



Events-to-Frame: Bringing Visual Tracking Algorithm to Event Cameras

Sixian Chan¹, Qianqian Liu¹, XiaoLong Zhou^{1,2}, Cong Bai¹, and Nan Chen³(✉)

¹ College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

² College of Electrical and Information Engineering, Quzhou University, Quzhou 324000, China

³ College of Continuing Education, Qilu Normal University, Jinan 250013, China
20153410@qlnu.edu.cn

Abstract. Event based cameras mean a significant shift to standard cameras by mimicking the work of the biological retina. Unlike the traditional cameras which output the image directly, they provide the relevant information asynchronously through the light intensity changes. This can produce a series of events that include the time, position, and polarity. Visual tracking based on event camera is a new research topic. In this paper, by accumulating a fixed number of events, the output of events stream by the event camera is transformed into the image representation. And it is applied to the tracking algorithm of the ordinary camera. In addition, the data sets of the ground-truth is relabeled and with the visual attributes such as noise events, occlusion, deformation and so on so that it can facilitate the evaluation of the tracker. The data sets are tested in the existing tracking algorithms. Extensive experiments have proved that the data sets created is reasonable and effective. And it can achieve fast and efficient target tracking through the SOTA tracking algorithm test.

Keywords: Dynamic vision sensor · Visual tracking · Correlation filters · Deep learning

1 Introduction

Event-based cameras are driven by the events which occur in a scene like their biological counterparts. They are different from conventional vision sensors, on the contrary, they are driven by the timing and control signals by man-made, and that have nothing to do with the source of visual information [18]. Dynamic Vision Sensor (DVS) [27] (Fig. 1a) is a kind of the event based, it provide a series of asynchronous events [9] (Fig. 1b). And these bio-inspired sensors overcome some of the limitations of traditional cameras: high temporal resolution, high dynamic range, low power consumption and so on. Hence, event cameras can take an advantage for high-speed and high dynamic range visual applications

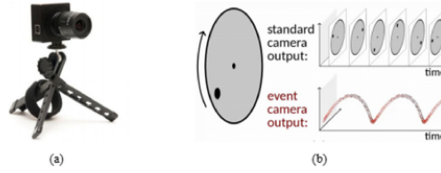


Fig. 1. (a) The Dynamic Vision Sensor (DVS). (b) The difference between event camera with traditional camera when a blackspot move on the platform.

in challenging scenarios with large brightness contrast. Since these advantages, vision algorithms based on event cameras have been applied in the areas of like event-based tracking, Simultaneous Localization and Mapping (SLAM), and object recognition [26, 31, 32, 35], etc.

Visual tracking is a hot topics and it is widely used in video surveillance, unmanned driving, human-computer interaction and so on. While tracking algorithms are well established and have achieved successful applications in many aspects. Their visual image acquisition methods are based on traditional fixed-frequency acquisition frames which suffer from high redundancy, high latency and high data volume. Event-based sensors provide a continuous steam of asynchronous events. The position, time and polarity of the each event is encoded by address event representation (AER) [19]. The AER is triggered by the event. The pixels work asynchronously and only outputs the address and information of the pixel whose light intensity changes. Instead of passively reading out each pixel information in the frame, they can eliminate redundant information from the source. Real-time dynamic response with scene change, image super-sparse representation, and the events of output asynchronously can be widely used in high-speed object tracking and robot vision.

The current visual tracking algorithm works on nature images. Since each pixel of the frame needs a uniform time for exposure, it will cause image blur and information loss when the object moves quickly. And the tracking algorithms are susceptible to lighting, fast movement of targets, etc. Event-based tracking maybe solve this problem. Event-based cameras cannot directly output the frames. Therefore, it cannot be directly applied to the computer vision algorithm of ordinary cameras.

In our work, we convert the event stream generated by the event camera into image representations. The converted image is formed by integrating a certain mount of events with a sliding event window. We select seven event data records from the DVS benchmark data sets [14]. By accumulating the DVS data into the frames, we have remarked the ground-truth to further accurately determine the locations of tracking object for the current tracking algorithms evaluation. All of the frames are annotated with axis-aligned bounding boxes, the sequences are relabeled with the visual attributes such as noise events, occlusion, deformation and so on. The experiments test and verify the reasonable validity of the labeled data and show that the tracking algorithms can track the specific targets with

high accuracy and robustness in complex scenes based on the output frames of the event camera.

2 Related Work

Many methods for the event-based tracking have been presented up to now. Because of the low data processing and latency of the event based cameras, early researchers track targets which moving in a static scene as the clusters of events, and they achieve good performances in applications such as traffic monitoring, high-speed robot tracking and so on [7]. At the same time, the event-by-event adaptive tracking algorithm has been proved in some high contrast user-defined shapes. Ni et al. [25] proposed the nearest neighbor strategy, which linked the incoming event with the target form and updated its conversion parameters. Glover et al. [10] proposed an improved particle filter which can automatically adjust the time window of the target observation for tracking a single target in event-space. All of the above methods need a experience premise or user-defined to define the target to be tracked. When the motion range of the object is gradually enlarged, other methods determine to distinct the natural features to track by analyzing the event [8]. Zhu et al. [36] proposed a soft data association modeled with probabilities, which relying on grouping events into a model. Features were generated by the motion compensated events, which generated to pointsets based on registered templates of new events. Lagorce et al. [18] proposed an event-based multi kernel algorithm, which tracked the characteristics of incoming events by integrating various kernels like Gaussian and user-defined kernels and so on. The appearance features of the event stream objects were obtained from a multi-scale space independent of the data foundation, and the original features could not be retained. Kogler et al. [17] presented an event-to frame converter and tested on two conventional stereo vision algorithms. Schraml et al. [30] proposed to integrate DVS events in a period of 5–50 ms and used them to track moving objects in stereo vision. However, one difference between an event camera and a normal camera is that the stationary object is not imaged, which result in the sparse data on the space and time. When integrating events at a fixed time, the time information is destroyed and the spatial data is sparse in different frames. Li et al. [20] proposed a tracking algorithm based on the CF mechanism by encoding the event-stream object by the rate coding. But it produces a lot of noise events.

3 Event-Image Representation Based on Event Time-Stamp

Event based cameras have independent pixels and response to changes in logarithm of light intensity $e_m = (X_m, t_m, p_m)$. In the ideal case of no noise, the event is triggered by the address-event representation combines the position, time, and polarity of the event (a signal with a change in brightness, an ON

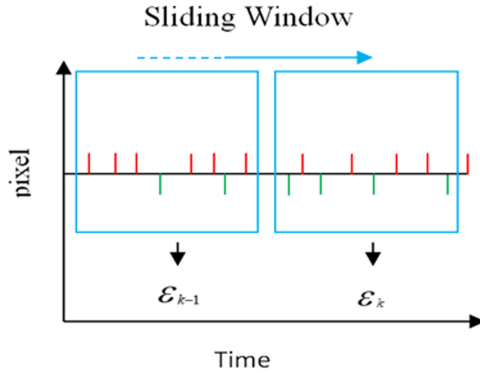


Fig. 2. The sliding event window diagram. The incoming event stream is depicted as red (positive events)/green (negative events) in the timeline). Events are divided into N events windows (blue box) by sliding windows. Each window is formed into each frame. In this example, $N = 8$. (Color figure online)

event is a positive event indicating an increase in brightness, and an OFF event is a negative event indicating a decrease in brightness). The event as:

$$e_m = (X_m, t_m, p_m) \quad (1)$$

Where: $X_m = (x_m, y_m)^T$ indicates the pixel address; t_m indicates the time when the event occurs; $p_m \in \{+1, -1\}$ indicates the polarity of event, $p_m = +1$ indicates the brightening event, otherwise it becomes dark. The triggered event means that the brightness increase from the last event reaches a preset threshold $\pm C$, namely:

$$\Delta L(X_m, t_m) = p_m C \quad (2)$$

where:

$$\Delta L(X_m, t_m) = L(X_m, t_m) - L(X_m, t_m - \Delta t_m) \quad (3)$$

Δt_m indicates the time elapsed since the last trigger event of pixel's X_m .

As mentioned above, the event camera outputs the captured image as an asynchronous event stream. Since a single data stream contains a lot of data, it should be processed in batches firstly. In this paper, a sliding event window [28, 29] with a fixed number of N events is used to slide on the data stream, thereby completing the transition from an event stream which containing huge amounts of data to a small batch of data with a fixed number of events. The sliding window divides the event stream into multiple small windows, and controls the flow of one small window every time. This method can overcome the large of traffic data. The sliding window works as shown in Fig. 2.

The valid information of the event stream rely on the number of events are processed at the same time. There are generally two methods for event processing: One is based on event-driven that work on event by event, and the other method that work on a set of events. The former process the every of

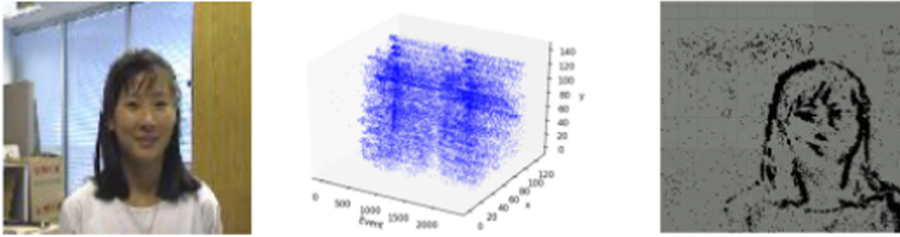


Fig. 3. An example of a girl scene that accumulates a certain number of events. The (a) shows a girl captured by a normal camera. The (b) shows the visualization of the space-time information within the stream of events of the girl. The (c) shows the girl image of corresponding Integral reconstruction.

incoming single event. However, an independent event does not provide enough information frequently and may generate a large number of noise events. The later method integrate all of the information contained in the events. we choose the fixed events information to integrate the event stream, which can achieve better results. We define $E(t)$ as the sum of events during a little time interval of the sliding window:

$$E(t) = \sum_{t \in \text{eventwindow}} (\text{event}_x \pm 1, \text{event}_y \pm 1, t) \tag{4}$$

We accumulate 7500 events into the frames in this paper. By the process of accumulating framing, the position where the event occurs is converted to the pixel position and formed the image frame. The position of the event in the world coordinates is mapped to the image coordinates by the mapping relationship by the Eq. 5:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 1/z_c \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{5}$$

Fixed events can make the deviation range between two adjacent frames not too big.

To reduce the effect of the image surface’s noise, we add an event counter synchronously during accumulating the events. At the same time, we record the every pixel’s coordinate position (x, y) and times in the event. The more events accumulate, the higher weight of events, and the higher of the activity in the process of data accumulating and framing. It is easier to get strongly vitality in the event selection and it is not easy to eliminate. On the contrary, the fewer the number of event accumulations, the easier to obtain a higher elimination rate in the event selection, and it is easy to die. Compared to the time surface [23,34], the ability to handle the local patch is flat (weaker), but the speed of processing is faster and the resource of consumption is less.

The accumulated events are represented by image binarization, and the pixel’s gray value is set to 0 or 255. Hence, the entire image exhibit a distinct between black and white effect. Since the size of the data sets provided by DVS benchmark data sets is different, we divide them into 200*200 pixels, and adjust the object to the middle of the lens to obtain effective information. The way of processing can improve the robustness of the input data. The Fig. 3 shows a scene by accumulating a fixed number of events.

Table 1. The distribution of the video sequence attributes and the length of the video recording table, 1 means the attribute, 0 means no.

Sequence	Long(s)	Noise events	Complicated background	Occlusion	Deformation	Scale variation
singer	13	1	1	1	1	1
figure_skating	34	1	1	1	1	1
girl	36	1	0	1	1	1
Sylvester	21	1	0	0	1	1
Vid_J_person_floor	21	1	1	1	1	1
Vid_E_person_part_occluded	8	1	1	1	0	0
Vid_D_person	22	1	1	0	0	0

4 Experiment and Evaluation

4.1 The DVS Data Sets

We choose seven event data records from the DVS benchmark which recorded by the DVS output of a DAVIS camera [14]. The tracking targets including the person, head, doll. These seven sequences are “figure_skating”, “singer”, “girl”, “sylvester”, “Vid_D_person”, “Vid_E_person_part_occluded”, “Vid_J_person_floor”, which are accumulated into 696, 263, 731, 437, 443, 160 and 422 frames, respectively.

The ground-truth provided by the tracking data sets is slightly offset from the actual position and size. Therefore, we relabel the locations of tracking object in the seven sequences. Each frame is annotated with an axis aligned bounding box. The description of boxes is (x, y, w, h) , where (x, y) is the left-top corner vertex position, and w, h represent the width and height. Moreover, all of the sequences are labeled with five visual attributes, including the noise events, occlusion, complicated background, scale variation and deformation. The distribution of these attributes and the long of recordings are presented in Table 1.

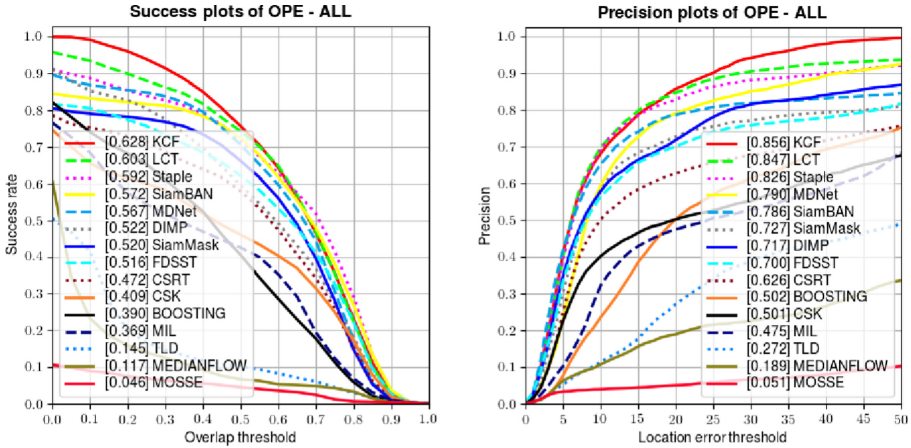


Fig. 4. The performance comparison of the tracking algorithms. The success and accuracy rate are shown in the left and right figures respectively.

4.2 Evaluation

The evaluation methodology: The success rate and accuracy are used to the tracking algorithms for evaluating. The former is measured by average overlap rate (AOR), that is, the percentage of sequence frames whose overlap exceeds the set threshold. And the latter is measured by the center location error (CLE), that is, the distance between the target and actual location is less than the percentage of the sequence frame with the set threshold.

The evaluation tracking algorithms: According to the learning mechanism of algorithms, we divide them into the tracking algorithms based on deep learning (SiamBAN [5], MDNet [24], DIMP [3], SiamMask [33]), the Correlation filtering tracking algorithms with hand-crafted features (CSK [12], KCF [13], FDSST [6], LCT [22], Staple [2], CSRT [21]), and other tracking algorithms like (CT [15], BOOSTING [11], MIL [1], TLD [16], MOSSE [4]).

Quantitative Analysis. It is found through the experiments that the visual tracking algorithms can effectively track our event stream sequences and achieve good performance. Among them, the KCF tracking algorithm performs best. The comparison results of accuracy and success rate plot are showed in Fig. 4. From the Fig. 4 we can see that the SOTA deep learning algorithms perform not well. The improved feature extraction algorithm (such as KCF with hog feature) can track accurately and effectively. The pre-20 and suc-50 are 0.856 and 0.628, respectively. It performs better than the CSK algorithm that use a cyclic matrix based on gray features, which is beyond doubt. But its accuracy is better than that the staple tracking algorithm that combine the Hog and color histogram Fortunately, this may be related to our video sequence with the binary grayscale images. In addition, the effect of tracking drift is easily caused

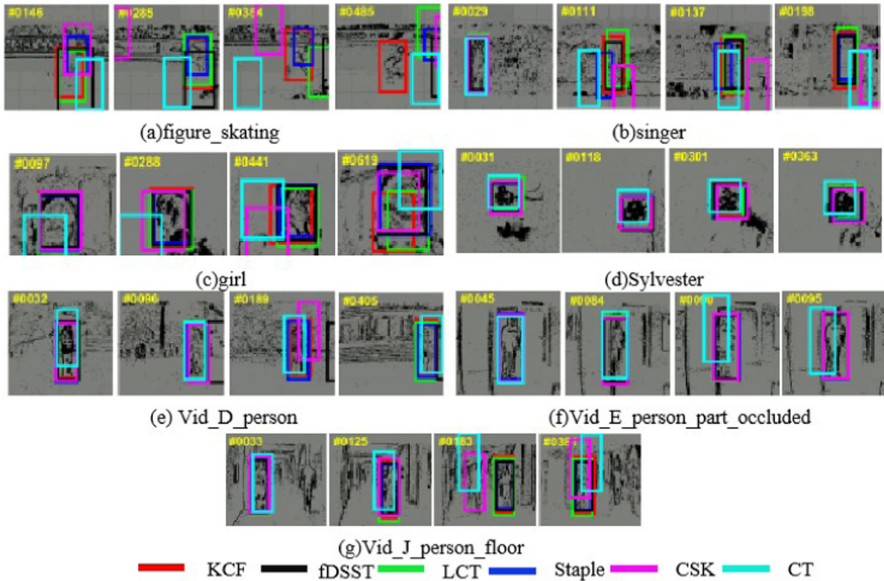


Fig. 5. Examples of the results of some tracking algorithms in seven event stream sequences.

by scale change, the FDSST algorithm introduce to the scale estimation, and the LCT algorithm add a confidence filter to the scale is slightly better than its performance, but they are worse than the KCF tracking algorithm. In summary, the data transformed by event camera is valid and reasonable.

Quantitative Analysis. The tracking results of some trackers in the seven event stream sequences are shown in Fig. 5. The scale of the skater change repeatedly for the figure_skating sequence. At the beginning, all of the six tracking algorithms are able to track the object, but when it continue to change, the tracker will drift gradually. At the 485th frame, the other five tracking algorithms completely lose the target, and the KCF can still keep tracking. In the singer sequence, under the disturbance of the background, the singer is blurred. Due to the sudden change of the stage (the 111th frame and the 131th frame), the target is lost for the CT and CSK tracking algorithm. Similarly, in the Vid_D_person and Vid_E_person_part_occluded sequence, the CSK and CT also drift. However, for the Sylvester sequence, even if the target is deformed, all the trackers can still track robustly. In the girl sequence, The face target is constantly deformed and changed, and the tracker gradually drift. At the 619th frame, only Staple and FDSST can keep effective tracking. In the Vid_J_person_floor sequence, When the target is interfered by another person, the CT and CSK will track the wrong target, which makes the tracking fail. But the rest of the tracking algorithm can

still achieve stable tracking when the two are separated (via. the 125th frame and 183th frame).

5 Conclusion

We accumulated the events stream into frames by integrating a certain number of asynchronous events. It applied to the SOTA tracking algorithm of the ordinary camera successfully, and achieved good performance in complex visual scenes. The experiments showed the rationality of converting the event stream into a frame image. At the same time, the processing of the method not only avoided the effects of lighting, but also reduced the effects of the background, which could protect the privacy outside the target. In the future, we will put forward a novel target tracking algorithm with high robustness for the data sets made in this paper.

Acknowledgment. This work is supported by National Natural Science Foundation of China under Grant No. 61906168 and Zhejiang Provincial Natural Science Foundation of China under Grant No. LY18F020032.

References

1. Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(8), 1619–1632 (2010)
2. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: complementary learners for real-time tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1401–1409 (2016)
3. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6182–6191 (2019)
4. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550. IEEE (2010)
5. Chen, Z., Zhong, B., Li, G., Zhang, S., Ji, R.: Siamese box adaptive network for visual tracking. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6668–6677 (2020)
6. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Discriminative scale space tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(8), 1561–1575 (2016)
7. Drazen, D., Lichtsteiner, P., Häfliger, P., Delbrück, T., Jensen, A.: Toward real-time particle tracking using an event-based dynamic vision sensor. *Exp. Fluids* **51**(5), 1465 (2011). <https://doi.org/10.1007/s00348-011-1207-y>
8. Gallego, G., et al.: Event-based vision: a survey. *arXiv preprint arXiv:1904.08405* (2019)
9. Gehrig, D., Rebecq, H., Gallego, G., Scaramuzza, D.: Asynchronous, photometric feature tracking using events and frames. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *ECCV 2018. LNCS*, vol. 11216, pp. 766–781. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01258-8_46

10. Glover, A., Bartolozzi, C.: Robust visual tracking with a freely-moving event camera. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3769–3776. IEEE (2017)
11. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88682-2_19
12. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7575, pp. 702–715. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_50
13. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(3), 583–596 (2014)
14. Hu, Y., Liu, H., Pfeiffer, M., Delbruck, T.: DVS benchmark datasets for object tracking, action recognition, and object recognition. *Front. Neurosci.* **10**, 405 (2016)
15. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33712-3_62
16. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(7), 1409–1422 (2011)
17. Kogler, J., Sulzbachner, C., Kubinger, W.: Bio-inspired stereo vision system with silicon retina imagers. In: Fritz, M., Schiele, B., Piater, J.H. (eds.) ICVS 2009. LNCS, vol. 5815, pp. 174–183. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04667-4_18
18. Lagorce, X., Meyer, C., Ieng, S.H., Filliat, D., Benosman, R.: Asynchronous event-based multikernel algorithm for high-speed visual features tracking. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(8), 1710–1720 (2014)
19. Lazzaro, J., Wawrzynek, J.: A multi-sender asynchronous extension to the AER protocol. In: Proceedings Sixteenth Conference on Advanced Research in VLSI, pp. 158–169. IEEE (1995)
20. Li, H., Shi, L.: Robust event-based object tracking combining correlation filter and CNN representation. *Front. Neurobot.* **13**, 82 (2019)
21. LuNežič, A., Vojtř, T., Čehovin Zajc, L., Matas, J., Kristan, M.: Discriminative correlation filter tracker with channel and spatial reliability. *Int. J. Comput. Vision* **126**(7), 671–688 (2018)
22. Ma, C., Yang, X., Zhang, C., Yang, M.H.: Long-term correlation tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5388–5396 (2015)
23. Manderscheid, J., Sironi, A., Bourdis, N., Migliore, D., Lepetit, V.: Speed invariant time surface for learning to detect corner points with event-based cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 10245–10254 (2019)
24. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4293–4302 (2016)
25. Ni, Z., Pacoret, C., Benosman, R., Ieng, S., RÉGNIER*, S.: Asynchronous event-based high speed vision for microparticle tracking. *J. Microsc.* **245**(3), 236–244 (2012)

26. Paredes-Vallés, F., Scheper, K.Y.W., De Croon, G.C.H.E.: Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: from events to global motion perception. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**, 2051–2061 (2019)
27. Patrick, L., Posch, C., Delbruck, T.: A 128x 128 120 db 15 μ s latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits* **43**, 566–576 (2008)
28. Ramesh, B., Zhang, S., Lee, Z.W., Gao, Z., Orchard, G., Xiang, C.: Long-term object tracking with a moving event camera. In: *BMVC*, p. 241 (2018)
29. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: Events-to-video: bringing modern computer vision to event cameras. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3857–3866 (2019)
30. Schraml, S., Belbachir, A.N., Milosevic, N., Schön, P.: Dynamic stereo vision system for real-time tracking. In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 1409–1412. IEEE (2010)
31. Seok, H., Lim, J.: Robust feature tracking in DVS event stream using Bézier mapping. In: *The IEEE Winter Conference on Applications of Computer Vision*, pp. 1658–1667 (2020)
32. Vidal, A.R., Rebecq, H., Horstschaefer, T., Scaramuzza, D.: Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. *IEEE Robot. Autom. Lett.* **3**(2), 994–1001 (2018)
33. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: a unifying approach. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1328–1338 (2019)
34. Wang, Q., Zhang, Y., Yuan, J., Lu, Y.: Space-time event clouds for gesture recognition: from RGB cameras to event cameras. In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1826–1835. IEEE (2019)
35. Xu, J., Jiang, M., Yu, L., Yang, W., Wang, W.: Robust motion compensation for event cameras with smooth constraint. *IEEE Trans. Comput. Imaging* **6**, 604–614 (2020)
36. Zhu, A.Z., Atanasov, N., Daniilidis, K.: Event-based feature tracking with probabilistic data association. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4465–4470. IEEE (2017)