Systems*, 2013.
- [6] A. Censi and D. Scaramuzza, "Low-latency event-based visual odometry," in *IEEE International Conference on Robotics & Automation*, 2014.
- [7] D. Weikersdorfer, D. B. Adrian, D. Cremers, and J. Conradt, "Event-based 3d slam with a depth-augmented dynamic vision sensor," in *IEEE International Conference on Robotics & Automation*, 2014.
- [8] E. Mueggler, B. Huber, and D. Scaramuzza, "Event-based, 6-dof pose tracking for high-speed maneuvers," in *IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2014.
- [9] G. Gallego, J. E. A. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, "Event-based, 6-dof camera tracking for high-speed applications," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2016.
- [10] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, "Low-latency visual odometry using event-based feature tracks," in *IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2016.
- [11] H. Kim, S. Leutenegger, and A. J. Davison, *Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera*, 2016.
- [12] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, "Emvs: Event-based multi-view stereo—3d reconstruction with an event camera in real-time," *International Journal of Computer Vision*, no. 2, pp. 1–21, 2017.
- [13] G. Gallego and D. Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robotics & Automation Letters*, vol. PP, no. 99, pp. 1–1, 2017.
- [14] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2017.
- [15] M. Elias, G. Guillermo, R. Henri, and S. Davide, "Continuous-time visual-inertial odometry for event cameras," *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1–16, 2017.
- [16] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high speed scenarios," *IEEE Robotics & Automation Letters*, vol. 3, no. 2, pp. 994–1001, 2018.
- [17] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics & Automation*, 2014.
- [18] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *IEEE & Acm International Symposium on Mixed & Augmented Reality*, 2008.
- [19] R. Murartal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2017.
- [20] J. Engel, T. Schöps, and D. Cremers, *LSD-SLAM: Large-Scale Direct Monocular SLAM*, 2014.
- [21] A. Hartley and A. Zisserman, *Multiple view geometry in computer vision* (2. ed.), 2003.
- [22] B. A. Maxwell and S. A. Shafer, "Computer vision and image understanding," *Machine Learning & Data Mining Methods & Applications*, vol. 72, no. 2, pp. 143–162(20), 2003.
- [23] G. Vogiatzis and C. Hernández, "Video-based, real-time multi-view stereo," *Image & Vision Computing*, vol. 29, no. 7, pp. 434–441, 2011.
- [24] A. Agarwal, C. V. Jawahar, and P. J. Narayanan, "A survey of planar homography estimation techniques," in *Technical Reports*, International Institute of Information Technology, 2005.
- [25] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, pp. 133–135, 1981.
- [26] —, "A computer algorithm for reconstructing a scene from two projections," vol. 293, no. 5828, pp. 133–135, 1981.
- [27] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset: Event-based data for pose estimation, visual odometry, and slam," *International Journal of Robotics Research*, vol. 36, no. 49, pp. 142–149, 2017.
- [28] A. Z. Zhu, D. Thakur, T. Ozaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multi vehicle stereo event camera dataset: An event camera dataset for 3d perception," *IEEE Robotics & Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018.
- [29] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza, "Lifetime estimation of events from dynamic vision sensors," in *IEEE International Conference on Robotics & Automation*, 2015.