Asynchronous Tracking-by-Detection on Adaptive Time Surfaces for Event-based Object Tracking

Haosheng Chen*
Xiamen University
Xiamen, China
haoshengchen@stu.xmu.edu.cn

Qiangqiang Wu*
Xiamen University
Xiamen, China
qiangwu@stu.xmu.edu.cn

Yanjie Liang*
Xiamen University
Xiamen, China
yanjieliang@yeah.net

Xinbo Gao
Xidian University
Xi'an, China
xbgao@mail.xidian.edu.cn

Hanzi Wang*†
Xiamen University
Xiamen, China
wang.hanzi@gmail.com

arXiv:2002.05583v1 [cs.CV] 13 Feb 2020

ABSTRACT

Event cameras, which are asynchronous bio-inspired vision sensors, have shown great potential in a variety of situations, such as fast motion and low illumination scenes. However, most of the event-based object tracking methods are designed for scenarios with untextured objects and uncluttered backgrounds. There are few event-based object tracking methods that support bounding box-based object tracking. The main idea behind this work is to propose an asynchronous Event-based Tracking-by-Detection (ETD) method for generic bounding box-based object tracking. To achieve this goal, we present an Adaptive Time-Surface with Linear Time Decay (ATSLTD) event-to-frame conversion algorithm, which asynchronously and effectively warps the spatio-temporal information of asynchronous retinal events to a sequence of ATSLTD frames with clear object contours. We feed the sequence of ATSLTD frames to the proposed ETD method to perform accurate and efficient object tracking, which leverages the high temporal resolution property of event cameras. We compare the proposed ETD method with seven popular object tracking methods, that are based on conventional cameras or event cameras, and two variants of ETD. The experimental results show the superiority of the proposed ETD method in handling various challenging environments.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Tracking**.

KEYWORDS

Event-based Object Tracking, Event-based Object Detection, Event Camera, Adaptive Time Surface

*With Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, China.
†The corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM '19, October 21–25, 2019, Nice, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350975

ACM Reference Format:
Haosheng Chen, Qiangqiang Wu, Yanjie Liang, Xinbo Gao, and Hanzi Wang. 2019. Asynchronous Tracking-by-Detection on Adaptive Time Surfaces for Event-based Object Tracking. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350975

1 INTRODUCTION

Event cameras (e.g., DVS [21], DAVIS [6] and ATIS [32]) are bio-inspired silicon-retina visual sensors with very high dynamic range and temporal resolution (>120 dB, <1 ms). Thanks to the asynchronous nature of the biological retina, it can precisely and efficiently capture motion information [28, 30], especially for motions caused by moving objects [29, 37], in natural scenes. Inspired by the biological retina, unlike traditional cameras, asynchronous event cameras contain an array of independent pixels that asynchronously respond to pixel intensity changes in the environment. The asynchronous property of event cameras makes themselves suitable for object tracking and motion estimation.

If the intensity of a pixel on an event camera exponentially increases, the event camera will record an "On" event, which includes the coordinate of the event pixel and the current timestamp. In contrast, if the intensity of the pixel exponentially decreases, the event camera will record an "Off" event with the pixel coordinate and the current timestamp. Since the intensity changes are usually caused by object motions, event cameras can filter out non-motion information (for example, the static part of a scene) from their visual input, under stable illumination conditions or infrequent light variations. Thus event-based methods can acquire a clear and accurate clue about where object movement occurs, and save a lot of computational cost for searching moving objects.

Event cameras have achieved many successes on camera motion estimation and optical flow estimation (e.g., [11, 18, 40]). These successes have shown the superiority of event cameras on motion related tasks. However, there are only a few studies devoted to event-based object tracking, and most of these studies are designed for some special scenarios (e.g., [31] is for the pedestrian tracking scenario). Therefore, in this paper, we study generic object tracking methods based on event cameras.

In this study, we present an asynchronous event-to-frame conversion algorithm, which asynchronously warps the spatio-temporal information in retinal events to a novel Adaptive Time-Surface

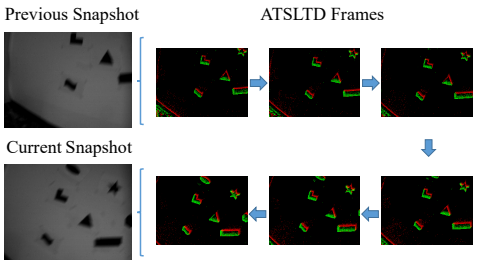

Figure 1: An illustration of a sequence of ATSLTD frames. The left two images show two snapshots, while the right six images show the sequence of ATSLTD frames between the two snapshots.

with Linear Time Decay (ATSLTD) frame representation, as shown in Fig. 1. The conversion is driven by object motions: (1) Fast object motions will create more ATSLTD frames than slow object motions, which creates a relatively large spatio-temporal space for object tracking; (2) If the target object does not have enough large motion (displacement), there is no any generated ATSLTD frame, which increases the computational speed. Since the intensity changes and retinal events usually occur at the edges of the target object, the ATSLTD frames record clear and sharp object contours. Based on the generated ATSLTD frames, we propose an effective and efficient Event-based Tracking-by-Detection (ETD) method for event-based object tracking. The proposed ETD method can leverage the spatio-temporal consistency between adjacent ATSLTD frames and perform accurate and high-speed event-based object tracking. Overall, this study makes the following contributions:

- We present an ATSLTD event-to-frame conversion algorithm to asynchronously warp the spatio-temporal information in retinal events, created by event cameras, to ATSLTD frames. The conversion is driven by object motions, and it does not rely on empirical cut-off thresholds.
- We introduce a new metric, named Non-Zero Grid Entropy (NZGE), to measure the amount of information in ATSLTD frames for adaptive event-to-frame conversion. And we calculate a confidence interval of NZGE values to perform the event-to-frame conversion algorithm asynchronously.
- We propose an Event-based Tracking-by-Detection (ETD) method to effectively and efficiently perform bounding box-based object tracking on the ATSLTD frames.

We evaluate the proposed ETD method on a mixed challenging event dataset consisting of a part of the event camera dataset and the extreme event dataset. The experimental results demonstrate the superiority of our ETD method when it is compared with several popular object tracking methods.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 describes the ATSLTD event-to-frame conversion algorithm and the proposed ETD method. The performance of the ETD method is extensively evaluated and analyzed in Section 4, and conclusions are drawn in Section 5.

2 RELATED WORK

In this section, we firstly review some event-to-frame conversion works, then we introduce previous works on event-based motion estimation and object tracking.

Event-to-frame conversion works. There are several state-of-the-art works [12, 19, 24, 25, 36, 40] proposed to convert asynchronous retinal events to synchronous frames. However, all these works rely on one or more empirical cut-off thresholds, which limits their applications. Among these works, the Time Surface-based algorithm [19] uses a fixed radius to compute the spatial neighborhood as its cut-off threshold. Because the spatial neighborhood is defined in the context of a timestamp map, the fixed radius can be treated as another form of the time window. The representations in [12, 25, 36, 40] respectively use a fixed size of time window as their cut-off thresholds. The Adaptive Time-Slice representation in [24] uses either a constant event number or a fixed search radius as its cut-off threshold.

Using either a fixed time window or a constant event number as the cut-off threshold of sequential retinal events is an easy and straightforward way to perform an event-to-frame conversion. However, both types of cut-off thresholds (i.e., a fixed time window and a constant event number) cannot adapt to all circumstances. The generated event-based frames will be affected by the speed of motions and the texture of objects. In this study, we propose an adaptive and asynchronous event-to-frame conversion algorithm without using empirical cut-off thresholds.

Event-based motion estimation and object tracking works. Event-based works have achieved great success on motion related tasks. In the field of event-based optical flow estimation, [2] proposes a sliding window variational optimization algorithm to estimate the optical flow. [12] exploits the best point trajectories of the event data for optical flow estimation. The optical flow can also be estimated by a self-supervised deep neural network [40], or by a time-slice block-matching method [24]. In the field of event-based camera motion estimation, the 3D 6-DoF camera motion can be estimated by using an interleaved probabilistic filter [18], a photometric depth map [11], or an image feature-based extended Kalman filter [42]. There also are some works in relation to event-based feature tracking [14] and 3D reconstruction [34]. From these motion related works, we can observe that event-based methods usually outperform conventional methods, especially when coping with fast motion and High Dynamic Range (HDR) scenes, due to the high temporal resolution and HDR properties of event cameras.

However, compared with the conventional object tracking methods (such as [4, 10, 13, 20, 38, 39]), there are only a few works devoted to event-based object tracking. The works can be roughly divided into two categories: i.e., clustering based works and non-clustering based works. The works in the first category require a clustering process to group retinal events into clusters. For example, [22] proposes a cluster-based object tracking method, which is inspired by the mean-shift algorithm [8, 9]. [31] uses the Gaussian mixture model to group retinal events into clusters and track the clusters with occlusions. Similarly, [7] also relies on a clustering algorithm, which is a variant of [22], to track objects with occlusions. [15] proposes a variant of the particle filters to track the clusters grouped by an event-based Hough transform algorithm [1].

In the second category of works, [23] proposes a particle filter-based object tracking method, which exploits the information from both video frames and retinal events. For some recent works, [26] proposes a Kalman filter-based object tracking method, which uses

Figure 2: Comparison of the Fixed Time Window (FTW) event-to-frame conversion algorithm and the proposed ATSLTD event-to-frame conversion algorithm. Video Frames (a-c) respectively show two snapshots, which are seen by a conventional camera before and after the events-to-frame conversion is implemented. FTW Frames (a-c) respectively show the generated FTW frames by the FTW event-to-frame conversion algorithm for fast object motion (a), slow object motion (b) and rich object texture (c) cases. ATSLTD Frames (a-c) respectively show the ATSLTD frames generated by the proposed ATSLTD event-to-frame conversion algorithm for the same cases. Red and green contours are respectively generated by "On" and "Off" events.

a motion compensation algorithm. [33] proposes a sliding window-based method for long-term object tracking. [3] presents a Kalman filter-based object tracking method for multi-target tracking.

For the above-mentioned object tracking methods, we observe that most of them are designed at pixel level, which cannot support generic bounding box-based object tracking. In this study, we propose an event-based tracking-by-detection method to support generic bounding box-based object tracking.

3 THE PROPOSED METHOD

In this section, we present the Adaptive Time-Surface with Linear Time Decay (ATSLTD) event-to-frame conversion algorithm for the sequential retinal events generated by an event camera in Sec. 3.1. Then we introduce a metric named Non-Zero Grid Entropy (NZGE), to measure the amount of information in generated ATSLTD frames and we compute a confidence interval of the NZGE values to perform the asynchronous ATSLTD event-to-frame conversion in Sec. 3.2. Finally, we propose an event-based tracking-by-detection (ETD) method which works on the ATSLTD frames in Sec. 3.3. Along with the ETD method, we also propose an event-based tracking recovery strategy for handling the tracking failure situation. Next, we will introduce the proposed method in detail.

3.1 Adaptive Time-Surface with Linear Time Decay

Along with the pixel-level intensity changes that are caused by camera motions and object motions, sequential retinal events $\mathcal{E}$ are generated asynchronously by an event camera. Each event e of $\mathcal{E}$ can be represented as a quadruple:

$$e \doteq (u, v, p, t), \tag{1}$$

where u and v are the horizontal and vertical coordinates of e; p indicates the polarity (On or Off) of e, and t is the timestamp of e. Event cameras have very high spatio-temporal resolution and high HDR, therefore event-based object tracking methods can benefit a lot from the retinal event input captured by event cameras. Since each pixel of an event camera can occur independently in response to log intensity changes, the asynchronous nature of retinal events makes it difficult for conventional frame-based object detection and tracking methods to process the retinal events directly. As a result, we need an event-to-frame conversion algorithm, which warps the retinal events to an event-based frame representation.

Effective event-based frame representations for object detection and tracking should have sharp and clear object contours for moving objects. Meanwhile, the object contours should not have too much displacement, and should not be too sparse. The state-of-the-art event-to-frame conversion algorithms [12, 19, 24, 25, 36, 40] use either a fixed time window or a constant event number as the cut-off threshold to perform the event-to-frame conversion. For the Fixed-Time-Window (FTW) event-to-frame conversion algorithms, we show some results obtained by an example FTW event-to-frame conversion algorithm using a fixed time window of 9 ms in the FTW Frames (a-c) of Fig. 2. The example conversion can create clear object contours for objects without highly speedy motions

and complicated textures. However, the example algorithm generates sparse object contours for objects with slow motions (as shown in the FTW Frames (b) of Fig. 2), and it generates blurred object contours with large contour displacement for the fast object motion and rich object texture situations (as shown in the FTW Frames (a) and (c) of Fig. 2, respectively). From Fig. 2, we can conclude that different object textures and speeds will result in different amounts of retinal events. Therefore, for the constant-event-number event-to-frame conversion algorithms, they are difficult to determine the constant event number. In addition, both types of the event-to-frame conversion algorithms are synchronous, which is incompatible with the asynchronous nature of event cameras.

In this work, we propose an event-to-frame conversion algorithm, which warps sequential retinal events to the Adaptive Time-Surface with Linear Time Decay (ATSLTD) frame representation. The ATSLTD representation also uses a time map to represent the spatio-temporal information in retinal events, which is similar to the Time-Surface representation in [19]. However, the main difference is that the proposed ATSLTD event-to-frame conversion algorithm is asynchronous. It is driven by object motions (faster object motions will create more ATSLTD frames than slower object motions), and it does not need any empirical cut-off thresholds for converting sequential retinal events to event-based frames. Moreover, we introduce an efficient linear time decay kernel to the proposed ATSLTD event-to-frame conversion algorithm for recording object contours on the ATSLTD frames.

When we implement the conversion of retinal events to the ATSLTD representation, the i-th ATSLTD frame $\mathcal{F}_i$ is initialized to a three-dimensional zero matrix $\mathcal{F}_i \in \mathbb{N}^{h\times w\times 2}$ at timestamp $\mathcal{T}_{i-1}$ when the previous (i-1)-th ATSLTD frame $\mathcal{F}_{i-1}$ is finished. Here h and w are the height and width of the event camera resolution, and the third dimension represents binary polarities of events (i.e., *On* retinal events and *Off* retinal events will be stored separately to avoid the mutual interference). Then each of the incoming retinal events will trigger an update on $\mathcal{F}_i$, which will multiply a linear time decay factor t_{k-1}/t_k with $\mathcal{F}_i$ and then set a value of 255 to an element of $\mathcal{F}_i$ at the coordinates (u, v) that correspond to the triggering event. For the k-th update caused by the k-th event $e_k = \{u_k, v_k, p_k, t_k\}$, the update is calculated as follow:

$$\mathcal{F}_i = round(\mathcal{F}_i * t_{k-1}/t_k), \quad \mathcal{F}_i(u_k, v_k) = 255, \tag{2}$$

where t_{k-1} is the timestamp of the (k-1)-th event e_{k-1} and t_k is the timestamp of e_k. Finally, when the amount of information in $\mathcal{F}_i$ reaches the calculated confidence interval at the current timestamp $\mathcal{T}_i$, the ATSLTD event-to-frame conversion for $\mathcal{F}_i$ is finished and $\mathcal{F}_i$ becomes a new ATSLTD frame for object tracking.

Since pixel intensity changes usually occur at the edges of moving objects, retinal events will occur at the edges of moving objects in response to the pixel intensity changes. As a result, after the warping process, the linear time decay factor t_{k-1}/t_k in Eq. (2) will form a pixel intensity decay at the edges of each moving object. The pixel intensity decay records a sequence of adjacent object contours (i.e., an object contour displacement) for the corresponding moving object, as shown in the ATSLTD Frames (a-c) of Fig. 2. And a newly generated object contour of the sequential object contours has a higher pixel intensity than a previously generated object contour, which makes object contour-based detectors (e.g., EdgeBoxes [43])

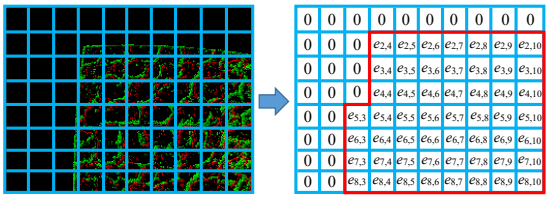

Figure 3: An illustration of the non-zero grid entropy calculation. The left and right sub-figures show an ATSLTD frame and the corresponding entropy map. The grids with non-zero entropy values are in the red zone of the map.

pay more attention to the newly generated object contours in object detection. Therefore, the ATSLTD frame representation facilitates the proposed event-based tracking-by-detection method to detect and locate moving objects effectively and efficiently.

3.2 Confidence Interval

As discussed in Sec. 3.1, effective event-based frame representations should have sharp and clear object contours for moving objects. Before calculating the confidence interval when the proposed asynchronous ATSLTD event-to-frame conversion algorithm is implemented, we need a metric to measure the displacement of the object contours in ATSLTD frames. Since the image entropy is a natural way to measure the amount of information in ATSLTD frames, in this work, we propose the Non-Zero Grid Entropy (NZGE) measure as the metric. To calculate the NZGE value of an ATSLTD frame, we divide the ATSLTD frame into $p \times q$ grids, each of which is a $r \times r$ image patch. Then we compute an image entropy for each of the grids to build an entropy map, as shown in Fig. 3. Finally the NZGE value of the i-th ATSLTD frame is calculated as follow:

$$NZGE_i = \frac{1}{n_i^{grid}} \sum_{x=1}^{p} \sum_{y=1}^{q} entropy_{x,y}^{i}, \tag{3}$$

where n_i^{grid} is the number of the grids with non-zero entropy values. $entropy_{x,y}^{i}$ is the image entropy of the image patch in the x-th row and y-th column. The image entropy is defined as follow:

$$entropy_{x,y} = -\sum_{z=0}^{255} prob_z^{x,y} \log prob_z^{x,y}, \tag{4}$$

where $prob_z^{x,y}$ is the probability of a pixel having gray level z in the x-th row and y-th column of the image patch (the range of the gray level are 0 to 255). The parameters p, q and r, are respectively set to 45, 60 and 4 in this work. For the proposed two-channel ATSLTD frame representation, we respectively calculate two image entropy values for the two channels. The mean of the two image entropy values will be the image entropy of the two-channel ATSLTD frame.

Event cameras will filter out the static part of a scene from their output. Thus, if we calculate an image entropy value over the whole ATSLTD frame, the zero entropy part of the ATSLTD frame will degrade the accuracy of the metric. Otherwise, if we calculate an image entropy value only for the pixels with non-zero gray levels, some small sensor noise may dominate the image entropy value. Therefore, we propose the NZGE to measure the amount of information in ATSLTD frames. The amount of information in an ATSLTD frame will increase along with object motions. As a

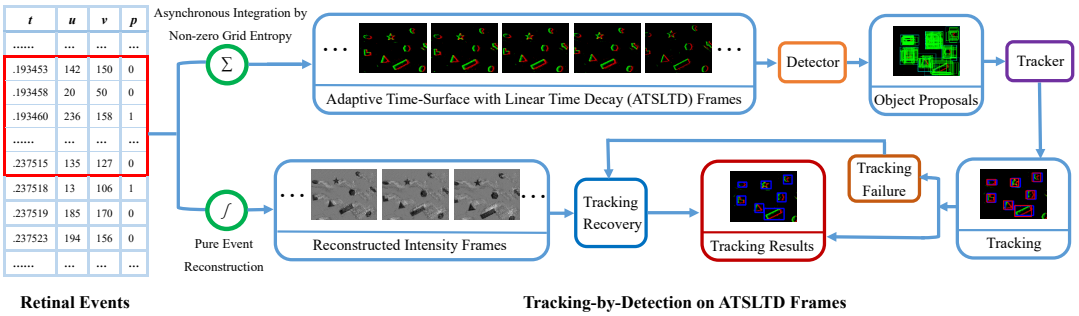

Figure 4: The pipeline of the proposed ETD method. Initially, retinal events are asynchronously warped to a sequence of ATSLTD frames. Then the detector of ETD generates object proposals on each of the ATSLTD frames. And the tracker of ETD selects the best bounding box from the generated object proposals as the tracking result. Finally, if the tracker loses the tracked object, ETD will use the intensity frames, which are reconstructed from the retinal events, to recover tracking from the failure.

result, we can use a confidence interval of NZGE values to maintain the object contour displacement in an ATSLTD frame within a reasonable range, which creates clear and sharp object contours for object detection and object tracking.

To calculate the lower and upper bounds of the confidence interval, we collect a set of NZGE values $C = \{c_1, c_2, \ldots, c_{n^s}\}$ from a set of sampled ATSLTD frames with a variety of clear and sharp object contours. The sample mean and the sample variance of C are $\overline{C}$ and S_C^2, respectively. We thereby can use the set of suitable NZGE values to create clear and sharp object contours. We assume that the suitable NZGE values are independent and distributed as a normal distribution $\mathcal{N}(\mu, \sigma^2)$. Thus, C is n^s observations from the normal distribution. Here we define a pivotal quantity g as:

$$g = \frac{\overline{C} - \mu}{S_C / \sqrt{n^s}} \tag{5}$$

As a result, g follows the t-distribution $t(n^s - 1)$ with $n^s - 1$ degrees of freedom. From Eq. (5), we can estimate a confidence interval for the mean μ of the normal distribution $\mathcal{N}(\mu, \sigma^2)$, as follow:

$$\overline{C} - |g_{\omega/2}| \frac{S_C}{\sqrt{n^s}} < \mu < \overline{C} + |g_{\omega/2}| \frac{S_C}{\sqrt{n^s}}, \tag{6}$$

where ω is a two-sided significance level. We use $\omega = 0.05$ in this work, which means:

$$\Pr\left(\overline{C} - |g_{0.025}| \frac{S_C}{\sqrt{n^s}} < \mu < \overline{C} + |g_{0.025}| \frac{S_C}{\sqrt{n^s}}\right) = 0.95 \tag{7}$$

Finally, the calculated confidence interval for NZGE values is:

$$[\alpha, \beta] = \left[\overline{C} - |g_{0.025}| \frac{S_C}{\sqrt{n^s}}, \overline{C} + |g_{0.025}| \frac{S_C}{\sqrt{n^s}}\right] \tag{8}$$

In this work, we collect $n^s = 100$ samples with $\overline{C} = 0.08795$ and $S_C = 0.02394$ for the calculation of the confidence interval. According to the t-distribution table, $g_{0.025} = 1.984$. Therefore, the calculated confidence interval $[\alpha, \beta] \approx [0.0832, 0.0927]$. The ATSLTD frames, whose NZGE values lie in the calculated confidence interval, will be more likely to have clear and sharp object contours. Thus the calculated confidence interval is used in the ATSLTD event-to-frame conversion algorithm to trigger the ATSLTD frame generation effectively and asynchronously.

3.3 Event-based Tracking-by-Detection

Since the generated ATSLTD frames have shown clear object contours and they have a high spatio-temporal resolution, the proposed event-based tracking-by-detection method does not need complicated mechanisms, which may increase the computational cost. In this work, we propose an effective and efficient two-stage Event-based Tracking-by-Detection (ETD) method, as shown in Fig. 4. In the first stage, the detector of the proposed ETD generates a set of object proposals on each of the sequential ATSLTD frames for the target object specified in the first ATSLTD frame. Then the ETD method uses an Intersection over Union (IoU)-based tracker to select the best object proposal from the generated object proposals as the tracking result in the current frame.

The generated ATSLTD frames have recorded clear and sharp object contours. Thus object contour-based detectors can take advantage of the input ATSLTD frames. In the detection stage, we exploit EdgeBoxes [43] as the detector of the proposed ETD method for its high recall rate and fast speed, and it is used to generate object proposals. Each of the generated object proposals is an object bounding box in the current i-th ATSLTD frame $\mathcal{F}_i$ for the bounding box O_{i-1} of the target object in the previous $(i-1)$-th ATSLTD frame $\mathcal{F}_{i-1}$. The center location and size of O_{i-1} are c_{i-1} and (w_{i-1}, h_{i-1}), respectively. We only detect object proposals within a searching region $\mathcal{R}_i$ for high computational speed. The center location and size of $\mathcal{R}_i$ are c_{i-1} and $(\tau w_{i-1}, \tau h_{i-1})$. τ is a parameter to render the searching region slightly larger than the previous size.

By leveraging the high spatio-temporal resolution property of the ATSLTD frames, we can improve the generated object proposals to create a set of refined object proposals for tracking. For each of the object proposals, we compute a score between O_{i-1} and the bounding box P_i of the object proposal. The score between O_{i-1} and P_i is calculated as follows:

$$\text{score}(O_{i-1}, P_i) = \phi\left(\frac{w_{i-1} \times h_{i-1}}{w_{p_i} \times h_{p_i}}\right) \times \phi\left(\frac{w_{i-1}/h_{i-1}}{w_{p_i}/h_{p_i}}\right), \tag{9}$$

where w_{i-1} and h_{i-1} are the width and height of O_{i-1}, w_{p_i} and h_{p_i} are the width and height of P_i, and the function $\phi(\cdot)$ is defined as:

$$\phi(x) = \begin{cases} x & 0 < x < 1 \\ 1/x & x \geq 1 \end{cases} \tag{10}$$

Eq. (9) indicates that if P_i is a refined object proposal of O_{i-1}, they should not have much change in terms of area and aspect ratio. After

Table 1: The details of the mixed event dataset. The FM, BC, SO, HDR and OC, in the Challenges column, are fast motion, background clutter, small object, HDR scene and occlusion, respectively.

Dataset	Sequence names	Feature	Challenges
ECD	shapes_translation	B&W shape objects mainly with translations	FM
ECD	shapes_6dof	B&W shape objects with various 6-DoF motions	FM
ECD	poster_6dof	Natural textures with cluttered background and various 6-DoF motions	FM+BC
ECD	slider_depth	Various artifacts at different depths with only translations	BC
EED	fast_drone	A fast moving drone under a very low illumination condition	FM+SO+HDR
EED	light_variations	Same with the upper one with extra periodical abrupt flash lights	FM+SO+HDR
EED	what_is_background	A thrown ball with a dense net as foreground	FM+OC
EED	occlusions	A thrown ball with a short occlusion under a dark environment	FM+OC+HDR

scoring all the generated object proposals, these object proposals are filtered to a set of refined object proposals that have a higher score than a threshold λ.

After the detection stage, we have a set of refined object proposals, which are the candidates of the target object in the current ATSLTD frame. The tracker of the proposed ETD method will further refine the candidates to find the best candidate for the target object. As described in Section 3.2, the object motion between every two adjacent ATSLTD frames is constrained to a moderate level in the ATSLTD event-to-frame conversion. Therefore the bounding boxes for the target object from two adjacent ATSLTD frames should have a large overlap with each other, which is suitable for IoU-based trackers (such as [5]). The IoU measure, between the two bounding boxes O_{i-1} and O_i, is defined as:

$$\mathrm{IoU}(O_{i-1}, O_i) = \frac{Area(O_{i-1}) \cap Area(O_i)}{Area(O_{i-1}) \cup Area(O_i)} \tag{11}$$

In the tracking stage, we exploit a simple yet effective IoU-based tracker, which chooses the candidate bounding box that has the largest IoU with the previous bounding box O_{i-1}, as the estimated bounding box O_i of the target object in the i-th ATSLTD frame $\mathcal{F}_i$.

For the tracking failure situation, the proposed ETD method also provides a tracking recovery strategy for it, as shown in Fig. 4. Along with the ATSLTD frames, there are sequential synchronized intensity event frames, which are reconstructed using the efficient pure event reconstruction method in [35]. If an IoU between an estimated bounding box and its previous bounding box, is under a threshold μ, we treat this situation as a tracking failure. And we use the DaSiamRPN tracker [41] to track the target object again on the reconstructed intensity event frames for recovering the tracking status from the tracking failure.

4 EXPERIMENTS

4.1 Experimental Settings

In this section, we evaluate the proposed ETD method and nine competing methods on a challenging mixed event dataset, which includes a part of the Event Camera Dataset (ECD) [27] and the Extreme Event Dataset (EED) [26]. The two datasets were recorded using a DAVIS [6] event camera in real-world environments. The details of the mixed event dataset are shown in Table 1. Note that the mixed event dataset consists of both the event data sequences and the corresponding video sequences for each sequence. Since the ECD dataset does not provide the ground truth for object tracking, we manually label a rectangle bounding box for each object as the ground truth in the ECD dataset.

In this work, we select nine competing object tracking methods, including KCF [16], TLD [17], SiamFC [4], ECO [10], DaSiamRPN [41], an event-based variant of DaSiamRPN [41] (called as DaSiamRPN-E), E-MS [3] and two variants of our ETD method (called as ETD-FTW and ETD-NR) as the competitors. About these competitors: KCF [16] is an effective correlation filter-based object tracking method. TLD [17] is a classical and robust Tracking-by-Detection method. SiamFC [4] is an efficient and effective object tracking method, which uses a fully-convolutional Siamese network to track objects. ECO [10] is a state-of-the-art object tracking method that employs compact samples to train continuous convolutional operators for visual tracking. DaSiamRPN [41] is a state-of-the-art tracking method that improves the fully-convolutional siamese network by using distractor-aware training. DaSiamRPN-E, which uses ATSLTD frames as its input, is an event-based variant of DaSiamRPN. E-MS [3] is a state-of-the-art event-based target tracking method based on both mean-shift clustering and Kalman tracking. Here the E-MS method is extended to support bounding box-based evaluation by using the minimum enclosing bounding boxes of those events belonging to the same cluster center as its tracking results. ETD-FTW and ETD-NR are two variants of ETD. ETD-FTW uses the Fixed Time Window (FTW) frames (as shown in Fig. 2) instead of using the ATSLTD frames as its input. ETD-NR removes the tracking recovery part from the original ETD method. The two variants are used to show the influence of the components of the proposed ETD method on its performance.

For KCF and TLD, we use their OpenCV implementations with the suggested settings. For SiamFC and ECO, we use their original Matlab implementations with the default parameters. For DaSiamRPN and E-MS, we respectively use their original Python and C++ implementations with the default parameters. For the proposed ETD method, the maxBoxes and minBoxArea of the Edgeboxes detector are respectively set to 1000 and 100 considering the balance between speed and accuracy. The parameter τ for the searching region is set to 1.5. The threshold λ for refining object proposals is set to 0.7. The threshold μ, which is used to judge the tracking failure situation, is set to 0.3. We fix these parameter values for all the following experiments.

4.2 Evaluation Metrics

During the evaluation, we run all of the competing methods N^{rep} times. If a sequence of the dataset includes multiple objects, we evaluate all the objects separately. If a tracking failure occurs for a competing method, we will reinitialize the method at the next frame. To evaluate the precision of all the methods, we calculate

Table 2: Results obtained by the nine competing methods and the proposed ETD method on the four event sequences from the ECD dataset. The best values are highlighted by bold.

Method	shapes_translation		shapes_6dof		poster_6dof		slider_depth	
	AP	AR	AP	AR	AP	AR	AP	AR
KCF[16]	0.306	0.468	0.307	0.430	0.345	0.472	0.674	0.780
TLD[17]	0.468	0.425	0.430	0.424	0.472	0.554	0.504	0.643
SiamFC[4]	0.685	0.872	0.668	0.842	0.585	0.726	0.806	0.910
ECO[10]	0.746	0.931	0.717	0.877	0.614	0.719	**0.915**	0.667
DaSiamRPN[41]	0.668	0.878	0.642	0.831	0.517	0.651	0.713	0.890
DaSiamRPN-E[41]	0.728	0.913	0.749	0.878	0.692	0.814	0.803	0.974
E-MS[3]	0.675	0.768	0.612	0.668	0.417	0.373	0.447	0.350
ETD-FTW	0.734	0.922	0.727	0.924	0.707	0.918	0.636	0.766
ETD-NR	0.793	0.982	0.783	0.972	0.775	0.992	0.803	0.983
ETD	**0.817**	**0.998**	**0.809**	**0.998**	**0.788**	**0.995**	0.816	**0.997**

Table 3: Results obtained by the nine competing methods and the proposed ETD method on the four event sequences of the EED dataset. The best values are highlighted by bold.

Method	fast_drone		light_variations		what_is_background		occlusions	
	AP	AR	AP	AR	AP	AR	AP	AR
KCF[16]	0.169	0.176	0.107	0.066	0.028	0.000	0.004	0.000
TLD[17]	0.315	0.118	0.045	0.066	0.269	0.333	0.092	0.167
SiamFC[4]	0.559	0.667	0.599	0.675	0.307	0.308	0.148	0.000
ECO[10]	0.637	0.833	0.586	0.688	0.616	0.692	0.108	0.143
DaSiamRPN[41]	0.673	0.853	0.654	0.894	**0.678**	**0.833**	0.189	0.333
DaSiamRPN-E[41]	0.646	0.847	0.705	0.902	0.573	0.742	0.373	0.605
E-MS[3]	0.313	0.307	0.325	0.321	0.362	0.360	0.356	0.353
ETD-FTW	0.576	0.673	0.722	0.874	0.562	0.733	0.263	0.533
ETD-NR	0.722	0.883	0.833	0.925	0.638	0.781	0.414	0.622
ETD	**0.738**	**0.897**	**0.842**	**0.933**	0.653	0.807	**0.431**	**0.647**

the Average Precision (AP) as follow:

$$AP = \frac{1}{N^{rep}} \frac{1}{N^{obj}} \sum_{a=1}^{N^{rep}} \sum_{b=1}^{N^{obj}} \frac{O_{a,b}^{E} \cap O_{a,b}^{G}}{O_{a,b}^{E} \cup O_{a,b}^{G}}, \tag{12}$$

where $O_{a,b}^{E}$ is the estimated bounding box in the a-th round of the evaluation for the b-th object, and $O_{a,b}^{G}$ is the corresponding ground truth. N^{rep} is the repeat times of the evaluation, and N^{obj} is the number of objects in the current sequence. We set N^{rep} to 5.

We also calculate the Average Robustness (AR) to measure the robustness of all the methods as follow:

$$AR = \frac{1}{N^{rep}} \frac{1}{N^{obj}} \sum_{a=1}^{N^{rep}} \sum_{b=1}^{N^{obj}} success_{a,b}, \tag{13}$$

where $success_{a,b}$ indicates that whether the tracking in the a-th round for the b-th object is successful or not (1 means success and 0 means failure). If the AP value obtained by a method for an object is under 0.5, we will consider it as a tracking failure case.

4.3 Evaluation on the Mixed Event Dataset

In the mixed event dataset, we use the *shapes_translation*, *shapes_6dof*, *poster_6dof* and *slider_depth* sequences as the representative sequences of the ECD dataset [27] for comparison. The first three sequences include fast object motions. The third and fourth sequences include complicated background clutters. And the object textures in the four sequences vary from simple B&W shapes to complicated artifacts. For these sequences, we mainly concern about the performance of all methods for a variety of object motions, especially for fast motion, and for background clutter.

The quantitative results obtained by the competing methods are given in Table 2. Moreover, some representative qualitative results obtained by SiamFC, ECO, DaSiamRPN, E-MS and the proposed ETD are shown in the top three rows of Fig. 5. From Table 2, we can see that the proposed ETD achieves the best performance on the first three sequences and it achieves the best AR and the second best AP on the fourth sequence. In comparison, ECO achieves the best AP on the fourth sequence. However, it has a relatively low AR compared with the proposed ETD. We find that it is the reinitialization protocol that helps ECO to achieve the high AP. In addition, as shown in Fig. 5, the proposed ETD has achieved better performance in handling fast object motions. In comparison, KCF, TLD, SiamFC, ECO and DaSiamRPN usually achieve low precision values, due to the influence of motion blur. E-MS can handle most fast motions. However, it is less effective to handle cluttered backgrounds (e.g., for the *poster_6dof* and *slider_depth* sequences). As a variant of DaSiamRPN, DaSiamRPN-E achieves satisfying results by exploiting the virtues of the ATSLTD representation. Comparing with the proposed ETD method, ETD-FTW and ETD-NR have achieved inferior performance, which shows that both the ATSLTD representation and the tracking recovery component can improve the performance of the proposed ETD. However, as the quantitative

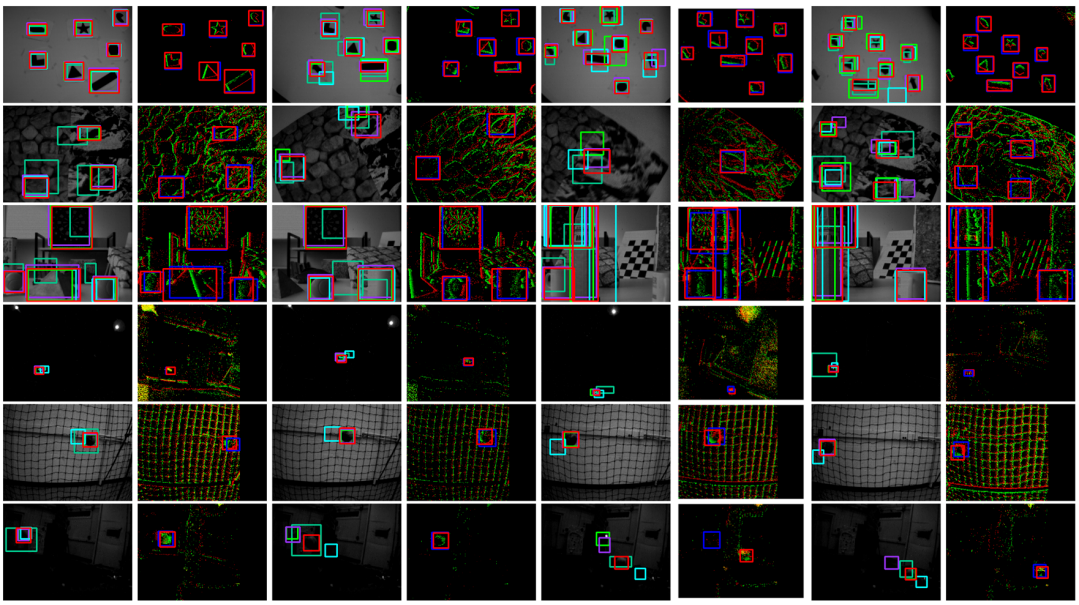

Figure 5: Tracking results obtained by SiamFC, ECO, DasiamRPN, E-MS and the proposed ETD method. Each row shows a representative sequence of the mixed event dataset. From top to bottom, the sequences are *shape_6dof*, *poster_6dof*, *slider_depth*, *light_variations*, *what_is_background* and *occlusions*, respectively. From left to right, the first, third, fifth and seventh columns respectively show the results of SiamFC, ECO, DaSiamRPN and E-MS. The second, fourth, sixth and eighth columns respectively show the results of the proposed ETD method. Best viewed in color.

results show, the ATSLTD representation has more influence on the performance of ETD than the tracking recovery component. In summary, the evaluation on the ECD dataset demonstrates the superiority of the proposed ETD in handling fast motion, various textured objects and cluttered backgrounds.

Moreover, we also use the recently proposed EED dataset to evaluate the competing methods. The EED dataset [26] contains four sequences: *fast_drone*, *light_variations*, *what_is_background* and *occlusions*. The first three sequences respectively record a small and fast moving drone under HDR environments. The fourth sequence records a moving ball with a net as foreground. Using this dataset, we evaluate the influence of HDR scenes and occlusions on the performance of the competing methods.

The quantitative results are shown in Table 3 and some representative qualitative results are shown in the bottom three rows of Fig. 5. From the results, we can see that the proposed ETD achieves the best performance on most of the sequences except for the *what_is_background* sequence, on which it obtains the second best AP and AR. This is because that the foreground net in the *what_is_background* sequence partially occludes the tracked ball, which results in a negative influence on the corresponding contour of the tracked ball. Among the nine competitors, KCF, TLD, SiamFC, ECO and DaSiamRPN cannot effectively handle with fast motion and low illumination conditions. We also find that the performance of E-MS is significantly affected by the sensor noises in HDR environments. As a result, E-MS achieves poor results on the EED dataset. Moreover, ETD-FTW and ETD-NR are inferior to the proposed ETD, which shows the contribution of the ATSLTD representation and the tracking recovery component of the proposed ETD. Overall, the evaluation on the EED dataset shows that the proposed ETD can effectively handle HDR scenarios.

4.4 Time Cost

The proposed method is implemented using Python on a PC with an Intel i7, 32G RAM and an NVIDIA GTX 1080 GPU. For the mixed event dataset, the average computational time for tracking an object is 21.97 ms per ATSLTD frame. Note that only the tracking recovery part of the proposed ETD method is accelerated by GPU, while the other parts of the ETD method work on CPU.

5 CONCLUSION

In this paper, we present a novel event-to-frame conversion algorithm that can asynchronously warp the spatio-temporal information in sequential retinal events to ATSLTD frames. The event-to-frame conversion is driven by motions and it can record object contours more clearly on the generated ATSLTD frames, which facilitates to detect and track moving objects. Then, we propose an event-based tracking-by-detection method. It can effectively and efficiently track objects on the ATSLTD frames. Extensive experiments demonstrate the great advantages of the proposed ETD method for object tracking in challenging situations such as fast motion and HDR scenes.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. U1605252, 61872307, 61432014, 61772402 and 61671339).

REFERENCES

[1] Dana H Ballard. 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition* 13, 2 (1981), 111–122.

[2] Patrick Bardow, Andrew J Davison, and Stefan Leutenegger. 2016. Simultaneous optical flow and intensity estimation from an event camera. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 884–892.

[3] Francisco Barranco, Cornelia Fermuller, and Eduardo Ros. 2018. Real-time clustering and multi-target tracking using event-based sensors. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5764–5769.

[4] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. 2016. Fully-convolutional siamese networks for object tracking. In *Proc. of European Conference on Computer Vision*. Springer, 850–865.

[5] Erik Bochinski, Volker Eiselein, and Thomas Sikora. 2017. High-speed tracking-by-detection without using image information. In *Proc. of IEEE International Conference on Advanced Video and Signal Based Surveillance*. 1–6.

[6] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. 2014. A 240× 180 130 dB 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits* 49, 10 (2014), 2333–2341.

[7] Luis A Camuñas-Mesa, Teresa Serrano-Gotarredona, Sio-Hoi Ieng, Ryad Benosman, and Bernabé Linares-Barranco. 2017. Event-driven stereo visual tracking algorithm to solve object occlusion. *IEEE Transactions on Neural Networks and Learning Systems* 29, 9 (2017), 4223–4237.

[8] Dorin Comaniciu and Peter Meer. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5 (2002), 603–619.

[9] Dorin Comaniciu and Visvanathan Ramesh. 2000. Mean shift and optimal prediction for efficient object tracking. In *Proc. of IEEE International Conference on Image Processing*, Vol. 3. 70–73.

[10] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, Michael Felsberg, et al. 2017. ECO: Efficient Convolution Operators for Tracking. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 6638–6646.

[11] Guillermo Gallego, Jon EA Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. 2017. Event-based, 6-DOF camera tracking from photometric depth maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 10 (2017), 2402–2412.

[12] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. 2018. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 3867–3876.

[13] Xu Gao and Tingting Jiang. 2018. OSMO: Online Specific Models for Occlusion in Multiple Object Tracking under Surveillance Scene. In *Proc. of ACM on Multimedia Conference*. 201–210.

[14] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. 2018. Asynchronous, photometric feature tracking using events and frames. In *Proc. of European Conference on Computer Vision*.

[15] Arren Glover and Chiara Bartolozzi. 2017. Robust visual tracking with a freely-moving event camera. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 3769–3776.

[16] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. 2015. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 3 (2015), 583–596.

[17] Zdenek Kalal, Krystian Mikolajczyk, Jiri Matas, et al. 2012. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 7 (2012), 1409.

[18] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. 2016. Real-time 3D reconstruction and 6-DoF tracking with an event camera. In *Proc. of European Conference on Computer Vision*. 349–364.

[19] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. 2017. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 7 (2017), 1346–1359.

[20] Yanjie Liang, Qiangqiang Wu, Yi Liu, Yan Yan, and Hanzi Wang. 2018. Robust Correlation Filter Tracking with Shepherded Instance-Aware Proposals. In *Proc. of ACM on Multimedia Conference*. 420–428.

[21] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. 2008. A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits* 43, 2 (2008), 566–576.

[22] M Litzenberger, C Posch, D Bauer, AN Belbachir, P Schon, B Kohn, and H Garn. 2006. Embedded vision system for real-time object tracking using an asynchronous transient vision sensor. In *Digital Signal Processing Workshop*. 173–178.

[23] H. Liu, D. P. Moeys, G. Das, D. Neil, S. Liu, and T. DelbrÃijck. 2016. Combined frame- and event-based detection and tracking. In *Proc. of IEEE International Symposium on Circuits and Systems*. 2511–2514.

[24] Min Liu and Tobi Delbruck. 2018. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In *British Machine Vision Conference*.

[25] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso Garcıa, and Davide Scaramuzza. 2018. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 5419–5427.

[26] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. 2018. Event-based moving object detection and tracking. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1–9.

[27] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. 2017. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *The International Journal of Robotics Research* 36, 2 (2017), 142–149.

[28] Ben L Murphy-Baum and Gautam B Awatramani. 2018. An old neuron learns new tricks: redefining motion processing in the primate retina. *Neuron* 97, 6 (2018), 1205–1207.

[29] Bence P Ölveczky, Stephen A Baccus, and Markus Meister. 2003. Segregation of object and background motion in the retina. *Nature* 423, 6938 (2003), 401.

[30] Clyde W Oyster. 1968. The analysis of image motion by the rabbit retina. *The Journal of Physiology* 199, 3 (1968), 613–635.

[31] Ewa Piatkowska, Ahmed Nabil Belbachir, Stephan Schraml, and Margrit Gelautz. 2012. Spatiotemporal multiple persons tracking using dynamic vision sensor. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 35–40.

[32] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. 2011. A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE Journal of Solid-State Circuits* 46, 1 (2011), 259–275.

[33] Bharath Ramesh, Shihao Zhang, Zhi Wei Lee, Zhi Gao, Garrick Orchard, and Cheng Xiang. 2018. Long-term object tracking with a moving event camera. In *British Machine Vision Conference*.

[34] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. 2018. EMVS: Event-based multi-view stereoâĂŤ3D reconstruction with an event camera in real-time. *International Journal of Computer Vision* 126, 12 (2018), 1394–1414.

[35] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. 2018. Continuous-time intensity estimation using event cameras. In *Asian Conference on Computer Vision*.

[36] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. 2018. Hats: histograms of averaged time surfaces for robust event-based object classification. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 1731–1740.

[37] Benedict Wild. 2018. How does the brain tell self-motion from object motion? *Journal of Neuroscience* 38, 16 (2018), 3875–3877.

[38] Lingxiao Yang, Risheng Liu, David Zhang, and Lei Zhang. 2017. Deep location-specific tracking. In *Proc. of ACM on Multimedia Conference*. 1309–1317.

[39] Mengdan Zhang, Jiashi Feng, and Weiming Hu. 2017. Robust Visual Object Tracking with Top-down Reasoning. In *Proc. of ACM on Multimedia Conference*. 226–234.

[40] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. 2018. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *Proc. of Robotics: Science and Systems*.

[41] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. 2018. Distractor-aware siamese networks for visual object tracking. In *Proc. of European Conference on Computer Vision*. 101–117.

[42] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. 2017. Event-based visual inertial odometry. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. 5391–5399.

[43] C Lawrence Zitnick and Piotr Dollár. 2014. Edge boxes: Locating object proposals from edges. In *Proc. of European Conference on Computer Vision*. 391–405.