Intelligent Computing pp 685-696

Mohamed S.A.S., Yasin J.N., Haghbayan MH., Heikkonen J., Tenhunen H., Plosila J. (2022) DBA-Filter: A Dynamic Background Activity Noise Filtering Algorithm for Event Cameras. In: Arai K. (eds) Intelligent Computing. Lecture Notes in Networks and Systems, vol 283. Springer, Cham. https://doi.org/10.1007/978-3-030-80119-9_44

https://doi.org/10.1007/978-3-030-80119-9_44

DBA-Filter: A Dynamic Background Activity Noise Filtering Algorithm for Event cameras

Sherif A.S. Mohamed[1], Jawad N. Yasin[1], Mohammad-Hashem Haghbayan[1], Jukka Heikkonen[1], Hannu Tenhunen[2], and Juha Plosila[1]

[1] University of Turku, 20500 Turku, Finland
[2] KTH Royal Institute of Technology, 11428 Stockholm

Abstract. Newly emerged dynamic vision sensors (DVS) offer a great potential over traditional sensors (e.g., CMOS) since they have a high temporal resolution in the order of μs, ultra-low power consumption and high dynamic range up to 140 dB compared to 60 dB in frame cameras. Unlike traditional cameras, the output of DVS cameras is a stream of events that encodes the location of the pixel, time, and polarity of the brightness change. An event is triggered when the change of brightness, i.e., log intensity, of a pixel exceeds a certain threshold. The output of event cameras often contains a significant amount of noise (outlier events) alongside the signal (inlier events). The main cause of that is transistor switch leakage and noise. This paper presents a dynamic background activity filtering, called DBA-filter, for event cameras based on an adaptation of the K-nearest neighbor (KNN) algorithm and the optical flow. Results show that the proposed algorithm is able to achieve a high signal to noise ratio up to 13.64 dB.

Keywords: event cameras, background filtering, KNN, dynamic, noise

1 Introduction

Novel silicon retina or "event cameras" are bio-inspired vision sensors that mimic the neural architecture of the human eyes. The output of event cameras is fundamentally different from traditional sensors (i.e., CMOS sensors). The output of traditional cameras is a sequence of images "absolute brightness" at a constant temporal rate (typically between 30 and 120Hz). Images can be either monochrome "gray-scale" images or images that have three channels per pixel, such images are called RGB images. On the other hand, event cameras measure per-pixel brightness changes (called "events") asynchronously. This results in a stream of events with a high dynamic range, low power consumption, and at a high temporal rate (up to 1MHz). Due to these characteristics, event cameras offer a great potential to solve classical as well as new computer vision problems [1]. Recently, they become commercially available in various models such as Dynamic Vision Sensors (DVS) [2], Dynamic and Active pixel Vision Sensor (DAVIS) [3], and Asynchronous Time-based Image Sensor (ATIS) [4].

Event cameras are similar to all other vision sensors they are noisy because of the inherent shot noise in photons and from transistor circuit noise, and they also have non-idealities [5]. To overcome this undesired effect, a sort of filtering block is required as a pre-processing block for event-based visual odometry (VO) algorithms. Alzugaray et. al [6] proposed a filtering method to remove redundant events caused by sudden and significant contrast change using a constant timestamp threshold $k = 50ms$. The mechanism of the proposed filter is that the incoming event is processed if it has a different polarity from the previous one at the same location. However, if it has the same polarity it is only processed if its timestamp (i.e., time of its occurrence) $t > t1 + k$, where $t1$ is the timestamp of the latest event triggered in the same location. In [7], the authors propose a filtering algorithm using Neural Network-Based Nearest Neighbor (NeuNN) to remove noise from data captured by an ATIS camera exploiting the neuromorphic architecture of the IBM TrueNorth processor [8]. The filter considers the activity of incoming events with its neighbors within the distance $D \leq \sqrt{2}$, i.e, the surrounding eight neighboring pixels. This activity, i.e., synaptic activity, is summed on to a neuron and the output of the activity is determined based on the configuration of the neuronal parameters. This summation of the synaptic, leak, and membrane voltage of the neuron is compared with a certain threshold. If it is greater than or equal to the threshold, then the neuron fires a spike. The main drawback of this filtering method is that the total number of required cores to perform the noise filtering increases rapidly with the size of the filter and the sensor, which makes real-time performance infeasible on embedded systems, i.e., resource-constrained systems.

In this paper, we propose a dynamic filtering algorithm that is able to achieve a high signal-to-noise ratio and achieve real-time performance on a Jetson TX2 system. The filter consists of two main stages: the timestamp (TS) filter and the BA filter. The TS-filter is based on optical flow and the timestamp of the previous event at the same location. The filter is able to discard noise, i.e., redundant events, caused by sudden movements and significant contrast changes. The BA-filter is based on an adaptation of the KNN algorithm and it is responsible to discard the BA noise caused by the hardware noise, i.e., transistor switch leakage. The proposed approach is able to significantly increase the signal-to-noise ratio and reduce the number of processed events captured by the event-based camera without the loss of relevant information.

The rest of this work is structured as follows. In Section 2, we illustrate the principle operation of event cameras and the cause of the noise. In Section 3, we explain the methodology of the proposed dynamic filtering. In Section 4, we present the experimental results. Finally, in Section 5, we draw a conclusion of the article.

2 Dynamic Vision Sensors

Recently, DVS (i.e, event-based) cameras offer a huge unprecedented raw performance characteristics and usability over traditional frame-based cameras. Many applications, such as localization and obstacle avoidance [9] [10] can exploit these powerful characteristics to achieve better real-time performance, power efficiency, and high accuracy. In [11], the authors proposed an algorithm to perform obstacle avoidance in low lighting conditions based event-camera. DVS cameras are particularly suitable for applications that require to perform localization or object tracking in challenging conditions such as high-speed maneuvers and low-light conditions.

2.1 Principle of Operation

A simplified circuit diagram of the DVS is shown in Figure 1. Each pixel in the sensor is composed of three main parts: *photoreceptor*, *differencer*, and *two comparators*. Pixels memorize the log intensity of their previous events and continuously monitors the changes in intensity. As illustrated in Figure 2, an event e_i is triggered at pixel $p_i = (x_i, y_i)$ and at timestamp ts_i when the change of intensity, i.e.,

$$\triangle \log I(p_i, ts_i) = \log I(p_i, ts_i) - \log I(p_i, ts_i - \triangle ts_i) \tag{1}$$

exceeds a certain threshold C, typically adjusted between 10%-50%, i.e.,

$$|\triangle \log I| > C \tag{2}$$

The Address Event Representation (AER) protocol [12] is used to generate the information for each event (quadruplets), i.e.,

$$e_i = (x_i, y_i, ts_i, pol_i) \tag{3}$$

where x_i and y_i denote the position of the pixel, i.e., the column and row of the sensor, respectively. The timestamp of the event is denoted by ts_i and polarity pol_i {+1,-1} is the sign of the intensity change.

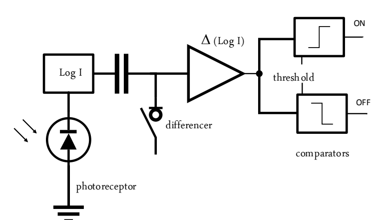


Fig. 1: Simplified circuit diagram of the DVS pixel

3 Filtering

To reduce the amount of redundant events and increase the signal to noise ratio, we propose a filtering method (called DBA-filter) that is composed of two stages: the timestamp-filter (TS-filter) and the background active filter (BA-filter) (Fig.4).

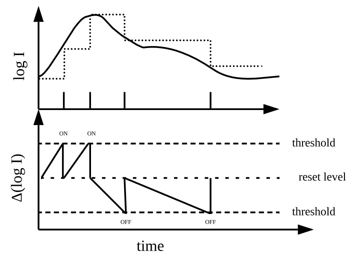


Fig. 2: Schematic of the operation of a DVS pixel, converting brightness changes into events

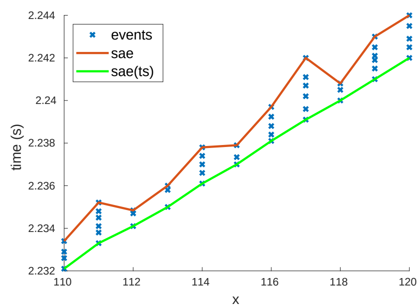


Fig. 3: The timestamp of the events triggered due to the translation of a moving dot. The (red) line represents the naive SAE, i.e., w/o filtering which stores the latest timestamp. On the other hand, our proposed TF-filter discards redundant events and the SAE_{ts} (green) stores only the timestamp of unique events which represent the time when the visual stimulus accrued.

3.1 TS-filter

According to Eq. 2, a sudden movement or a significant contrast change would trigger multiple events at the same pixel in a short time window. Processing those duplicate events increases the computational cost unnecessarily. Moreover, it affects the accuracy since the timestamp stored in the Surface Active Event (SAE) does not represent the time when the visual stimulus was accrued, as illustrated in Figure 3. SAE is a common way which uses the spatio-temporal domain. It can be considered as an elevation map, which is composed of the two-dimensional sensor frame (x, y) and one dimension to represent the time.

The first stage of the proposed algorithm, i.e., TS-filter, is illustrated in Algorithm 1. We use a global SAE (G-SAE) to store the information for filtered events. The G-SAE has the same dimension of sensors which stores the timestamp of events (32-bit) and the polarity of events (1-bit). For instance, using a DVS with 240x180 pixels will result in a G-SAE of size 175 KB. The filter checks the polarity of the previous event at the same location in G-SAE. If the polarity

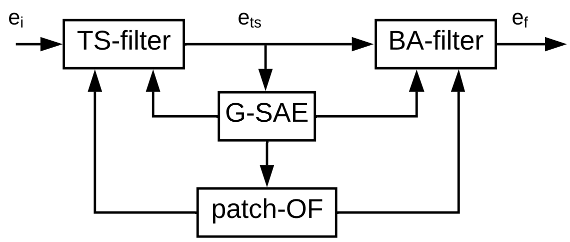


Fig. 4: An overall system architecture of the proposed filtering

Algorithm 1: Dynamic TS-filter

Result: e_{ts}
initialization;
if $pol_i \neq pol_j$ **then**
    Process event;
    Update G-SAE;
else
    Select batch $e_i \in P_N$;
    Obtain batch $\leftarrow OF_{P_N}$;
    Calculate $\rightarrow t^r$;
    if $ts_i \leq ts_j + t^r$ **then**
        Discard event;
    else
        Process event;
        Update G-SAE;
    end
end

of the incoming event differs from the polarity of the previous event in the same location, the incoming event is processed. Otherwise, events triggered within time-window t^r are discarded and only events with timestamp bigger than the threshold are processed, i.e.,

$$ts_i > ts_j + ts_r \tag{4}$$

where the timestamp of the incoming event is denoted by ts_i. The timestamp of the previous event at the same location is denoted by ts_j, and ts_r denotes the timestamp threshold. The optical follow (OF) of each batch of size 15x15 is computed using local planar [13]. A suitable timestamp threshold is obtained for each batch, i.e.,

$$t_i^r = \frac{A}{OF_{P_k} + B}; \qquad e_i \in P_k \tag{5}$$

3.2 BA-filter

Background activity (BA) events are triggered without any brightness or movement changes, however, due to hardware limitations. The main causes of BA noise are thermal noise and current leakage of switches [2]. This noise degrades the quality of captured data and increases the computational costs unnecessarily. Hence, a filtering algorithm to eliminate BA noise is necessary to obtain a high-performance localization. The main characteristic of BA events is that they lack correlation with events in their spatiotemporal neighborhood, i.e.,

$$corr(e_1, e_2) = \begin{cases} |x_1 - x_2| \leq dP \\ |y_1 - y_2| \leq dP \\ |t_1 - t_2| \leq dT \end{cases} \tag{6}$$

The spatiotemporal correlation of two events $e_1(x_1, y_1, t_1)$ and $e_2(x_2, y_2, t_2)$ is denoted by $corr(e_1, e_2)$ as illustrated in Equation 6. The spatial window size is denoted by dP and the time size is denoted by dT.

The probability of observing BA events on a pixel on the DVS sensor can be considered as a Poisson distribution [14], i.e.,

$$P(e_i = w) = \frac{(\lambda t)^w e^{-\lambda t}}{w} \qquad ; \lambda > 0 \tag{7}$$

where w is the number of BA events triggered in the time interval t and λ is the BA rate.

In [14–16], the authors have used a fixed size window to filter the BA events. Based on the movement speed of the camera or the object in the scene, two or more BA events might be close enough in the spatiotemporal neighborhood. Hence, using a fixed spatiotemporal neighborhood, see Figure 5, would pass BA

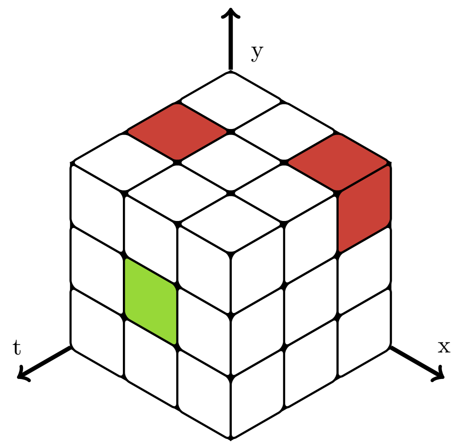


Fig. 5: Principal of spatiotemporal neighbourhood of size (3x3x3). Green block indicates the incoming event and red blocks denotes the neighboring events.

noise or filter real events in some cases. Therefore, we propose a second stage filtering, i.e., a dynamic BA filtering algorithm based on an adaptive spatiotemporal neighborhood window and KNN to filter the BA noise. The size of the neighborhood window is determined based on the optical flow of the batch. The K parameter of the KNN algorithm is obtained based on the entropy of the batch, i.e., number of events in the last t time interval. The incoming event will pass if it has neighbors more or equal to K threshold within its neighborhood window.

The proposed algorithm is summarized in Algorithm 2. The sensor is subsampled into N number of batches, each batch has a size of 15x15 pixels, which is used to calculate the optical flow (OF). The OF is computed using the RANSAC algorithm by fitting a local planar [13]. The size of the spatiotemporal neighborhood (STN) window is an odd number and varies from 3x3 to 15x15. An incoming event in a high-speed batch will have a size of 3x3, on the other hand, events fall in slow batches will STN of size 15x15. The entropy of each batch is computed by calculating the number of events that occur within a t time interval, which is used to determine the K parameter. If the number of neighboring events with STN of the incoming event is less than K the event is eliminated, otherwise it passes the filter.

4 Results

4.1 Experimental Setup

We evaluated the proposed filtering algorithm by running the algorithm on publicly available datasets of [17], [18]. We carefully selected a number of subsets to ensure a fair and comprehensive evaluation scheme. The selected subsets are

Algorithm 2: Dynamic BA-filter

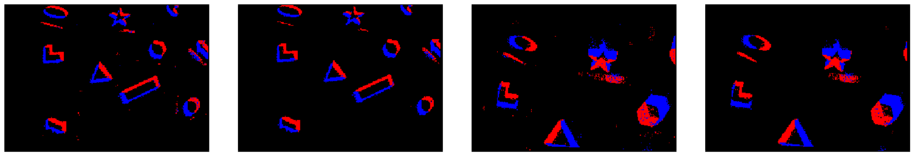
Result: e_f
subsampling $\rightarrow P_N$;
Calculate OF $\rightarrow OF_{P_N}$;
Calculate Entropy $\rightarrow E_{P_N}$;
while e_{ts} **do**
  Select batch $e_{ts} \in P_N$;
  Obtain batch $\leftarrow OF_{P_N}$;
  Compute STN(x,y,t);
  Obtain batch $\leftarrow E_{P_N}$;
  Compute K;
  if $neighbours \in SN < K$ **then**
    Discard event;
  else
    Pass event;
  end
end

composed of simple and complex scenes, including low and high textured environments, and slow and fast motions. In a slow-motion scenario, the DAVIS camera generates up to 3×10^5 events per second. On the other hand, in rapid camera/object movements, it can generate up to 3 million events per second. The datasets are recorded by a Dynamic and Active-pixel Vision Sensor DAVIS-240, which contain many sequences of frame-based, i.e., intensity images, and asynchronous events at the resolution of 240x180. Note that the intensity images are only used to obtain the ground-truth for evaluation purposes. The proposed algorithm has been implemented in software in C++. The application was run on an Nvidia Jetson TX2 board with quad-core ARM Cortex-A57 CPU @ 2GHz clock frequency.

4.2 Qualitative Evaluation

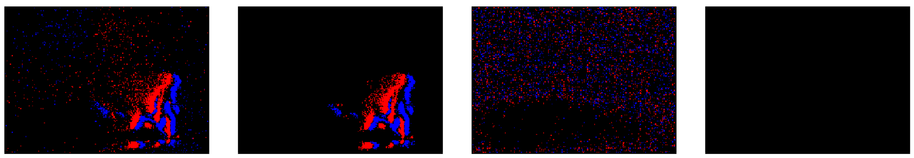
Figure 6 shows a qualitative comparison of the video output of the DVS camera without using any filter and using our proposed filter. The dataset (i.e., *shapes_6dof*) used in this evaluation is recorded indoors using a DVS camera moving freely in 6-DOF with different motion speeds. The shapes appear in the video are 2D pictures mounted on a wall. The results show that the proposed filter is able to cope with both slow-motion and high-speed camera motion to remove background noise and filter redundant events generated from the sudden movement, significant contrast changes, and hardware limitations.

In order to comprehensively evaluate the proposed algorithm, we have tested it on the *night_run* dataset. This dataset is recorded outdoors by a static camera, while a human is running in front of the camera. Figure 7, show that the proposed filter is able to filter noise not only in static scenes but also in a dynamic environment without losing valuable information.



(a) slow motion w/o filter (b) slow motion w/ filter (c) fast motion w/o filter (d) fast motion w/ filter

Fig. 6: Shapes Sequence. Screenshots of the DVS video output at various time instances.



(a) moving object w/o filter (b) moving object w/ filter (c) no motion w/o filter (d) no motion w/ filter

Fig. 7: Night run sequence. Screenshots of the DVS video output at various time instances.

4.3 Quantitative Evaluation

In order to quantify the noise filtering performance using the algorithms detailed in Section 3 and their implementation on Jetson TX2 described in Section 4.1, we first define the following terms and the metrics below to compute them on the unfiltered and filtered data, see Table 1.

Abbr.	Definition
$\mathbf{R_{w/o}}$	Total no. of real events without filtering
$\mathbf{N_{w/o}}$	Total no. of noise events without filtering
$\mathbf{R_f}$	Total no. of real events with filtering
$\mathbf{N_f}$	Total no. of noise events with filtering
$\mathbf{P_{RR}}$	Per. % of remaining real events w/ filtering
$\mathbf{P_{NR}}$	Per. % of remaining noise events w/ filtering
$\mathbf{RNR_{w/o}}$	Real to noise ratio without filtering
$\mathbf{RNR_f}$	Real to noise ratio with filtering

Table 1: Quantitative metric

The results of the quantitative evaluation are summarized in Table 2. Two different datasets, i.e., shapes_6dof and nigh_run, were used for the evaluation. The results show that the proposed algorithm is capable of filtering noise caused by sudden movement and background activity effectively. The filter is able to maintain an RNR, i.e., signal-to-noise ratio (SNR), up to 13.64 dB in a static scene, and up to 6.709 dB in a high dynamics scene.

	shapes_6dof	night_run
$\mathbf{R_{w/o}}$	10.31×10^6	0.34×10^6
$\mathbf{N_{w/o}}$	7.62×10^6	0.455×10^6
$\mathbf{R_f}$	9.73×10^6	0.3×10^6
$\mathbf{N_f}$	0.42×10^6	0.064×10^6
$\mathbf{P_{RR}}$	94.37%	88.2%
$\mathbf{P_{NR}}$	5.24%	14.06%
$\mathbf{RNR_{w/o}}$	1.31 dB	-1.27%
$\mathbf{RNR_f}$	13.64 dB	6.709 dB

Table 2: The quantitative performance of proposed algorithm on dataset.

5 Conclusion

We have presented DBA-filter, adaptive filtering method that redundant events and noise based on asynchronous events. There are two main sources of noise such as sudden and significant contrast changes and background activities. Our method is composed of two-stage 1) The first stage, i.e., TF-filter is to discard redundant events using dynamic timestamp techniques. The timestamp is computed based on optical flow for each batch 2) the second stage, i.e., BA-filter is designed to discard background events based on an adaptive KNN algorithm. We demonstrated that our approach facilitates execution under different scenarios. Our method is able to discard unnecessary events without loss of relevant information.

Acknowledgment

This work was supported by the Academy of Finland under the project (314048).

References

1. Mohamed, S.A.S., Haghbayan, M., Westerlund, T., Heikkonen, J., Tenhunen, H., Plosila, J.: A survey on odometry for autonomous navigation systems. IEEE Access pp. 97466–97486 (2019)
2. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor. IEEE Journal of Solid-State Circuits 43(2), 566–576 (Feb 2008)
3. Brandli, C., Berner, R., Yang, M., Liu, S., Delbruck, T.: A 240 x 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. IEEE Journal of Solid-State Circuits 49(10), 2333–2341 (Oct 2014)

4. Posch, C., Matolin, D., Wohlgenannt, R.: A QVGA 143 db dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. J. Solid-State Circuits 46(1), 259–275 (2011)
5. Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A.J., Conradt, J., Daniilidis, K., Scaramuzza, D.: Event-based vision: A survey. CoRR abs/1904.08405 (2019)
6. Alzugaray, I., Chli, M.: Asynchronous corner detection and tracking for event cameras in real time. IEEE Robotics and Automation Letters 3(4), 3177–3184 (Oct 2018)
7. Padala, V., Basu, A., Orchard, G.: A noise filtering algorithm for event-based asynchronous change detection image sensors on truenorth and its implementation on truenorth. Frontiers in Neuroscience 12, 118 (2018)
8. Merolla, P.A., Arthur, J.V., Alvarez-Icaza, R., Cassidy, A.S., Sawada, J., Akopyan, F., Jackson, B.L., Imam, N., Guo, C., Nakamura, Y., Brezzo, B., Vo, I., Esser, S.K., Appuswamy, R., Taba, B., Amir, A., Flickner, M.D., Risk, W.P., Manohar, R., Modha, D.S.: A million spiking-neuron integrated circuit with a scalable communication network and interface. Science 345(6197), 668–673 (2014)
9. Yasin, J.N., Mohamed, S.A.S., Haghbayan, M.H., Heikkonen, J., Tenhunen, H., Yasin, M.M., Plosila, J.: Energy-efficient formation morphing for collision avoidance in a swarm of drones. IEEE Access pp. 1–1 (2020)
10. Yasin, J.N., Mohamed, S.A.S., Haghbayan, M., Heikkonen, J., Tenhunen, H., Plosila, J.: Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches. IEEE Access 8, 105139–105155 (2020)
11. Yasin, J.N., Mohamed, S.A.S., Haghbayan, M.H., Heikkonen, J., Tenhunen, H., Yasin, M.M., Plosila, J.: Night vision obstacle detection and avoidance based on bio-inspired vision sensors. In: 2020 IEEE SENSORS. pp. 1–4 (2020)
12. Mahowald, M.: An Analog VLSI System for Stereoscopic Vision. Kluwer Academic Publishers, USA (1994)
13. Benosman, R., Clercq, C., Lagorce, X., Ieng, S., Bartolozzi, C.: Event-based visual flow. IEEE Trans. Neural Networks Learn. Syst. 25(2), 407–417 (2014), https://doi.org/10.1109/TNNLS.2013.2273537
14. Khodamoradi, A., Kastner, R.: O(n)-space spatiotemporal filter for reducing noise in neuromorphic vision sensors. IEEE Transactions on Emerging Topics in Computing pp. 1–1 (2017)
15. Delbruck, T.: Frame-free dynamic digital vision. Proceedings of the International Symposium on Secure-Life Electronics, Advanced Electronics for Quality Life and Society (03 2008)
16. Liu, H., Brandli, C., Li, C., Liu, S., Delbruck, T.: Design of a spatiotemporal correlation filter for event-based sensors. In: 2015 IEEE International Symposium on Circuits and Systems (ISCAS). pp. 722–725 (2015)
17. 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. I. J. Robotics Res. 36(2), 142–149 (2017)
18. Scheerlinck, C., Barnes, N., Mahony, R.: Continuous-time intensity estimation using event cameras. In: Asian Conf. Comput. Vis. (ACCV) (December 2018)