Towards Dynamic Monocular Visual Odometry Based on an Event Camera and IMU Sensor

Sherif A.S. Mohamed[1(✉)], Mohammad-Hashem Haghbayan[1], Mohammed Rabah[3], Jukka Heikkonen[1], Hannu Tenhunen[1,2], and Juha Plosila[1]

[1] University of Turku (UTU), 20500 Turku, Finland
samoha@utu.fi
[2] Royal Institute of Technology (KTH), 11419 Stockholm, Sweden
[3] Kunsan National University (KNU), Gunsan 54150, South Korea

Abstract. Visual odometry (VO) and visual simultaneous localization and mapping (V-SLAM) have gained a lot of attention in the field of autonomous robots due to the high amount of information per unit cost vision sensors can provide. One main problem in VO techniques is the high amount of data that a pixelated image has, affecting negatively the overall performance of such techniques. An event-based camera, as an alternative to a normal frame-based camera, is a prominent candidate to solve this problem by considering only pixel changes in consecutive events that can be observed with high time resolution. However, processing the event data that is captured by event-based cameras requires specific algorithms to extract and track features applicable for odometry. We propose a novel approach to process the data of an event-based camera and use it for odometry. It is a hybrid method that combines the abilities of event-based and frame-based cameras to reach a near-optimal solution for VO. Our approach can be split into two main contributions that are (1) using information theory and non-euclidean geometry to estimate the number of events that should be processed for efficient odometry and (2) using a normal pixelated frame to determine the location of features in an event-based camera. The obtained experimental results show that our proposed technique can significantly increase performance while keeping the accuracy of pose estimation in an acceptable range.

Keywords: Event-based camera · Monocular · Visual-odometry · IMU

1 Introduction

Visual odometry (VO) is one of the most popular topics in machine vision (MV) that is used in broad types of applications, such as autonomous navigation, object avoidance, and 3D scene reconstruction. VO estimates the position and orientation of a moving platform by analyzing the variations induced by the

A. L. Martins et al. (Eds.): INTSYS 2019, LNICST 310, pp. 249–263, 2020.
https://doi.org/10.1007/978-3-030-38822-5_17

motion of the camera on a sequence of consecutive images, i.e., *ego-motion estimation*. *Frame-based cameras* are widely used in conventional VO approaches, because they can provide high-resolution images with low cost and have a similar output to human vision. In VO techniques based on frame-based camera, the location of the camera is reconstructed by computing the optical flow (OF) from key information extracted from two consecutive *frames*. The key information in a frame, e.g., corners, is extracted using a *frame feature detector*, such as *Moravec* [1] or *Harris* [2], and the reconstructed scene can be refined using bundle adjustment [3] or another offline optimization method. Even though there has been significant advancement in the field of odomery based on frame-based cameras, there still exist practical problems in using such cameras in odometry, for example high latency of image delivery, motion blur phenomenon, and low dynamic range, which negatively affect the efficiency of the odometry algorithm w.r.t. the accuracy of the result and performance.

Another type of camera that can be used for odometry is the *event-based camera* [4]. Opposite to frame-based cameras that acquire the intensity of all pixels simultaneously and generate frames at fixed rates, event-based cameras use biologically inspired vision sensors to output pixel-level temporal intensity changes, i.e., *events*. This feature of event-based cameras makes them very appealing and efficient for odometry. An event is triggered whenever the brightness of a pixel changes. In such a case, the location of an event in a pixelated image *(u,v)*, polarity of the brightness change (1 or 0), and time-stamp, are passed as a single event to the camera output. Therefore, such cameras produce a stream of events that has no redundant data and can thereby reduce the latency (response time) down to 10 μs. Moreover, the power consumption of odometry can be significantly reduced, even by the factor of 50, which is an important aspect especially for resource limited devices.

The benefits event-based cameras provide make them attractive for odometry in navigation and tracking on high speed agile robotic platforms that operate under challenging lighting conditions. However, processing data of event-based cameras for odometry is not straightforward, since the output of these cameras is fundamentally different from that of frame-based cameras. For example, unlike in frame-based cameras, features cannot be extracted easily in event-based cameras, making odometry difficult in practice. Moreover, reconstruction of a frame based on data captured by an event-based camera is problematic. The main question here is that how many events should be considered together to form an instantaneous frame? There are recent studies that aim at solving these problems by defining the new features of event-based cameras and by determining the number of events per frame in a given time interval [5]. However, such techniques are efficient only in special situations, e.g., when the number of events is not changing for different scenes, and, therefore, they do not offer general solutions. The main drawback of such techniques is that their efficiency, in terms of performance and accuracy, is dependent on the velocity of the camera and the number of events in a scene. In dynamic situations, where the scene and velocity of the camera change drastically, i.e., the number of events in a frame changes

rapidly in time, processing the events for accurate and fast (real-time) odometry poses still a big challenge.

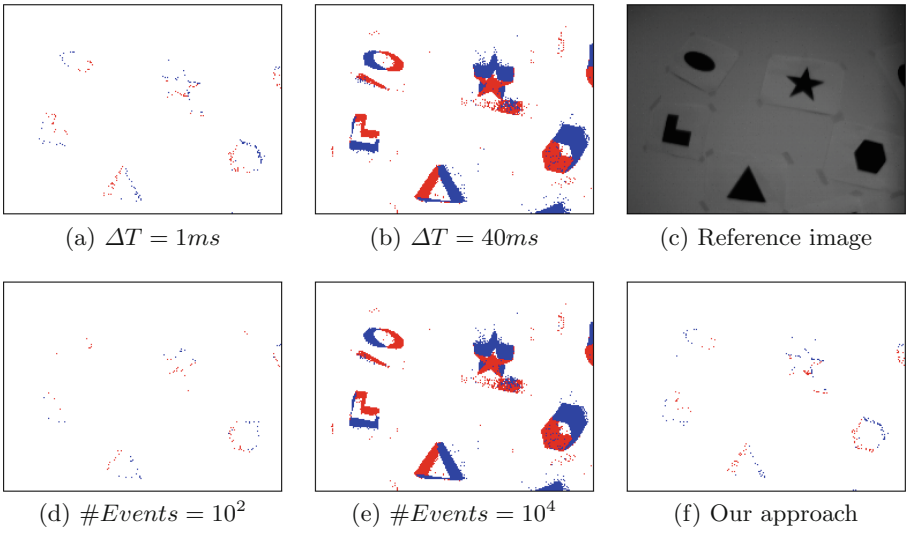
In this paper, we propose a hybrid technique for odometry based on data captured by both frame-based and event-based cameras, combining the ability of frame-based cameras to detect and track features and the low latency and high dynamic range of event-based cameras to achieve fast and accurate odometry. To do this, our proposed techniques can be divided into two main novel methods that are: (1) defining dynamically the number of events for a frame to reach a minimized amount of data processing for a scene while keeping the accuracy of pose estimation, and (2) using a frame-based camera as a reference guide for an event-based camera to recognize and track features in an event-based camera output. To determine the number of events to construct a frame in run-time, we define two factors that together affect the number of events in an instantaneous scene: (1) *entropy* of the scene that is the entropy of a pixelated image [6], i.e., the amount of *information* the image can convey from an event-based camera, and (2) velocity of the camera that determines how fast the environment changes and is one of the main factors in ego-motion. In this definition, we assume all the objects in a scene are stationary and only the camera is moving. To estimate the velocity of the scene, we use IMU data that can report the acceleration of motion in run time. To extract features from the output of an event-based camera, we use a common method, FAST [7], to first extract features from a frame-based image and then use this information to initiate tracking of those features in the event-based camera output. After this, the detected features on the event-based camera are tracked. Periodically, features from the frame-based camera are used to correct the error of feature tracking in the event-based camera.

We organize the remaining part of this paper as follows. In Sect. 2, we demonstrate motivation to show the existing problem in odometry based on event-based camera and review the related work in visual odometry for both traditional and event cameras. In Sect. 3, we illustrate the overall system workflow, which consists of three steps: event frame generator, feature tracking, and visual odometry. In Sect. 4, we present the experimental results. Finally, in Sect. 5, we draw the conclusions.

2 Motivation and Related Work

In standard frame-based cameras, e.g., those cameras with global-shutter or rolling-shutter sensors, images are generated at a fixed rate by obtaining the intensity of pixels in the whole image simultaneously. To estimate the pose based on these cameras, two main problems might occur. The first problem is the amount of redundant repetitive data that the next frame might contain in the case where the scene does not change much, i.e., the information in the image is low or the movement of the camera is slow. Such amount of redundant data takes unnecessary transfer and process cost and does not add any new information w.r.t. the previously captured data. The second problem, from the other side, is the amount of information that might be missed between two frames, i.e., *blind*

time, when the change in the scene is too fast due to rapid movement of the camera and high amount of information in the scene between two frames. To solve this problem, one solution is to dynamically change the frame rate of the camera based on the speed and entropy of the captured image[1]. However, this solution does not solve the problem of redundant repetitive data between two frames, and also there is a strict limitation to change the frame rates of those cameras.



(a) $\Delta T = 1ms$ (b) $\Delta T = 40ms$ (c) Reference image

(d) $\#Events = 10^2$ (e) $\#Events = 10^4$ (f) Our approach

Fig. 1. The effect of applying different techniques to define the number of events for a frame on the resolution of the output image while rapid movement

Event-based cameras solve this problem by transmitting a stream of asynchronous events that happen in the pixels of an image. Therefore, instead of reporting the scene at each time interval, like in the case of frame-based cameras, only the events are reported, and the new scene can be updated based on the events and the current history of the scene. If the change in the scene is slow, then the number of generated events is small, and the information can be processed fast. Since the accepted interval between two events is small, in the range of microseconds, such cameras can transfer the information at a very high resolution in the cases where the change in the scene is very fast, making this approach well-suited for cases where fast actions are needed due rapid changes in the scene.

As mentioned in the introduction part, even though an event-based camera provides rich and small data that is suitable to be transferred and computed

[1] This technique is widely used in cinema to show the importance or inferiority of a scene by applying slow-motion, fast motion, and time-elapsed photography.

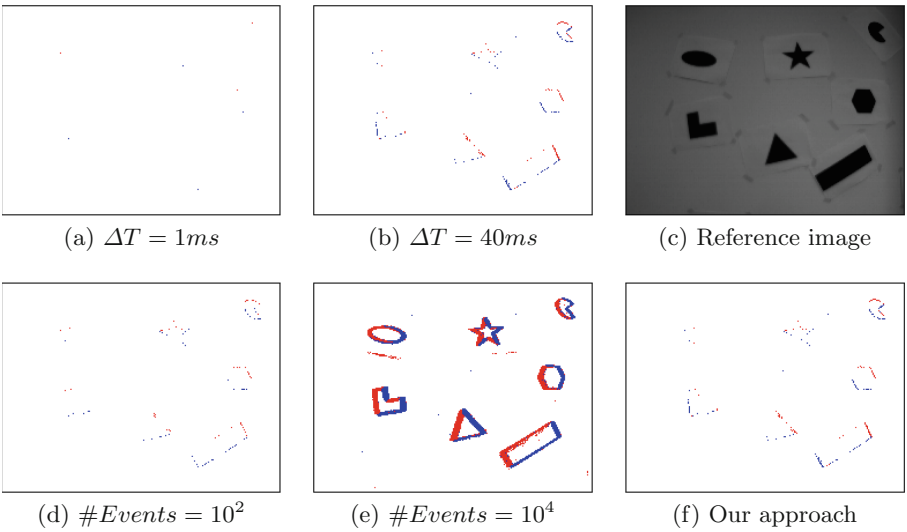


Fig. 2. The effect of applying different techniques to define the number of events for a frame on the resolution of the output image while slow movement

fast and accurately, the captured data requires specific processing to be used for odometry. To process a scene that is constructed from event-based camera data, a frame of such a scene should be first constructed based on the captured events. The first issue that makes such processing challenging is the number of events that can be considered to make a frame. Indeed, since an event-based camera only reports a stream of events, determining the suitable amount of events to make a frame becomes an important problem. In [8], the authors propose a fixed time interval, i.e., ΔT, and the events accumulated during this interval are considered a frame. In [5], the author proposes a fixed number of events, i.e., $\#Events$, to form a frame. Using a fixed ΔT or $\#Events$ causes a problem corresponding to a fixed frame rate in frame-based cameras, i.e., inflexibility in dynamic situations

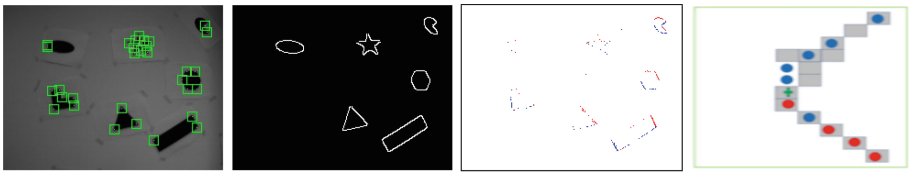


Fig. 3. Feature detection and matching. (a) Frame with detected corners and patches (green boxes). (b) edge map using Canny detector. (c) accumulated events in ΔT time. (d) (zoom for a patch) point sets used for feature matching: edges (in gray) and events in (red and blue). (Color figure online)

where the number of events for different frames varies significantly. Inspired by this observation, we propose a technique to dynamically calculate the number of events based on the velocity of the camera and entropy of the capture pixelated image of the scene. Figures 1 and 2 show the reconstructed frames based on two different values for ΔT and $\#Events$ and in two different situations wherein the velocity of the camera is high (a high amount of information per unit time) and low (a low amount of information per unit time), respectively. At the right side of Figs. 1 and 2, the result of our proposed dynamic frame construction technique is shown to demonstrate how dynamicity in detecting $\#Events$ can help to reconstruct the frame in these two different environmental conditions. As can be seen in these two figures, while using a large ΔT and $\#Events$, the constructed frame is blurry and non-informative when the velocity of the camera is slow. In contrary, small ΔT and $\#Events$ result in a weakly depicted frame for rapid motion and is not suitable for odometry.

After frame construction, the next step is to utilize the reconstructed frame for odometry. As indicated above, the event-based approach requires a specific technique to process the frame, differing from traditional odometry that is based on frame-based cameras. Generally, the traditional odometry techniques can be divided into two main strategies that are (1) *feature-based*, also know as *indirect*, and (2) *direct* methods. In a feature-based method, instead of processing all the pixels in an image, some selected interest points, i.e., features, are processed. Feature-based techniques can be further categorized into two main branches that are *sparse* and *dense* methods. The sparse feature-based method is the most widely used algorithm to estimate the 6-Dof pose from a set of features that are extracted from an image. The optimization process is performed by minimizing the estimated geometric error without any notion of neighborhood [9, 10]. Dense approaches [11] use the geometric error estimation and geometric prior, i.e., smoothness of the flow field, together for odometry. Direct approaches analyze the intensity of pixels in the image to recover the pose of the camera [12–14]. Sparse direct methods, such as SVO [15] and DSO [16], use only selected pixels in an image, which reduces the computational cost drastically. However, direct methods do not cope very well with large frame-to-frame motions, because they obtain the pose by minimizing the photometric error.

Recently, odometry based on event-based cameras has been used in several SLAM algorithms. Since an event-based camera generates asynchronous events, obtaining the ego-motion, i.e., the 6-DoF motion, is a challenging problem. In [17], the authors propose an algorithm to estimate the rotational ego-motion and reconstruct intensity images based on captured events. In [18], a 2D SLAM system is presented to estimate the planar motion based on captured events. This is extended for 3D in [19] with the help of an extra RGB-D camera. In [20], the authors propose an approach to estimate the 3D rotation of the camera based on a particle filtering framework. In [21], the authors propose a VO system which first extracts features from intensity images and then tracks those features in events produced by an event-based camera. In [22], the authors present a system to estimate the motion and depth of a 3D scene, by reconstructing the image

intensity based on captured events. Most of the mentioned techniques are computationally intensive due to high amount of processing needed to understand the events and process the additional data, especially in the cases where an extra frame-based camera is used.

3 System Architecture of the Proposed Framework

The overall system architecture of the proposed hybrid algorithm is shown in Fig. 4. The system consists of two main novel parts as explained earlier: (1) dynamic frame generation based on captured events and (2) feature extraction and tracking. For the rest of the algorithm we use conventional methods of 3D mapping component and pose optimization.

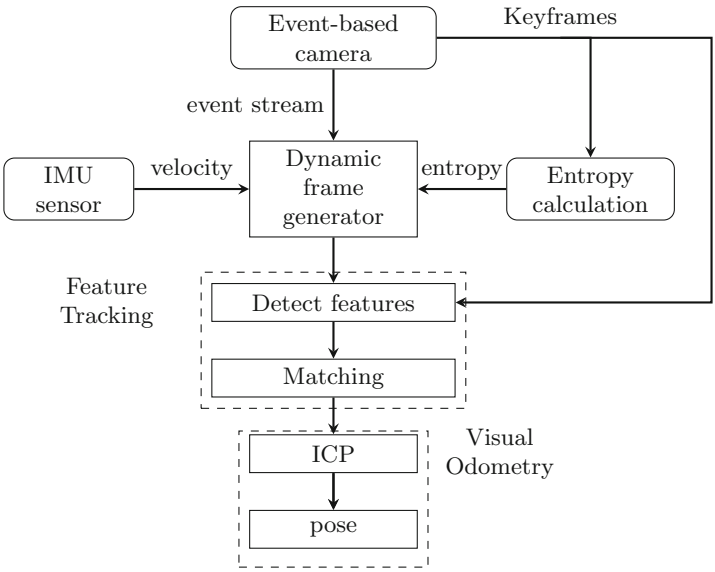


Fig. 4. A overall system architecture of the proposed framework for visual odometry

3.1 Dynamic Event-Based Frame Generation

As mentioned in the previous sections, generating a suitable frame highly depends on richness of information captured by a scene. To estimate this richness, we propose two metrics that are the velocity of the camera and the entropy of the pixelated image of the scene. Entropy or average information in a pixelated image is a common metric used in different vision applications, e.g., automatic image focusing, and can be determined approximately from the histogram of the image, where the histogram shows the different grayscale probabilities in the

image. For example, in automatic image focusing, the state of a camera's focus can be determined by image entropy, i.e., whenever the focus state varies, so does its entropy [23]. Even though image entropy provides richness of a scene, in a moving object, the scene changes over time based on the velocity of the camera in ego-motion. Therefore, velocity is needed as additional information to model the change of entropy over time. This combination results in a suitable metric to estimate how many events are needed to construct a frame. To properly determine such a metric, the relationship of velocity and entropy on the number of events has to be discussed. To find such a relationship, we first show the relationship between the camera velocity and the number of events.

Definition 1: Let V and W be two vector spaces based on the same field F. *linear map* is a function $f : V \rightarrow W$ if for any two vectors u and v in V and any scalar $c \in F$ the following two conditions are always satisfied: $f(u + v) = f(u) + f(v)$ and $f(cu) = cf(u)$. The former condition is called additivity and the latter is called the operation of the scalar multiplication.

Based on this definition we can formulate the following lemma:

Lemma 1: In a pinhole camera with a projection matrix, the velocity of the 2D pixels associated to a constant object in 3D environment is the result of *linear map* of the camera velocity.

Proof: In a pin hole camera, a 3D point in the *world metric coordinate system* $X = [X, Y, Z, 1]^T$ can be mapped onto a 2D point in the *image pixel coordinate system* $x = [u, v, 1]^T$ by knowing the mathematical model of the camera, i.e., the intrinsic and extrinsic parameters of the camera[2]. If it is assumed that the origin of the world coordinates and camera projection are the same, and the Z axis of the camera, i.e., the principal axis, and the world coordinate lie on each other, then the point x is calculated as follows:

$$\begin{bmatrix} \mathrm{x}_p \\ \mathrm{y}_p \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f_u & \alpha_u & \mathrm{u}_0 & 0 \\ 0 & \mathrm{f}_v & \mathrm{v}_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{bmatrix} \tag{1}$$

where f_u and f_v are the focal length in the dimension of pixels, and u_0 and v_0 are the principle point. The parameter α is determined when the pixels are rectangular and is called the *skew factor* [24]. According to the formula of velocity, which is $v = dx/dt$, where x is the pose and t is the time, we get:

$$v_p = d \begin{bmatrix} \mathrm{x}_p \\ \mathrm{y}_p \\ 1 \end{bmatrix} /dt = \frac{1}{Z} \begin{bmatrix} f_u & \alpha_u & \mathrm{u}_0 & 0 \\ 0 & \mathrm{f}_v & \mathrm{v}_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} d \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{bmatrix} /dt \tag{2}$$

[2] The points are represented by homogeneous vectors in projective geometry.

The above equation shows that the velocity of pixels in an image is a linear map of the velocity of the camera.

In an event-based camera, any variation in the pixels of an image causes an event. Such variation is caused by movement of pixels. Based on this and Lemma 1, it can be concluded that the number of events in an even-based camera has a relationship with the linear map of the velocity of the camera and can be modeled by the transformation matrix of the camera. In other words, to use the velocity as a parameter in the model, we need to consider the velocities of all the linear-mapped pixels. Based on this, we propose our metric value for the velocity, to be considered to determine the number of events in a frame, as follows:

$$\iint_{(x_p, y_p)} d \begin{bmatrix} \mathrm{x}_p \\ \mathrm{y}_p \\ 1 \end{bmatrix} / dt \, dx_p \, dy_p \tag{3}$$

which shows the average summation of all the velocities of the pixels in x and y directions.

Even though the number of events generated by an event-based camera has an relationship with the velocity of the camera, the velocity alone cannot provide us with a good metric to estimate a suitable number of the events in a frame. To understand this fact, let us imagine that the camera is moving very fast but in an empty area with a totally black scene. In such a case, regardless of the velocity of the camera, no data will be generated as an event. Based on this simple example, it can be concluded that another factor should be included in the estimation of the number of events in a frame. This factor highly depends on the amount of contrast a scene provides. Such contrast is directly connected to the amount of information a scene contains. The entropy $H(x)$ of a pixelated frame gives us a metric to measure the amount of information and is calculated as follows:

$$H(x) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{4}$$

where p denotes the occurrence probability of a given intensity and n is the number of pixels in the image.

Figure 5 shows the linear relationship between the number of high intensity pixels in a totally black image and entropy. As can be seen, by increasing the amount of contrast in an image, the entropy linearly increases. Our final metric to determine the number of events in a frame, is the product of the entropy and the result of Eq. 3, providing us a suitable method to estimate the number of events in a scene. It should be mentioned that usually event-based cameras provide also the pixelated normal image that can be used to calculate the entropy. Another important fact is that, calculating the entropy is not necessarily needed to generate each frame, and entropy estimation can be done in a longer epoch than frame reconstruction. As mentioned earlier, in this estimation, we assume

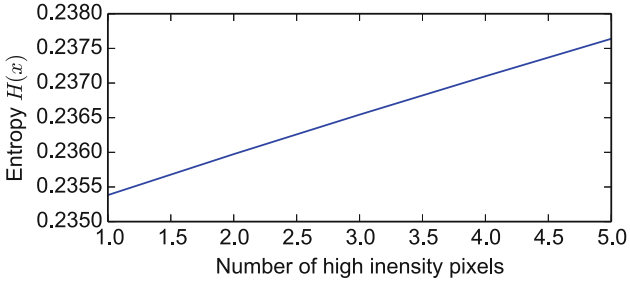


Fig. 5. Relationship between entropy and pixel intensity

for simplicity that all the objects in a scene are stationary and only the camera is moving.

In our proposed techniques in Fig. 4, we use an IMU sensor to estimate the velocity of the camera. IMUs typically consist of an accelerometer and gyroscope unit to obtain linear and angular acceleration at a high data rate, up to 5kHz. Three-axis accelerometer gets merged with the three-axis gyroscope to measure the sensor's angular rate and linear acceleration. In order to acquire the velocity of an object, the current orientation of the IMU is calculated by integrating the gyroscope output. Next, the obtained orientation of the IMU is used to construct a rotation matrix that will transform the accelerometer readings from the IMU "body frame" of reference to the "world frame" of reference. Finally, by integrating the transformed accelerometer output, the current speed of the IMU in the world frame is obtained. After determining the velocity of the camera, this velocity, accompanied by the camera parameters needed for estimating the velocity of pixels in different parts of the image, and the estimation of the image's entropy, are passed to the frame generation module to create a frame based on the estimated number of events.

3.2 Feature Extraction and Tracking

After reconstructing the frame based on the determined number of events, feature extraction and tracking for the captured events is performed. It should be noted that most event-based cameras provide also normal frames that can be used whenever needed. In our proposed algorithm, which is based on the DAVIS [25] event-based camera, we use features extracted from the normal frame-based output of the camera to initiate tracking in event-based frames. To detect features, we use FAST [7] due to its low computational cost and high performance as is shown in Fig. 3. We also use the Canny detector [26] to detect edges as other key features inside patches. To reduce the computational complexity of feature detection in normal frame-based images, the initialization process is performed infrequently (with a long interval), to correct potential errors that might happen in the feature tracking process on event-based frames.

We use Iterative Closest Point (ICP) algorithm [27] to minimize the distance between the edges and events as follow:

$$T_{k,k-1} = \arg\min_{T_{k,k-1}} \sum_{i=1}^{N} \frac{1}{2} \left\| e_i R + t - f_i \right\|^2 \tag{5}$$

where the set of edge features and events in patches are denoted by f_i and e_i, respectively. The Euclidean transformation $T_{k,k-1}$ is obtained which minimizes the distance between the edges points and event points in each detected corner. The operation of this algorithm consists of three main stages: (1) finding point correspondences according to the minimum Euclidean distance, (2) estimating the transformation matrix, and (3) applying the same process on the edge features. The algorithm converges when the error difference between two consecutive iterations is below a given threshold.

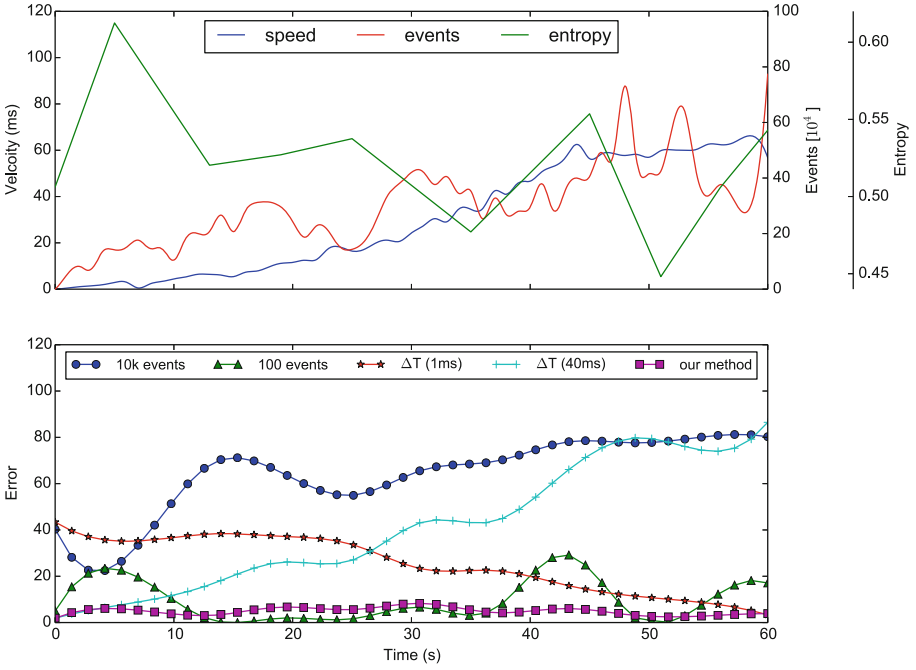


Fig. 6. Comparison of error for several odometry techniques based on an event-based camera. The instantaneous value of the number of detected events for the camera, entropy of the pixelated image of the scene, and velocity of the camera are also shown, used for analyzing the behavior of each technique w.r.t. the change in the environment setup

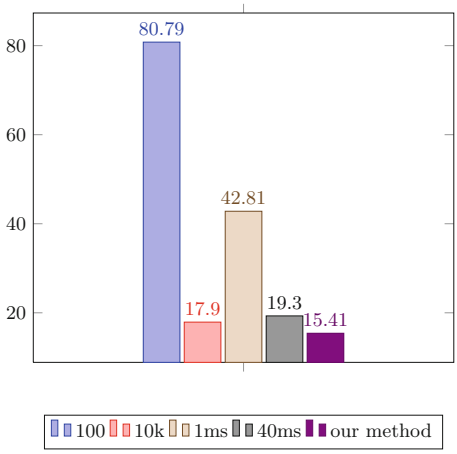


Fig. 7. The measured response time for different odometry techniques based on event-driven camera

4 Experimental Results

We evaluate the proposed event-based VO by running the algorithm considering several situations of the camera movement in normal, rapid, and slow motion. For this, we use the Event-Camera Dataset [28] which contains IMU measurements at 1kHz and many sequences captured with a DAVIS-240C camera. Kalman filtering (KF) is used to merge both accelerometer and gyroscope of noisy IMU data to obtain smooth and rather precise estimation of the acceleration. The camera can capture events and intensity images at the resolution of 240×180. In a normal motion scenario, the DAVIS camera generates up to 10^5 events per second. On the other hand, in rapid camera movements, it can generate up to 1.5 million events per second. The proposed algorithm is tested on a Jetson TX2 board with a quad-core ARM Cortex-A57 CPU @ 2GHz clock frequency.

Figure 6 shows a comparison of the obtained accuracy for different odometry techniques based on the event-based camera, including our method. The velocity of the camera, entropy, and the number of generated events are shown as separate synced graphs to demonstrate how the behavior of the camera and environment can affect the number of generated events and accuracy. In Fig. 7, the response times of the considered event-based VO techniques are depicted.

As can be seen, our proposed method outperforms the other techniques by obtaining the best accuracy, i.e., the smallest error, and the shortest response time. The technique with a large ΔT time to accumulate the events to be processed in a frame loses accuracy whenever the speed of the camera increases. On the other hand, the technique with a small ΔT provides better results when the speed of the camera is high, but for the low camera speeds the error is quite high. A similar analysis can be applied to the techniques that are based on the number of events. In these cases, since the number of events is strongly affected

by the amount of information coming from the environment, together with the velocity of the camera, the effect of entropy can be clearly seen in the accuracy of the algorithms. The accuracy of the method with the fixed number of 10k events is not good, while for the technique with 100 events the accuracy is acceptable most of the time, decreasing only when entropy increases. Our proposed algorithm keeps the accuracy constantly high, because it is aware of the velocity of the camera and entropy of the environment simultaneously.

The other aspect to be discussed is the response time of the algorithm. As can be observed, the response times are quite high for the techniques with a small ΔT or $\#Events$. On the other hand, the techniques with a large ΔT or $\#Events$ can provide small response times, because their iterative processes take place in longer intervals, sacrificing the accuracy of the methods. The proposed technique, in turn, can keep the execution time at a reasonable level, making it suitable for different applications in which odometry is needed to be performed in real time.

5 Conclusion

In this paper, a hybrid technique was proposed to enable efficient odometry based on data captured by event-based cameras. The main contribution of the approach is its ability to flexibly change the number of events that are processed as a frame. To do this, we employed concepts of 3D projection and information theory to define a metric to dynamically determine the number of events that are considered for each frame. We used normal pixelated image data to extract the features in events and track those features in event-based frames. Experimental results show that the proposed hybrid method outperforms the traditional approaches that are based on considering either a fixed number of events per frame or a fixed time interval to accumulate the events. Our algorithm can operate efficiently and accurately in different environmental conditions and camera velocities.

References

1. Morevec, H.P.: Towards automatic visual obstacle avoidance. In: Proceedings of the 5th International Joint Conference on Artificial Intelligence, ser. IJCAI 1977, vol. 2, pp. 584–584 (1977)
2. Harris, C.G., Pike, J.M.: 3d positional integration from image sequences. In: Proceedings of Alvey Vision Conference, Cambridge, England (1987)
3. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment - a modern synthesis. In: Proceedings of the International Workshop on Vision Algorithms: Theory and Practice, ser. ICCV 1999, pp. 298–372 (2000)
4. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor. IEEE J. Solid-State Circ. **43**(2), 566–576 (2008)
5. Rebecq, H., Horstschaefer, T., Gallego, G., Scaramuzza, D.: EVO: a geometric approach to event-based 6-dof parallel tracking and mapping in real time. IEEE Robot. Autom. Lett. **2**(2), 593–600 (2017)

6. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(3), 379–423 (1948)
7. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006). https://doi.org/10.1007/11744023_34
8. Alzugaray, I., Chli, M.: Asynchronous corner detection and tracking for event cameras in real time. IEEE Robot. Autom. Lett. **3**(4), 3177–3184 (2018)
9. Mur-Artal, R., Montiel, J.M.M., Tardós, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. CoRR (2015)
10. Klein, G., Murray, D.: Parallel tracking and mapping on a camera phone. In: Proceedings of Eigth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2009), Orlando, October 2009
11. Ranftl, R., Vineet, V., Chen, Q., Koltun, V.: Dense monocular depth estimation in complex dynamic scenes. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4058–4066, June 2016
12. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: dense tracking and mapping in real-time. In: 2011 International Conference on Computer Vision, pp. 2320–2327, November 2011
13. Pizzoli, M., Forster, C., Scaramuzza, D.: REMODE: probabilistic, monocular dense reconstruction in real time. In: 2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, 31 May - 7 June 2014, pp. 2609–2616 (2014)
14. Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: 2013 IEEE International Conference on Computer Vision, pp. 1449–1456, December 2013
15. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: fast semi-direct monocular visual odometry. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 15–22 (2014)
16. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 611–625 (2018)
17. Cook, M., Gugelmann, L., Jug, F., Krautz, C., Steger, A.: Interacting maps for fast visual interpretation. In: The 2011 International Joint Conference on Neural Networks, pp. 770–776, July 2011
18. Weikersdorfer, D., Hoffmann, R., Conradt, J.: Simultaneous localization and mapping for event-based vision systems. In: Chen, M., Leibe, B., Neumann, B. (eds.) ICVS 2013. LNCS, vol. 7963, pp. 133–142. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39402-7_14
19. Weikersdorfer, D., Adrian, D.B., Cremers, D., Conradt, J.: Event-based 3d SLAM with a depth-augmented dynamic vision sensor. In: 2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, 31 May - 7 June 2014, pp. 359–364 (2014)
20. Kim, H., Handa, A., Benosman, R., Ieng, S.-H., Davison, A.: Simultaneous mosaicing and tracking with an event camera. In: Proceedings of the British Machine Vision Conference. BMVA Press (2014)
21. Kueng, B., Mueggler, E., Gallego, G., Scaramuzza, D.: Low-latency visual odometry using event-based feature tracks. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2016, Daejeon, South Korea, 9–14 October 2016, pp. 16–23 (2016)

22. Kim, H., Leutenegger, S., Davison, A.J.: Real-time 3d reconstruction and 6-dof tracking with an event camera. In: Computer Vision - ECCV 2016 - 14th European Conference, Proceedings, Part VI, Amsterdam, The Netherlands, 11–14 October 2016, pp. 349–364 (2016)
23. Thum, C.: Measurement of the entropy of an image with application to image focusing. Opt. Acta: Int. J. Opt. **31**(2), 203–211 (1984)
24. Grinberg, M.: Feature-based probabilistic data association for video-based multi-object tracking," Ph.D. dissertation, Karlsruhe Institute of Technology, Germany (2018)
25. Brandli, C., Berner, R., Yang, M., Liu, S., Delbruck, T.: A 240×180 130 db 3μs latency global shutter spatiotemporal vision sensor. IEEE J. Solid-State Circ. **49**(10), 2333–2341 (2014)
26. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI–8**(6), 679–698 (1986)
27. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
28. Mueggler, E., Rebecq, H., Gallego, G., Delbrück, T., Scaramuzza, D.: The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. Int. J. Robotics Res. **36**(2), 142–149 (2017)