Object Tracking with a fusion of Event-based Camera and Frame-based Camera

Haixin Sun, Vincent Frémont

Object Tracking with a fusion of Event-based Camera and Frame-based Camera

Haixin SUN[1] and Vincent FREMONT[1]

Nantes University, Ecole Centrale de Nantes, CNRS LS2N, Nantes, France,
{Haixin.Sun, vincent.fremont}@ec-nantes.fr

Abstract. The framed-based camera is commonly used for object tracking tasks. However, under poor light conditions, the frame camera cannot work properly and provide stable detection for the tracking system. Also, it has motion blur when the object is moving fast. Compared to the frame-based camera, the event camera has a higher temporal resolution and dynamic range. The event data is transmitted immediately when the brightness of each pixel changes. The higher temporal resolution helps the event camera avoid the motion blur and provide more precise temporal information. Moreover, the higher dynamic range allows the event camera to work in extreme light conditions, such as evening, raining, and strong sunshine environments. In all, those advantages make the event cameras an applicable complementary sensor to the frame cameras. In this paper, several optical flow algorithms for the event data are tested. Then, a clustering algorithm is proposed to generate the detection and hybrid it with the detection from frame images. At last, the PMBM filter is used to realize object tracking and test it on our self-recorded dataset with the DAVIS346 and several publicly available datasets.

Keywords: Sensor Fusion, Event-based Camera, Optical Flow, Object Tracking

1 Introduction

1.1 Literature review

Monocular cameras are one of the most commonly used sensors for object tracking tasks in many different areas, such as autonomous vehicles, unmanned aerial vehicles (UAVs), and industrial robots. It transmits images synchronously frame by frame at a fixed rate. It has drawbacks such as low temporal resolution, redundant information and low dynamic range, etc. More importantly, the frame image is transmitted at a fixed frequency and does not contain time information, which is important to the object tracking problem. The event-based camera, a bio-inspired technology of silicon retinas, was proposed to solve classical and new computer vision tasks [1, 2]. Event cameras are asynchronous sensors that monitor the brightness change of each pixel related to the viewed scene with a precise timestamp, which means the event camera provides time information. Thus, event cameras have a large potential for computer vision applications

in challenging scenarios compared to standard cameras. Typically, event cameras are used on the sensing modalities such as Unmanned aerial vehicle (UAV) [3], robotics [4] or wearable electronics [5], where operation is under unrealistic lighting conditions and sensitive to the temporal resolution. The common applications for event cameras are object tracking [4], surveillance and monitoring [6], optical flow estimation [7, 8]. For autonomous driving, [9] proposed a method that can predict the vehicle's steering angle according to the event data, and [10] proposed a dataset that contains event data along with the vehicle control and diagnostic data.

The event cameras transmit data asynchronously, and the event data [2] is in a format of tuple $e(x, y, t, p)$, where (x, y) is the location of pixel, t denotes the timestamp of the event and p represents the polarity of the brightness change. When the logarithmic intensity of a pixel changes beyond the threshold τ, an event will be emitted. Apparently, the frame-based vision algorithms cannot be directly used on the event data, and new methods have to be developed. One of the important directions to explore the temporal information in the event data is optical flow. [8] leverages the high fidelity of frame image and the high temporal resolution of event data to calculate more accurate optical flow. [7, 11] use the deep neural network, and make use of a self-supervised training method to solve the problem of the lack of dataset. There are also several object detection and feature tracking algorithm [3, 12, 13] for event data that has been developed. However, despite the advantages that the event camera has, at the current time the frame image is still more robust in most scenarios. Because of the limitations of the object detection algorithms and the spatial resolution of the event camera. So it is better to use a hybrid of event data and frame image.

Once the object detection has been given, it should be tracked over time. Object tracking algorithms that are based on random finite sets (RFSs) are popular recently [14]. The RFS theory is proposed to model the MOT problems; it is the mathematically simplest version of point process theory [15]. It provides a carefully constructed practitioners' toolbox of explicit, rigorous, systematic, and general procedures. Among the family of MOT algorithms based on RFS, the Poisson multi-Bernoulli mixture (PMBM) is the state-of-art MOT algorithm [16], which has better performance than the cardinalized probability density (CPHD) filters and the generalized labeled multi-Bernoulli (GLMB) [17, 18]. The other competitors extract appearances of the objects from the image and achieve better association performance [19]. But the extraction requires an extra model which will increase the complexity of the tracking algorithm, especially with multiple sensors. The PMBM filter only relies on the location of objects and this makes the PMBM filter a suitable framework for the fusion between different sensors. So PMBM filter is used in our work.

1.2 Our contributions

This paper's main contribution is an object tracking framework that combines both Event-based camera and Frame-based camera information. This framework uses the event camera to add robustness in the frame camera in poor light or

fast motion environments. We present a new density-based clustering algorithm, STF-DBSCAN for the detection based on event data. We proposed a tracking approach that uses a hybrid of the event-based camera and frame-based camera, and this is the first approach that uses the event camera to assist the frame based camera in tracking objects. The presented object tracking algorithms have three components: detector, fusion strategy, and PMBM filter. The detector for the event camera is based on a clustering algorithm that makes use of the position, timestamp, and optical flow of the event data. For the frame image, we adopt the deep learning model and obtain the bounding box as the detection. Since the form of detection from the two sensors are different, and repeating measurements can reduce the Signal-to-noise ratio (SNR), the fusion strategy is needed to unite the form and remove the repetitive measurements.

The paper is structured as follows: In Section 2, we present some basic concepts and an overview of our approach. In section 3, the optical flow and STF-DBSCAN algorithms for event cameras are discussed, and in section 4, we present the fusion strategy and the details about the PMBM filters. The performance of our approach is shown and discussed in section 5.

2 Basic Concepts and Algorithm Overview

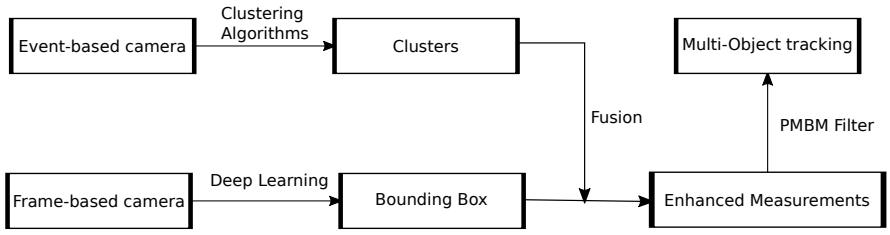


Fig. 1. Diagram of our approach.

The objective of our approach is to achieve more robust object tracking with a hybrid of event data and frame images. For the frame image, there are enough algorithms for reliable and stable detection. But in the extreme light environment, the quality of the frame image can be poor. The event camera can be used to fix the problem because of its high dynamic range.

The problem tackled in this paper is the processing of image $\mathbf{I}_{t_k}$ and the event data $\mathbf{e}_{t_k} = [e_1, e_2, ..., e_m]$ into a set of estimated tracks of objects $\hat{\mathbf{X}}_k = [\boldsymbol{x}_{t_k}^1, \boldsymbol{x}_{t_k}^2, ..., \boldsymbol{x}_{t_k}^n]$, as shown in equation (1),

$$\begin{cases} \mathbf{I}_0, \mathbf{I}_1, ..., \mathbf{I}_{t_k} \\ \mathbf{e}_0, \mathbf{e}_1, ..., \mathbf{e}_{t_k} \end{cases} \quad \Longrightarrow \quad \hat{\mathbf{X}}_\mathbf{0}, \hat{\mathbf{X}}_\mathbf{1}, ... \hat{\mathbf{X}}_\mathbf{k} \tag{1}$$

here $\mathbf{I}_{t_k}$ denotes the frame image at time t_k, $\mathbf{e}_{t_k}$ represents a group of event data within the temporal window $[t_k - \Delta t, t_k + \Delta t]$ of time t_k. The object is represented by the state vector:

$$\boldsymbol{x} = [x, y, \dot{x}, \dot{y}, w, h] \tag{2}$$

where (x, y) is the location of the object, $(\dot{x}, \dot{y})$ is the speed of object. (w, h) is the width and height of the object. Correspondingly, the measurement vector is:

$$\boldsymbol{z} = [x_z, y_z, w_z, h_z, conf] \tag{3}$$

where (x_z, y_z) is the measured location of the object which is the top-left point of the bounding box, (w_z, h_z) is the width and height of the bounding box, and $conf$ is the confidence indicator of the measurement.

For the frame image, the detection is in the form of a bounding box $\boldsymbol{b} = [x_1, y_1, w, h]$, and the top-left point $p = (x_1, y_1)$ is used as the object location. For the event-based camera, the detection is in the form of the clusters $c = \{e_1, e_2, ..., e_k\}$. There are two type of sensors in our framework, the fusion strategy is required to generate the final enhanced measurements $\mathbf{Z}_k$, where each $z_k^i \in \mathbf{Z}_k$ is a measured object. After that, the PMBM filter will take measurement $\mathbf{Z}_k$ as input and output the tracks $\hat{\mathbf{X}}_k$.

3 Object Detection

In this section, we describe how to generate object detection with event data. We first describe the optical flow algorithm based on the event camera, and then we propose the STF-DBSCAN method to cluster the event data into detection based on the event data and its corresponding optical flow. For the object detection with frame image, we use RetinaNet to generate bounding box as detection, and the details are in [20].

3.1 Optical Flow for the Event Camera

For the optical flow algorithm of event data, we choose three states of art algorithms: DAViS Camera Optical Flow [8], EV-FlowNet [7], Spike-FlowNet [11]. The EKLT [13] is not chosen here is because they use the gray image for the feature point extraction and cannot provide dense optical flow.

The DAViS Camera Optical Flow algorithm uses a conventional optical flow equation (4) to compute the motion field, it calculates the temporal and spatial gradients from event data and frame image, respectively.

$$\nabla \boldsymbol{I}\left((x, y), t\right) \boldsymbol{V}((x, y), t) + \boldsymbol{I}_t((x, y), t) \approx 0 \tag{4}$$

Here, the $\nabla \boldsymbol{I}((x, y), t)$ denotes the spatial gradient and is calculated by the frame image. $\boldsymbol{I}_t((x, y), t)$ is the temporal gradient that can be obtained from the event data.

The EV-FlowNet is the event version of the FlowNet [21], they accumulate the event data into a four-channel image and use the deep neural network to predict the optical flow. The structure of the EV-FLownet is shown in Fig. 2. The network uses an encoder-decoder style and predict the flow for each pixel. Note that the event data is asynchronous data, accumulating it into an matrix format will lose its origin characteristics. The Spike-FlowNet tries to solve this problem by replacing the 'encoder' part of the EV-FlowNet with Spike neural network and achieve better performance.

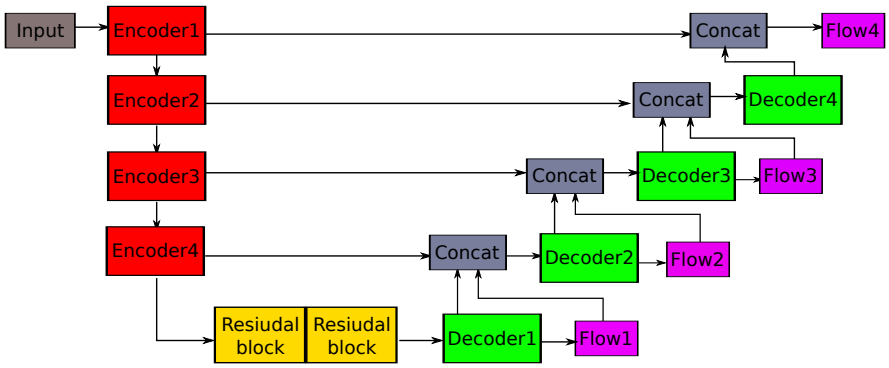


Fig. 2. The structure of the EV-Flownet.

3.2 Clustering Algorithm

In this paper, we use a clustering algorithm to directly generate the detection, because the clustering algorithm is an intuitive and straightforward method for the event points, and the objective here is to distinguish the event points between the background and foreground.

ST-DBSCAN is an extension to DBSCAN for clustering spatial-temporal data [22], it uses two distance parameters (ϵ_s, ϵ_t) to measure the similarity of data according to their spatial and temporal attribute. The raw event data is just spatial-temporal data that can use the ST-DBSCAN. However, it is not enough only considering the spatial-temporal information. Suppose there are two objects that stand closely and move in a different direction simultaneously, the optical flow is different between the two objects but the ST-DBSCAN will consider the two objects as one object. Therefore, we propose the STF-DBSCAN as an extension of ST-DBSCAN to solve this problem.

In order to support three dimensions, we use $(\epsilon_s, \epsilon_t, \epsilon_{of})$ to measure the similarity of spatial, temporal and optical flow, respectively. For event data we have (x, y, u, v, t), which denotes the location, optical flow, and timestamp respectively. Two points will be considered as neighbours only when the three

conditions are met at the same time:

$$\begin{aligned} &\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2)} < \epsilon_s \\ &|(t_1 - t_2)| < \epsilon_t \\ &\sqrt{(u_1 - u_2)^2 + (v_1 - v_2)^2} < \epsilon_{of} \end{aligned} \tag{5}$$

The algorithm of our STF-DBSCAN is shown in algorithm 1. The input of the algorithm is a set of event data and their optical flow: $\mathbf{e}_k = \{e_1, e_2, ..., e_m\}$, where $e_m = (x, y, t, u, v)$. (x, y) is the location of the event, t is the timestamp and (u, v) is the optical flow of the event. The parameters of the algorithm are $(\epsilon_s, \epsilon_t, \epsilon_{of}, minPts, \Delta\epsilon)$. The $minPts$ is the required minimum number of points that a new cluster can be created and the $\Delta\epsilon$ is the threshold value to merge a cluster. The output is the label for all the event points. First, the algorithm will pick one point and find all the neighbours of this point. If the number of neighbour points exceed the threshold $minPts$, this point is a core point and a cluster is formed. For the newly created cluster, we will expand it using a chain rules. If there is another core point in the cluster, we will try merge the new cluster into the previous cluster as shown in line 12-23 of algorithm 1.

In the next section, we will introduce the fusion strategy between the detection of event camera and frame camera, the details about the PMBM filter will also be discussed.

4 Multi-Object Tracking

In this section, we will describe the fusion strategy for the detection of the event camera and the frame camera. Then the motion model and initialization strategy of the PMBM filter is discussed.

4.1 Fusion strategy

The fusion strategy is shown in algorithm 2. The inputs are the bounding boxes $\mathcal{B}_k$ of the frame image and the clusters C_k from event data. The output is the enhanced measurement set $\mathbf{Z}_k$. First, we generate the bounding box $\mathcal{B}_{ek}$ for the clusters C_k. After that, we calculate the intersection over union (IoU) of each pair of the bounding boxes between the $\mathcal{B}_{ek}$ and $\mathcal{B}_k$. If the IoU is greater than the threshold ϵ, then the two bounding boxes are paired and the two measurements are merged.

In order to reduce the noise detection, the enhanced measurements need a label to indicate the source of detection. If the detection is from the frame camera and the event camera together, which means the detection from the two sensors are paired, then the measurement will be labeled with '2'. The detection of the frame camera will be labeled with '1' and the detection from the event camera is considered as 'potential missed detection', and marked with '0'. Next, we will discuss the details about the implementation of PMBM filter given the enhanced measurement.

Algorithm 1 STF-DBSCAN

Input: $\mathbf{e}_k, \epsilon_s, \epsilon_t, \epsilon_{of}, minPts, \Delta\epsilon$
Output: labels

for e_m in $\mathbf{e}_k$ **do**
  if $label_m$ undefined **then**
    $Neighbors = find\_neighbor(e_m, \epsilon_s, \epsilon_t, \epsilon_{of})$
    if $|Neighbors| < minPts$ **then**
      $label_m = noise$
      continue
    end if
    $label_m \leftarrow$ next cluster label
    for p in $Neighbors$ **do**
      $label_p = label_m$
    end for
    $cluster =$ push($point$ in $Neighbors$)
    while $cluster$ is not empty **do**
      $point = cluster.pop()$
      $Neighbors_p = find\_neighbor(point, \epsilon_s, \epsilon_t, \epsilon_{of})$
      if $|Neighbors_p| > minPts$ **then**
        for q in $Neighbors_p$ **do**
          if ($label_q \neq noise$ or $label_q$) undefined and $distance(cluster_Avg, q) < \Delta\epsilon$ **then**
            $label_q = label_m$
          end if
        end for
      end if
    end while
  end if
end for

Algorithm 2 Fusion clusters and Detections

Input: $C_k, \mathcal{B}_k$
Output: Enhanced measurements $\mathbf{Z}_k$

Generate bounding box matrix $\mathcal{B}_{ek}$ for event clusters C_k
for $\boldsymbol{b}$ in $\mathcal{B}_k$, $\boldsymbol{b_e}$ in $\mathcal{B}_{ek}$ **do**
  if $IoU(\boldsymbol{b}, \boldsymbol{b_e}) \geq \epsilon$ **then**
    $\mathbf{Z}_k(\cdot) = [x, y, w, h, 2]$
    $\mathcal{B}_{ek}.pop(\boldsymbol{b_e})$
    $\mathcal{B}_k.pop(\boldsymbol{b})$
  end if
end for
for $\boldsymbol{b}$ in $\mathcal{B}_k$ **do**
  $\mathbf{Z}_k(\cdot) = [x, y, w, h, 1]$
end for
for $\boldsymbol{b_e}$ in $\mathcal{B}_{ek}$ **do**
  $\mathbf{Z}_k(\cdot) = [x, y, w, h, 0]$
end for

4.2 PMBM filter

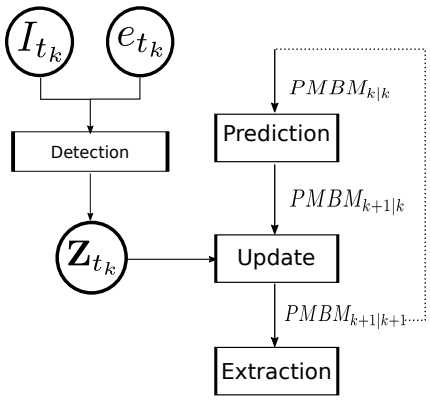


Fig. 3. Overview of the PMBM filter.

The PMBM filter [16] is a multi-object tracking algorithm based on the Bayesian filter framework, which consists of the recursion of predict and update steps. The framework of the PMBM filter is shown in the Fig. 3. Given the PMBM density $PMBM_{k|k}(\mathbf{X}_k)$ at time step k, the PMBM filter can be written as:

$$\begin{aligned} &PMBM_{k+1|k}(\mathbf{X}_k) = \int f(\mathbf{X}_{k+1}|\mathbf{X}_k) PMBM_{k|k}(\mathbf{X}_k)\delta\mathbf{X}_k \\ &PMBM_{k+1|k+1}(\mathbf{X}_{k+1}) \\ &\qquad = \frac{p(\mathbf{Z}_{k+1}|\mathbf{X}_{k+1}) PMBM_{k+1|k}(\mathbf{X}_k)}{\int p(\mathbf{Z}_{k+1}|\mathbf{X}_k) PMBM_{k|k}(\mathbf{X}_k)\delta\mathbf{X}_k} \end{aligned} \tag{6}$$

Here, $f(\mathbf{X}_{k+1}|\mathbf{X}_k)$ is the motion model, and $p(\mathbf{Z}_{k+1}|\mathbf{X}_{k+1})$ is measurement model. In our approach, we use constant velocity motion model. Given the object state described in (2), we can define the motion model as:

$$\begin{aligned} x_t &= x_{t-1} + dt * \dot{x}_{t-1} \\ y_t &= y_{t-1} + dt * \dot{y}_{t-1} \\ \dot{x}_t &= \dot{x}_{t-1} \\ \dot{y}_t &= \dot{y}_{t-1} \\ w_t &= w_{t-1} \\ h_t &= h_{t-1} \end{aligned} \tag{7}$$

The multi-object tracking problem also has to resolve problems like initialize the object birth $PMBM_{0|0}$. Normally, the PMBM filter will consider all the measurement that fails to associate with the current tracks as birth objects.

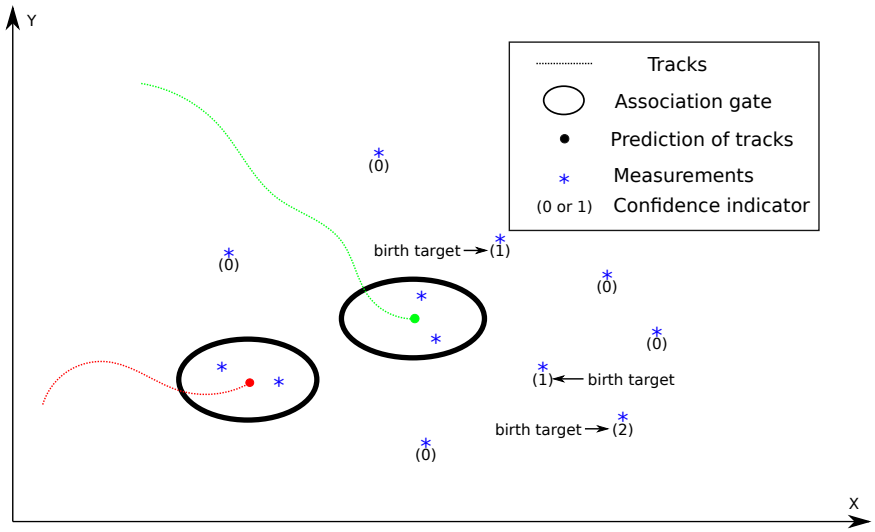


Fig. 4. The birth strategy of our approach.

However, in our framework, the detection is from two sensors, too much birth target will increase clusters and reduce the tracking performance. So we adopt the adaptive birth intensity strategy by using the confidence indicator in the measurement. Only the measurement that has a positive confidence indicator can be seen as a birth target. The strategy is shown in Fig. 4. For those measurements that fail to be associated with current tracks, only the one that has a positive confidence indicator can be used as a birth target.

5 Experiment

The evaluation contains three parts: optical flow, detection, and tracking. We first make a comparison between the three optical flow algorithms. And then, we test our clustering algorithm and the object tracking framework with MOT Challenge15 (MOT15) dataset [23]. We convert the dataset into the event-based camera version with the event camera simulator [24]. After that, we test the approach with our experiment vehicle and DAVIS346 camera to examine the feasibility of our approach in the real environment.

5.1 Optical Flow

We test the three optical flow algorithms with the Multi-Vehicle Stereo Event Camera (MVSEC) dataset [7]. This dataset contains event data for a collection of environments (e.g., indoor flying and outdoor driving) and also provides the

corresponding ground truth optical flow. The metric we used for the optical flow is Average End-point Error (AEE), it is given by:

$$AEE = \frac{1}{m} \sum_{m} || (u, v)_{pred} - (u, v)_{gt} ||_2 \tag{8}$$

The results are shown in the table 1. All three algorithms can provide accurate and stable optical flow. In our approach, we choose the DAViS flow as the optical flow module. Because the DAViS flow calculates the optical flow event by event and is more fit to our clustering approach. Also, the two deep learning algorithms require more computation time and extra event encoding calculation.

Table 1. Average Endpoint Error (AEE) ↓ for the three optical flow algorithms

	indoor1	indoor2	indoor3	outdoor1
EV-FlowNet	1.03	1.72	1.53	**0.49**
Spike-FlowNet	**0.84**	**1.28**	**1.11**	**0.49**
Davis Flow	1.11	1.89	1.64	1.01

5.2 Detection Rate

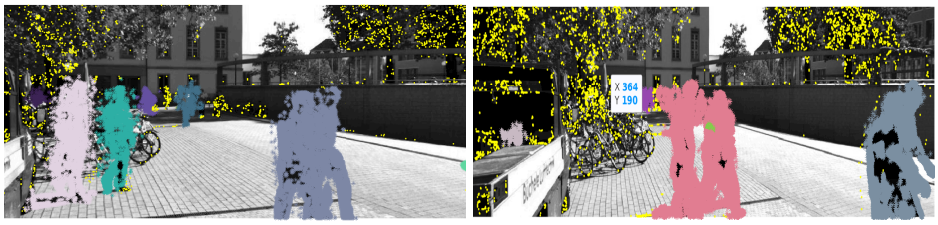
For the clustering algorithm, the detection rate is used to evaluate the detection performance [25]. We first test our approach with the converted MOT15 dataset. Then we test our approach with three challenging event-based camera datasets: EED [26], MOD [3], EV-IMO [27], where EED is a real dataset in extreme light conditions; MOD is a synthetic dataset designed for training the neural network of object detection; EV-IMO is real dataset focus on the fast camera motion and rich texture surface.

Table 2. Comparison of the Detection Rate

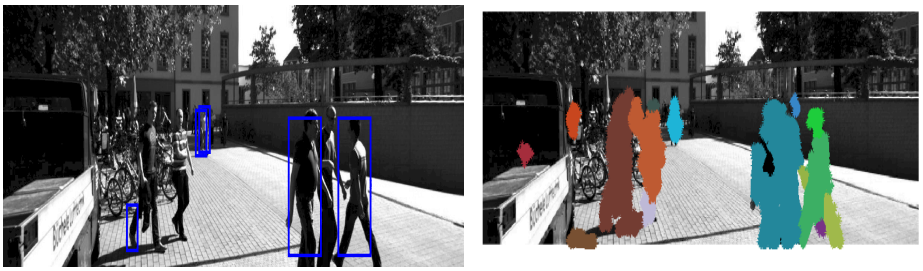
Methods	Detection rate ↑ for dataset (%)			
	MOT15	EED	MOD	EV-IMO
k-means	64.34	61.46	36.83	42.59
DBSCAN	79.78	80.91	40.15	45.37
DAViS Flow + k-means	74.05	85.49	46.83	**55.73**
Mitrokhin et al. [26]	-	88.93	**70.12**	48.79
Stoffregen et al. [25]	-	**93.17**	-	-
Ours	**80.15**	92.32	64.36	48.82

The quantitative is shown in table 2. The k-means and DBSCAN represent the two cluster algorithms that use the position information only. DAViS Flow

(a) Miss Detection of frame image



(b) Detection of event data



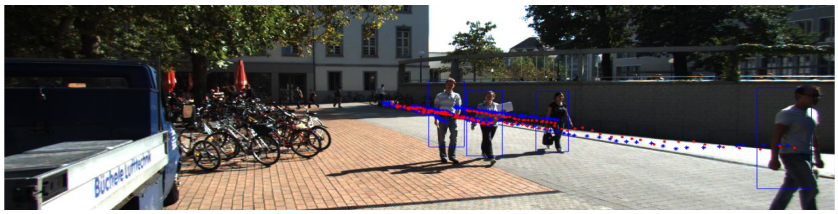
(c) Detection of the dense pedestrians

Fig. 5. Detection evaluation based on the event data.

+ k-means represents the k-means algorithm that uses optical flow and position information. For the three challenging event-based camera datasets. The results show that our approach has a better performance than the other clustering algorithms. Compare to the [26, 25], our approach has similar performance. But our approach requires fewer accumulated events and is naturally synchronized with the frame image. So, our detection method is more suitable for hybridizing the event camera and the frame camera. Our approach's performance in extreme light scenarios (EED) and the MOT15 is good. But for the scenario that is full of texture and has fast camera motion (MOD, EV-IMO), our approach still needs to be improved. The performance is unsatisfactory under two circumstances: 1. fast self-motion of the event-based camera. 2. the object has no visible movement relative to the camera. The qualitative results are shown in Fig. 5. Fig. 5(a) is

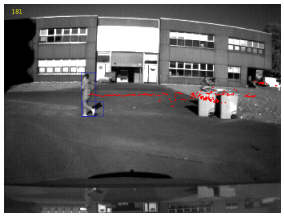
the detection of the frame image and detection is shown in blue bounding boxes. Fig. 5(b) is the clustering results of our approach. The clusters are represented in different colors and the yellow points are those event data that belong to the background. The results show that when the detection from frame camera is missing, the event data can provide backup detection.

5.3 Multi-object Tracking Performance

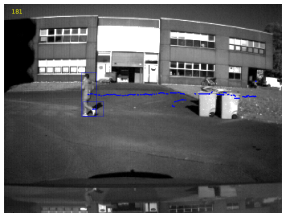


(a) Track visualization of our approach

(b) Autonomous vehicle



(c) Our approach



(d) Without event camera

Fig. 6. Visualization of our tracking approach. (a) is the visualization of the estimation and ground truth tracks for KITTI-17, the red point is the estimated trajectory, and the blue star is the ground truth. (b)-(d) Visualization of our tracking approach in the real environment. The person is running from right to left in front of the vehicle, the tracking with frame camera is in blue and the track of our approach is red.

For the object tracking evaluation, we compare our approach with the PMBM filter that only uses the detection of frame images. We use the HOTA (Higher Order Tracking Accuracy) metric described in [28]. This metric balances the effect of performing accurate detection, localization, and association into a single unified metric: HOTA. The quantitative results of our approach are shown in the table 3. The results show that our approach has better overall performance then tracking with frame camera only. But the association accuracy (AssA) and detection recall (DetRe) of our approach is slightly worse, because the event camera increases the noise of the detection.

The evaluation we have done is based on the event camera simulator, all the data is synthetic by doing interpolation between two frame images. And their spatial resolution is different from the real event camera. In order to examine our approach is also feasible with the real event camera and vehicle, we also test

Table 3. Comparison of Object tracking results

Methods	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr	LocA
Our approach	**31.21**	**48.54**	20.39	**64.82**	54.63	**21.92**	**73.45**	**75.40**
frame image only	28.96	41.72	**20.57**	48.49	**60.22**	21.44	72.98	74.90

the approach on the experiment vehicle. Fig. 6 is the tracking results of a simple scenario: the experiment vehicle is moving forwards slowly while a pedestrian is running across the road. The camera we use is DAVIS346. Fig. 6c is the track of the person based on our approach, Fig. 6d is the track based on the PMBM filter and frame camera. When the pedestrian passes the trash bin, the object is lost due to the interfere of the trash bin. But our approach fixes this problem with the detection from the event-based camera.

6 Conclusion

This paper presented a novel approach that realizes object tracking based on a hybrid of event camera and frame camera. The approach exploits the temporal information by calculating the optical flow with the event data and generating clusters according to their position, location, and optical flow. Then the clusters are combined with detection from the frame image and fed into the PMBM filter. Our approach utilizes the advantages of the event camera to achieve better tracking performance by providing complementary detection for the frame camera. Because a clustering algorithm is adopted, our method has limitations in the crowded object and rich texture background environment.

In the future, improving the object detection algorithm for the event camera under rich texture and crowded environment is a perspective direction. We will also specifically record datasets for the scenario with motion blur and extreme light conditions where the frame camera fails with the event camera, so we can develop more robust detection algorithm with the datasets.

References

1. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. IEEE Journal of Solid-State Circuits 43(2), 566–576 (2008)
2. Brandli, C., Berner, R., Yang, M., Liu, S.C., Delbruck, T.: A 240 × 180 130 db 3 µs latency global shutter spatiotemporal vision sensor. IEEE Journal of Solid-State Circuits 49(10), 2333–2341 (2014)
3. Sanket, N., Parameshwara, C., Singh, C., Kuruttukulam, A.V., Fermüller, C., Scaramuzza, D., Aloimonos, Y.: Evdodgenet: Deep dynamic obstacle dodging with event cameras. pp. 10651–10657 (05 2020)
4. Delbruck, T., Lang, M.: Robotic goalie with 3 ms reaction time at 4% cpu load using event-based dynamic vision sensor. Frontiers in Neuroscience 7, 223 (2013), https://www.frontiersin.org/article/10.3389/fnins.2013.00223

5. Delbruck, T.: Neuromorophic vision sensing and processing. In: 2016 46th European Solid-State Device Research Conference (ESSDERC). pp. 7–14 (2016)
6. Litzenberger, M., Kohn, B., Belbachir, A., Donath, N., Gritsch, G., Garn, H., Posch, C., Schraml, S.: Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor. In: 2006 IEEE Intelligent Transportation Systems Conference. pp. 653–658 (2006)
7. Zhu, A., Yuan, L., Chaney, K., Daniilidis, K.: Ev-flownet: Self-supervised optical flow estimation for event-based cameras. In: Proceedings of Robotics: Science and Systems. Pittsburgh, Pennsylvania (June 2018)
8. Almatrafi, M., Hirakawa, K.: Davis camera optical flow. IEEE Transactions on Computational Imaging 6, 396–407 (2020)
9. Maqueda, A., Loquercio, A., Gallego, G., Garcia, N., Scaramuzza, D.: Event-based vision meets deep learning on steering prediction for self-driving cars. pp. 5419–5427 (06 2018)
10. Binas, J., Neil, D., Liu, S.C., Delbruck, T.: Ddd17: End-to-end davis driving dataset (11 2017)
11. Lee, C., Kosta, A., Zhu, A.Z., Chaney, K., Daniilidis, K., Roy, K.: Spike-flownet: Event-based optical flow estimation with energy-efficient hybrid neural networks. In: European Conference on Computer Vision. pp. 366–382. Springer (2020)
12. Alzugaray, I., Chli, M.: Asynchronous multi-hypothesis tracking of features with event cameras. In: 2019 International Conference on 3D Vision (3DV). pp. 269–278 (2019)
13. Gehrig, D., Rebecq, H., Gallego, G., Scaramuzza, D.: EKLT: Asynchronous, photometric feature tracking using events and frames. Int. J. Comput. Vis. (2019)
14. Scheidegger, S., Benjaminsson, J., Rosenberg, E., Krishnan, A., Granström, K.: Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering. pp. 433–440 (06 2018)
15. Mahler, R.P.: Advances in Statistical Multisource-multitarget Information Fusion. ARTECH House, Boston (2014)
16. Garcia-Fernandez, A.F., Williams, J.L., Granström, K., Svensson, L.: Poisson multi-bernoulli mixture filter: Direct derivation and implementation. IEEE Transactions on Aerospace and Electronic Systems 54(4), 1883–1901 (2018)
17. Vo, B.N., Vo, B.T., Beard, M.: Multi-sensor multi-object tracking with the generalized labeled multi-bernoulli filter. IEEE Transactions on Signal Processing 67(23), 5952–5967 (2019)
18. Xia, Y., Granstrcom, K., Svensson, L., Garcia-Fernandez, A.F.: Performance evaluation of multi-bernoulli conjugate priors for multi-target filtering. In: 2017 20th International Conference on Information Fusion (Fusion). pp. 1–8 (2017)
19. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP). pp. 3645–3649 (2017)
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 42(2), 318–327 (2020)
21. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (December 2015)
22. Birant, D., Kut, A.: St-dbscan: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering 60(1), 208–221 (2007), https://www.sciencedirect.com/science/article/pii/S0169023X06000218, intelligent Data Mining

23. Leal-Taixé, L., Milan, A., Reid, I., Roth, S.: Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942 abs/1504.01942 (04 2015)
24. Rebecq, H., Gehrig, D., Scaramuzza, D.: ESIM: an open event camera simulator. Conf. on Robotics Learning (CoRL) (Oct 2018)
25. Stoffregen, T., Gallego, G., Drummond, T., Kleeman, L., Scaramuzza, D.: Event-based motion segmentation by motion compensation. pp. 7243–7252 (10 2019)
26. Mitrokhin, A., Fermüller, C., Parameshwara, C., Aloimonos, Y.: Event-based moving object detection and tracking. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1–9 (2018)
27. Mitrokhin, A., Ye, C., Fermüller, C., Aloimonos, Y., Delbruck, T.: Ev-imo: Motion segmentation dataset and learning pipeline for event cameras. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 6105–6112 (2019)
28. Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., Leibe, B.: Hota: A higher order metric for evaluating multi-object tracking. International Journal of Computer Vision pp. 1–31 (2020)