



## Research Article

# Neuromorphic Vision Based Multivehicle Detection and Tracking for Intelligent Transportation System

**Guang Chen** <sup>1,2</sup> **Hu Cao**,<sup>3</sup> **Muhammad Aafaque**,<sup>2</sup> **Jieneng Chen**,<sup>4</sup> **Canbo Ye**,<sup>1</sup> **Florian Röhrbein**,<sup>2</sup> **Jörg Conradt**,<sup>5</sup> **Kai Chen**,<sup>1</sup> **Zhenshan Bing**,<sup>2</sup> **Xingbo Liu**,<sup>1</sup> **Gereon Hinz**,<sup>2</sup> **Walter Stechele** <sup>6</sup> and **Alois Knoll**<sup>2</sup>

<sup>1</sup>College of Automotive Engineering, Tongji University, China

<sup>2</sup>Robotics and Embedded Systems, Technische Universität München, Germany

<sup>3</sup>State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, China

<sup>4</sup>College of Electronics and Information Engineering, Tongji University, China

<sup>5</sup>Department of Computational Science and Technology, KTH Royal Institute of Technology, Sweden

<sup>6</sup>Integrated Systems, Technische Universität München, Germany

Correspondence should be addressed to Guang Chen; [guang@in.tum.de](mailto:guang@in.tum.de)

Received 10 August 2018; Revised 1 October 2018; Accepted 6 November 2018; Published 2 December 2018

Academic Editor: Krzysztof Okarma

Copyright © 2018 Guang Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Neuromorphic vision sensor is a new passive sensing modality and a frameless sensor with a number of advantages over traditional cameras. Instead of wastefully sending entire images at fixed frame rate, neuromorphic vision sensor only transmits the local pixel-level changes caused by the movement in a scene *at the time they occur*. This results in advantageous characteristics, in terms of low energy consumption, high dynamic range, sparse event stream, and low response latency, which can be very useful in intelligent perception systems for modern intelligent transportation system (ITS) that requires efficient wireless data communication and low power embedded computing resources. In this paper, we propose the first neuromorphic vision based multivehicle detection and tracking system in ITS. The performance of the system is evaluated with a dataset recorded by a neuromorphic vision sensor mounted on a highway bridge. We performed a preliminary multivehicle tracking-by-clustering study using three classical clustering approaches and four tracking approaches. Our experiment results indicate that, by making full use of the low latency and sparse event stream, we could easily integrate an online tracking-by-clustering system running at a high frame rate, which far exceeds the real-time capabilities of traditional frame-based cameras. If the accuracy is prioritized, the tracking task can also be performed robustly at a relatively high rate with different combinations of algorithms. We also provide our dataset and evaluation approaches serving as the first neuromorphic benchmark in ITS and hopefully can motivate further research on neuromorphic vision sensors for ITS solutions.

## 1. Introduction

Neuromorphic vision sensors, inspired by biological vision, use an event-driven frameless approach to capture transients in visual scenes. In contrast to conventional cameras, neuromorphic vision sensors only transmit local pixel-level changes (called “events”) caused by movement in a scene *at the time of occurrence* and provide an information rich stream of events with a latency within tens of microseconds. To be specific, a single event is a tuple  $(t, x, y, p)$ , where  $x, y$  are the

pixel coordinates of the event in 2D space,  $t$  is the time-stamp of the event, and  $p$  is the polarity of the event, which is the sign of the brightness change (increasing or decreasing). Furthermore, the requirements for data storage and computational resources are drastically reduced due to the sparse nature of the event stream. Apart from the low latency and high storage efficiency, neuromorphic vision sensors also enjoy a high dynamic range of 120 dB. In combination, these properties of neuromorphic vision sensors inspire entirely new designs of intelligent transportation systems. In order

to elucidate the mechanism of neuromorphic sensors more clearly, a comparison between standard frame-based cameras and neuromorphic vision sensors is shown in Figure 1.

Traditionally, frame-based vision sensors serve as the main information sources for vision perception tasks of ITS, which results in well-known challenges, such as the limited real-time performance and substantial computational costs. The key problem lies in the fact that conventional cameras sample their environment with a fixed frequency and produce a series of frames, which actually contain enormous amounts of redundant information but lost all the information between two adjacent frames. Hence, traditional vision sensors waste memory access, energy, computational power, and time on the one hand and also discard significant information between continuous frames on the other hand. These properties bring about great limitations on its applications. For an intelligent transportation system equipped with conventional cameras, appearance feature extraction based on learning methods is the major strategy of environment perception tasks, which is acknowledged to be computationally demanding [3]. Moreover, in order to get good detection and tracking performance, large amounts of labeled data as well as dedicated and expensive hardware such as GPU are indispensable for the training and learning process.

In this paper, a novel approach for the tracking system of the intelligent transportation systems (ITS) is proposed based on the neuromorphic vision sensor. And we will publish our dataset and evaluation approaches as well, aiming to provide the first neuromorphic benchmark in ITS and motivate further research on neuromorphic vision sensors for ITS solutions. To fully demonstrate the feasibility and potential of the approach, different detection and tracking algorithms are presented and compared in this paper. In detection stage, we utilize and evaluate three classical clustering approaches: mean-shift clustering (MeanShift) [4], density based spatial clustering of applications with noise (DBSCAN) [5], and WaveCluster [6]. In terms of tracking stage, we carry out online multitarget tracking via four different algorithms: simple online and real-time tracking (SORT) [7], the Gaussian mixture probability hypothesis density filter (GM-PHD) [8], the cardinalized probability hypothesis density filter (GM-CPHD) [9], and probabilistic data association filter (PDAF) [10].

In combination, we propose the first neuromorphic vision based multivehicle detection and tracking system in ITS, with the unique properties of neuromorphic vision sensors mentioned above. The performance of the system is evaluated with a dataset recorded by a neuromorphic vision sensor mounted on a highway bridge. According to the experiment results, the tracking-by-clustering system can run at a rate of more than 110Hz, which far exceeds the real-time performance of traditional frame-based cameras. With priority given to accuracy, the tracking task can also be performed more robustly and precisely using different algorithm combinations. This work is extended from a conference paper which is published on the Joint German/Austrian Conference on Artificial Intelligence, 2017 [11]. We extended it from 4 aspects. First, we extend the testing data sequences

to 3 sequences for the experiment section. Second, we evaluate 3 detection-by-clustering approaches instead of 2 in [11]. Third, we extend to evaluate 4 tracking approaches instead of 1 in [11]. Finally, based on these differences, we analyze the results in different views with new outcomes [11].

The rest of this paper is organized as follows. In Section 2, we list the related work in the context of previous multivehicle detection and tracking methods. In Section 3, we introduce the variety of neuromorphic vision sensors and dataset. The algorithms utilized for detection and tracking are illustrated, respectively, in Section 4. The experiment results are analyzed and discussed in Section 5. In Section 6, we draw the conclusion and point out the possible further work.

## 2. Related Work

In the past decade, detecting and tracking multiple vehicles in traffic scenes for traffic surveillance, traffic control, and road traffic information systems is an emerging research area for intelligent transport systems [12–15]. Most of the existing vehicle tracking systems are based on the video cameras [16]. Previous approaches of vision based multiple vehicles detection and tracking could be subdivided into four categories: frames difference and motion based methods [17–19], background subtraction methods [15, 20], and feature based methods [21, 22]. Meanwhile, a few camera-based datasets for vehicle detection and tracking come to light in recent years [23–25], which promote the research for ITS.

All previous multivehicle detection and tracking methods leverage images acquired by traditional frame-based cameras. Conventional cameras may suffer from various motion-related issues (motion blur, rolling shutter, etc.) which may impact performance for high-speed vehicles detection and tracking. Neuromorphic vision sensors are widely applied to robotics [26–29] and vehicles [30–32]. A few relevant neuromorphic vision datasets [33, 34] have been released in recent years, which facilitate the neuromorphic vision application for object detection and tracking. Recent years also witness the various applications for detection and tracking tasks with neuromorphic vision sensor such as feature tracking [35, 36], line tracking [37], and microparticle tracking [38].

However, there is still a lack of neuromorphic datasets and relevant applications with neuromorphic vision sensors in intelligent transport system, albeit such sensors instinctively enjoy superiority in high-speed motion recording, which can correspondingly facilitates the high-speed multiple vehicle detection and tracking in ITS systems. Thus, it is meaningful to apply neuromorphic vision techniques to ITS systems.

## 3. Neuromorphic Vision Sensor and Dataset

**3.1. Neuromorphic Vision Sensor.** A short description of different versions of neuromorphic vision sensors is provided in this section which is also mentioned in [11]. The purpose is to encourage researchers who are not familiar with neuromorphic vision sensors to explore the potential applications

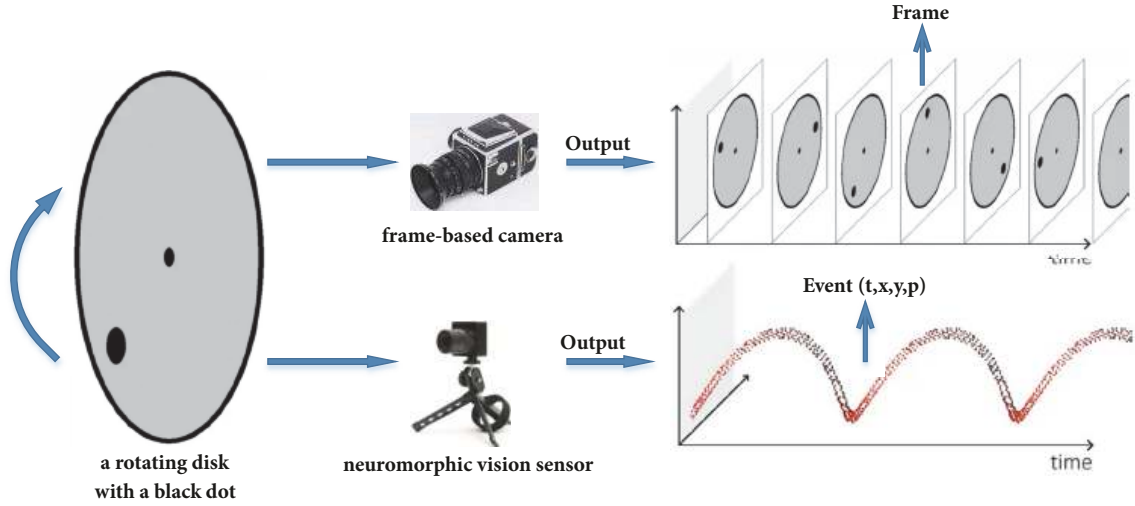


FIGURE 1: Visualization of the output from a neuromorphic vision sensor and a standard frame-based camera when facing a rotating disk with a black dot. Comparing to conventional frame-based camera which transmitted complete images at fixed latency, the neuromorphic vision sensor [1] emitted events individually and asynchronously at the time they occur. This figure is adopted from [2].

in the intelligent system. Figure 2 shows different versions of neuromorphic vision sensors.

**Dynamic Vision Sensor (DVS).** Dynamic Vision Sensors (DVS) are a new generation of cameras that are sensitive to intensity change, more specifically, to intensity logarithmic change. A DVS pixel typically generates one to four events (spikes) when an edge crosses it. DVS output consists of a continuous flow of events (spikes) in time, each with submicrosecond time resolution, representing the observed moving reality as it changes, without waiting to assemble or scan artificial time-constrained frames (images).

**Embedded Dynamic Vision Sensor (eDVS).** For embedded systems in mobile robotics such as unmanned aerial vehicle, an USB interface to transmit raw events is not desirable, nor is a desktop PC for event processing acceptable. For this purpose, a small embedded DVS (eDVS) is developed consisted of a DVS chip and a compact 64MHz 32bit microcontroller directly connected to the DVS chip.

**Miniature Embedded Dynamic Vision Sensor (meDVS).** The miniaturized version of the eDVS(meDVS) has minimum size (18cm×18cm) and lightest weight (2.2g) of DVS so far. The typical power consumption is 300mW. The strengths of meDVS make it desirable to any applications on the limited storage, bandwidth, and low latency of the on-board embedded system of the intelligent system.

**Dynamic and Active Pixel Vision Sensor (DAVIS).** In this paper we use a new neuromorphic vision sensor which is named the Dynamic and Active Pixel Vision Sensor (DAVIS) [39]. The model DAVIS240 camera has a higher resolution of 240×180, higher dynamic range, and lower power consumption and allows a concurrent readout of global shutter image frames, which are captured using the same photodiodes as for

the DVS event generation. In this work, we only use the event data.

**3.2. Dataset and Benchmark.** We present a labeled dataset for the evaluation of an online multivehicle detection and tracking system in ITS domain. The raw event data are collected by a neuromorphic vision sensor which is mounted on the bridge in a highway scenario. The neuromorphic vision sensor used in this paper is called dynamic and active pixel sensor (DAVIS) with a model No. DAVIS240C. We have labeled three event sequences in this work. The first event sequence (named EventSeq-Vehicle1) is of length 45.4s having 110.7Mevents and on average contains 2,438Keps (Kilo events per second). The second event sequences (named EventSeq-Vehicle2) is of length 32.4s with 79.4Mevents and on average contains 2,450Keps. The third event sequence (named EventSeq-Vehicle3) is of length 21.8s having 53.4Mevents and on average contains 2,450Keps. The vehicles are moving in both the directions, i.e., towards and away from the camera in multiple lanes. The vehicles in the dataset range from small cars to the trailers and trucks, which makes the dataset diverse and challenging in nature.

We manually annotated all the vehicles' positions and unique identity in three event sequences using the openly available video annotation tool called ViTBAT [40]. For annotation, the video was created from the binary events data file. We accumulated events data into video frames at three different time intervals: 10ms, 20ms, and 30ms. The description and data format of our dataset can be seen from Table 1.

## 4. Online Multitarget Detection and Tracking

We illustrate our multiobject tracking-by-clustering system in this section. In contrast to traditional object detection

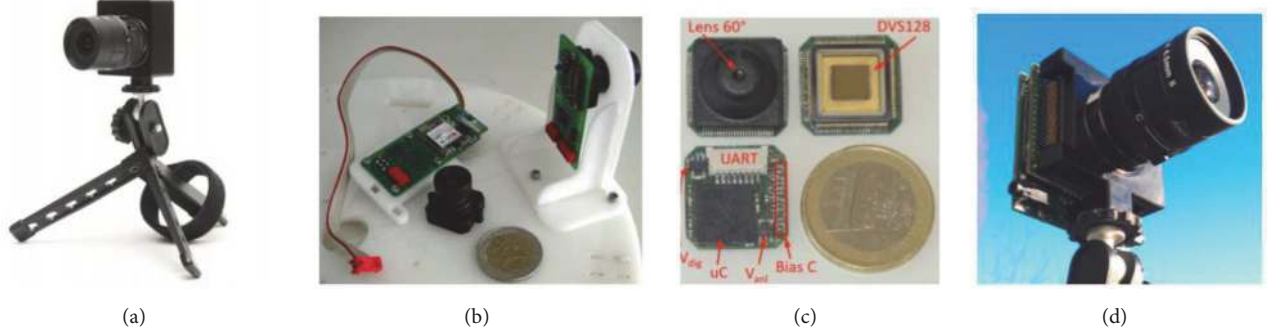


FIGURE 2: Different versions of neuromorphic vision sensors: (a) Dynamic Vision Sensor 128 (DVS128). (b) Embedded Dynamic Vision Sensor (eDVS). (c) Miniature Embedded Dynamic Vision Sensor (meDVS). (d) Dynamic and Active Pixel Vision Sensor (DAVIS). This figure is adopted from [11].

TABLE 1: The standard file format in our benchmark. In event.txt, timestamp recodes the timestamp of each raw event, x and y are the coordinates of the event's origin, and p is the event's polarity. In det.txt, box\_x and box\_y are x and y coordinates of the bounding box and box\_width and box\_height are the width and height of each bounding box. The events which fall into a bounding box from start timestamp to end timestamp are accumulated to a cluster. Each cluster has an ID. The items' definitions in gt.txt and track.txt are similar to det.txt file, except that the gt.txt shows the ground truth and track.txt shows the result of tracking.

File name	Description	Format
event.txt	One event per line	(timestamp, x, y, p)
det.txt	One detection measurement per line	(starttimestamp, endtimestamp, box_x, box_y, box_width, box_height)
gt.txt	One ground-truth measurement per line	(starttimestamp, endtimestamp, ID, box_x, box_y, box_width, box_height)
track.txt	One tracking measurement per line	(starttimestamp, endtimestamp, ID, box_x, box_y, box_width, box_height)

approaches, we generate our object hypothesis directly from the measurements with a classic clustering method. The advantage is that we can skip the background modeling step (dynamic foreground segmentation), as most events transmitted by the dynamic vision sensor are generated by dynamic objects. In order to estimate the states of the actual objects, we integrate an online multitarget tracking method into our system. It is our opinion that only highly effective and online tracking methodology can take full advantage of neuromorphic vision cameras.

**4.1. Vehicle Detection by Clustering.** As neuromorphic sensors only transmit relative light intensity changes for each pixel, methods using appearance features, such as color and texture as input, cannot be utilized. Clustering methods, on the contrary, are very suitable for this situation. Hence we present three different clustering algorithms in this section, which do not depend upon the prior knowledge of the number and shape of the clusters. In addition, only dynamic information in the form of sparse streams of asynchronous time-stamped events can be gained from neuromorphic vision sensors. In order to arrive at a meaningful interpretation and make the most of neuromorphic vision sensors' advantages, it is necessary to accumulate event streams before applying clustering algorithms. We accumulate event data for different time intervals (10ms, 20ms, and 30ms), making it synchronized and more informative, after which three

classic clustering approaches, MeanShift [4], DBSCAN [5] and WaveCluster [6], are carried out and compared. The following subsections illustrate these clustering approaches briefly.

**Detection by MeanShift [11].** Estimation of the gradient of a density function via MeanShift and the iterative mode seeking procedure was developed by Fukunaga and Hostetler in [41]. The mean-shift algorithm has been exploited in low level computer vision tasks, including image segmentation, color space analysis, face tracking, etc., by reason of its properties of data compaction and dimensionality reduction [4]. Specifically, the mean-shift algorithm considers the input as a probability density function and the objective of the algorithm is to find the modes of this function. These modes represent the centers of the discovered clusters. The input points are fed to the kernel density estimation and then the gradient ascent method is applied to the density estimate. The density estimation kernel uses two inputs: the total amount of points and the bandwidth or the size of the window. The main disadvantage of the mean-shift algorithm lies in its iterative nature and difficulty of filtering out noise.

**Detection by DBSCAN [11].** DBSCAN uses density based spatial clustering for applications with noise. For each point  $p$ , the associated density is calculated by counting the number of points in a search area of specified radius,  $Eps$  (the maximum radius of the neighbourhood from point), around



the point. The points with density higher than the specified threshold value, MinPts (the minimum number of points required to form a dense region), are classified as core points while the rest are classified as noncore points. A cluster is yielded if  $p$  is a core point; otherwise, if  $p$  is a border point, then no point is density-reachable from  $p$  and DBSCAN takes the next point from the database [5]. The main advantage of DBSCAN is that it can find the clusters of arbitrary shapes.

*Detection by WaveCluster.* The basic idea of WaveCluster is to quantize the feature space of the image firstly and then apply discrete wavelet transform on it, after which we can find the connected components (clusters) in the subbands of transformed feature space [6]. For best clustering result, the quantization scale as well as the component connection algorithm should be applied according to the raw data. In the context of this paper, the accumulated event data can be regarded as 2-dimensional data. With selected interval  $m$  in each dimension, we can now divide the event data into  $m^2$  grids, and each grid  $i$  contains  $N_i$  data point. Considering the multiresolution property of wavelet transform, different grid sizes can be adopted at different scales of transform. In the second step of WaveCluster algorithm, discrete wavelet transform will be applied on the quantized feature space [6]. Afterwards, a new feature space  $T_k$  is acquired. We can also filter out the noise in  $T_k$  with a selected threshold. With the new set of units  $T_k$ , connected components in the transformed feature space can be detected as clusters. Details of the algorithm can refer to [6].

*4.2. Online Multitarget Tracking.* In order to make full use of the advantages of event data, we have chosen four classic tracking algorithms, which are relatively small in computation and highly effective. Our online multitarget tracking is a simple and standard method which is widely explored in traditional camera-based multiobject tracking [42]. As the event data have no texture information, we use the bounding box overlap as a simple association metric for the data association problem. All these tracking algorithms are briefly described in the following sections.

*Tracking by SORT [11].* We utilize a single hypothesis tracking methodology with standard Kalman filter and data association using Hungarian method [7]. In order to assign detected clusters to existing targets, each target's geometry and image coordinates are estimated by predicting its new state in the current frame. The cost matrix for each detected cluster and each existing target is calculated as the intersection over union distance (IOU). The Hungarian algorithm is used to optimally solve the assignment problem. We also define a minimum IOU to reject assignments where the detected cluster to target cluster overlap is less than the threshold. When a new cluster enters into the camera field of view or when an existing target leaves the camera view, target identities get updated, either by adding new IDs or by according deletion. The same methodology for tracking has been used in this work as presented in [7]. Instead of solving for detection for tracking in a global assignment

problem, we choose an early deletion of lost targets policy, which prevents unbounded growth of the number of trackers.

*Tracking by GM-PHD.* GM-PHD filter is a recursive algorithm which jointly estimates the time-varying number of targets and their states from the observation sets in the presence of data association uncertainty, noise, false alarms, and detection uncertainty. The algorithm models the respective collection of targets and measurements as random finite sets and applies the probability hypothesis density (PHD) recursively for posterior intensity propagation, which is basically the first order-statistic of the random finite set in time. With linear and Gaussian assumptions, the target dynamics and birth process and the posterior intensity at any time step are considered to be Gaussian mixture. The recursions with number of Gaussian components management increase the efficiency. In tracking world, the intensity is also known as probability hypothesis density [8]. The further mathematical insights into the algorithm and its recursive linear Gaussian version can be studied in [8]. As stated in the previous section, the birth model for the targets is chosen to be linear in this work, which also stands for this and upcoming approaches for tracking.

*Tracking by GM-CPHD.* In probability hypothesis density (PHD) filter, the posterior intensity of the random finite set of targets is propagated, recursively. In cardinalized PHD (CPHD) filter, both the posterior intensity and posterior cardinality distribution are propagated jointly, hence making it a generalization of PHD recursion. The accuracy and stability are increased by incorporating the cardinality information [9]. This work is basically the implementation of closed-form solution to CPHD recursion under the assumption of linear Gaussian target dynamics and birth model. The algorithm can also be extended to nonlinear models using linearization and unscented transformation techniques. While comparing with standard PHD filter, CPHD filter not only side steps the need of data association task in conventional tracking methods but also improves the accuracy of the individual target state estimates and the variance of the estimated number of targets [9].

*Tracking by PDAF.* The probabilistic data association filter (PDAF) computes the probabilities for target being tracked for each valid measurement. This measurement origin uncertainty is accounted by this probabilistic or Bayesian information. As the linear models for the targets birth dynamics and measurement equations are assumed, therefore, the developed PDAF algorithm is based on Kalman filter. PDAF works on the validated measurements at the current time and for each measurement, an association probability is calculated for computing the weight of current measurement in a combined innovation. This combined innovation helps in updating the estimation of the state. And finally, the state covariances are updated for computing the measurement origin uncertainty [10]. The detailed mathematical insights into the PDAF algorithm with its extensions can be studied from [10].

TABLE 2: Evaluation metrics for vehicle detection, TP is the total number of true positives, FP is the total number of false positives, and GT indicates ground truth.

Metrics	Better	Perfect	Description
Precision	higher	100%	Ratio of TP / (TP+FP)
Recall	higher	100%	Ratio of correct detections to total number of GT boxes

## 5. Experiments and Results

We evaluate the performance of various tracking-by-clustering implementations on our dataset. The evaluation results are provided by following the standard MOT challenge metrics [43]. We analyze the performance and runtime of the three classical clustering algorithms, as well as the four tracking algorithms for multivehicle tracking-by-clustering task, where stream inputs are accumulated at different intervals (10ms, 20ms, and 30ms time intervals).

**5.1. Metrics.** For performance evaluation, we follow the current evaluation protocols for visual object detection and multiobject tracking. Although these protocols are designed for frame-based vision sensors, they are still suitable for quantitative evaluation of our tracking method. In this work, we accumulate events to frames in different time intervals. In this work we have two evaluation metrics (see Table 2) which are defined in [44].

Since our detection results from clustering methods have no probability score, we are not able to provide the mean precision to summarize the shape of the precision/recall (ROC) curve which is widely adopted in object detection evaluation in computer vision. The evaluation metrics for multivehicle tracking used in this work is defined in [43], well-known as the MOT challenge metrics. Evaluation scripts are available on MOT Challenge official website (<https://motchallenge.net>). More details are as follows:

- (i) MOTA( $\uparrow$ ): Multiple Object Tracking Accuracy. This measure combines three error sources: false positives, missed targets, and identity switches.
- (ii) MOTP( $\uparrow$ ): Multiple Object Tracking Precision. The misalignment between the annotated and the predicted bounding boxes.
- (iii) MT( $\uparrow$ ): mostly tracked targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at least 80% of their respective lifespan.
- (iv) PT( $\uparrow$ ): number of partially tracked trajectory.
- (v) ML( $\downarrow$ ): mostly lost targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at most 20% of their respective lifespan.
- (vi) FP( $\downarrow$ ): the total number of false positives.
- (vii) FN( $\downarrow$ ): the total number of false negatives (missed targets).
- (viii) IDs( $\downarrow$ ): the total number of identity switches.

- (ix) FM( $\downarrow$ ): the total number of times a trajectory is fragmented (i.e., interrupted during tracking).

**5.2. Performance Evaluation.** In this section, we report the performance and runtime of the selected approaches for multivehicle detection and tracking. Firstly, we compare the detection performance of the three clustering methods (DBSCAN, MeanShift, and WaveCluster). Then the impacts of different sampling time intervals on detection results are studied. Finally, the tracking performance and runtime for different tracking methods are evaluated.

**5.2.1. Online Multivehicle Detection.** In this work, event data are considered as pure 2D point data. The clustering technique is applied to generate object proposals. The event data for different time intervals (10ms, 20ms, and 30ms) are accumulated and can be seen in Figure 3. It is straightforward to see that clusters of event data reflect moving vehicles. The noise events surrounding each cluster are mainly generated by the environmental changes and sensor noise. Therefore, prior to generating object hypotheses, a background activity filtering step is performed to filter out the noise from the events. For each event, background activity filter checks whether one of the 8 (vertical and horizontal) neighbouring pixels has had an event within the last “us\_Time” microseconds. If not, the event being checked will be considered as noise and removed. In other words, whether a new event is considered as “signal” or “noise” is determined by whether there is a neighbouring event generated within a set interval (us\_Time). Figure 4 shows the accumulated events frame before and after the application of activity filter.

Figure 5(a) shows DBSCAN clustering results. For DBSCAN, the search radius, *Eps*, is chosen as 5 and the density, *MinPts*, is chosen as 10. The points with density higher than the specified threshold value, *MinPts*, are classified as core points while the rest are classified as noncore points. Those noncore points are also classified as noise points. Seven clusters including noise events have been detected. Figure 5(b) shows the MeanShift clustering results, with a chosen bandwidth of 20. The MeanShift algorithm successfully detected six clusters. And we can see many clusters were detected using WaveCluster from Figure 5(c). MeanShift divides many noises and objects into one cluster, and WaveCluster treats many noises as a single cluster. Their common shortcoming is that they cannot distinguish between object (here is car) and noise well.

The detection performance is assessed by clustering approach in terms of the metrics of recall and precision. The evaluation of DBSCAN, MeanShift, and WaveCluster on our neuromorphic data with different time intervals is shown in Table 3. We can see that the performance of clustering algorithms increases significantly from 10ms time interval to 20ms time interval, which shows that detection-by-clustering methods, used in this work, perform better with more events per time interval. But, from the performance of 30 ms interval, we can also know that, with the accumulation of events, more and more noise points appear, and the accuracy of the detection algorithm decreases. The results indicate

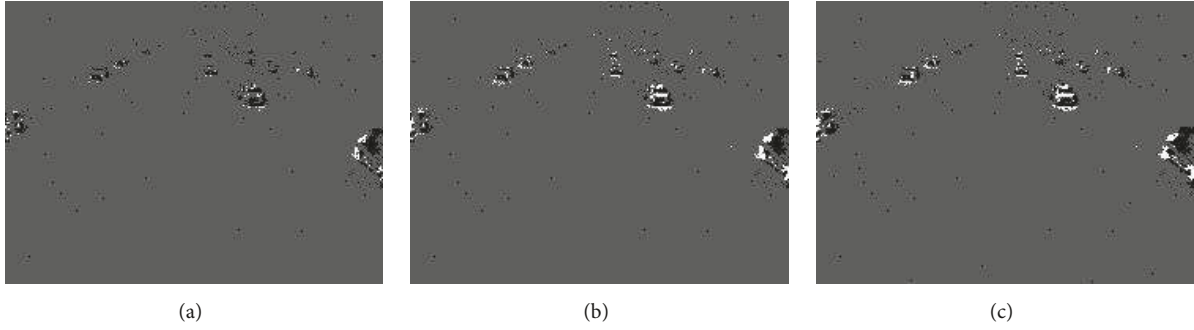


FIGURE 3: (a) Events accumulated in 10ms time interval. (b) Events accumulated in 20ms time interval. (c) Events accumulated in 30ms time interval.

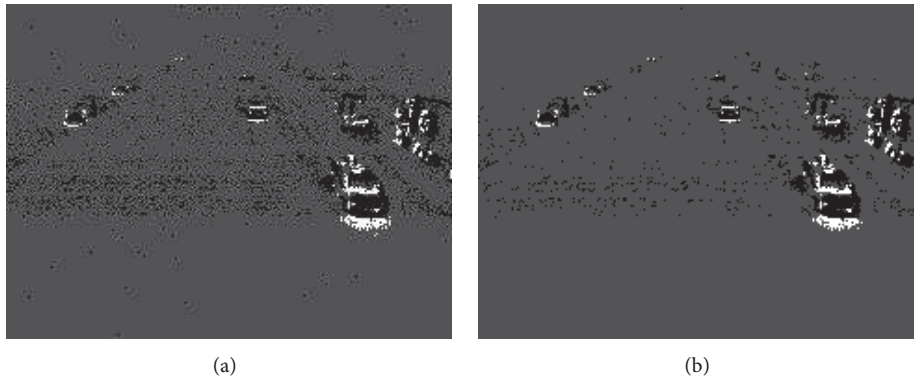


FIGURE 4: Background activity filter. (a) Events accumulated in 20ms time interval prior to filtering. (b) Events accumulated in 20ms time interval after filtering. This figure is adopted from [11].

that the detection performance is highly dependent on the number of events during the accumulated time. This points out an alternative way of accumulating a constant number of events instead of constant time intervals may increase the robustness of our detection-by-clustering approach. Among the three algorithms, MeanShift performs the worst. The reason behind it is that the density estimation of MeanShift is affected by the random noise from DAVIS. Secondly, since the MeanShift is aiming at globular clustering, it may merge some small targets when detecting as illustrated in Figure 5. Lastly, the kernel bandwidth and window size remain the same in detection, resulting in bad performance when detecting fast moving and size-changing vehicles in our scenario. From Table 3, the detection accuracy of WaveCluster is higher overall. But, the detection effect of WaveCluster at 10ms time interval is relatively poor, noise cannot be eliminated, and the detection performance is greatly affected by the number of events. In order to make the tracking algorithms get a better performance in different time intervals of the three datasets. We choose DBSCAN as detection algorithm used for comparing the tracking results.

**5.2.2. Online Multivehicle Tracking.** In this part, the four tracking algorithms have been implemented, i.e., simple

online and real-time tracking (SORT), GM-PHD filter, GM-CPHD filter, and the PDA filter. The tracking performance for the four trackers applied to the three vehicle sequences datasets is presented below.

Figure 6 shows the tracking results of SORT, GM-PHD filter, GM-CPHD filter, and the PDA filter using a series of input events for 20ms time interval. It can be seen from the continuous figures, such as Figures 6(a), 6(b), and 6(c), that our tracking algorithms perform better with moving vehicles when a new vehicle enters into the camera field of view or when an existing target leaves the camera view, target identities get updated, either by adding new IDs or by according deletion.

If any detected target in the current event frame had an overlap with an untracked detected target in previous frame, it would be registered with a new ID. As can be seen from Figure 6, most of the targets are well tracked. Especially, in the same continuous time interval, SORT tracks 29 targets, which is the largest number of targets tracked in the four algorithms. And the GM-PHD tracks 19 targets, followed by GM-CPHD with 15. However ID switching or target missing errors can also be witnessed from Figures 6(d)–6(l), PDAF performs the worst in terms of the problems of the number of targets, ID assignment and missed targets. And it can be

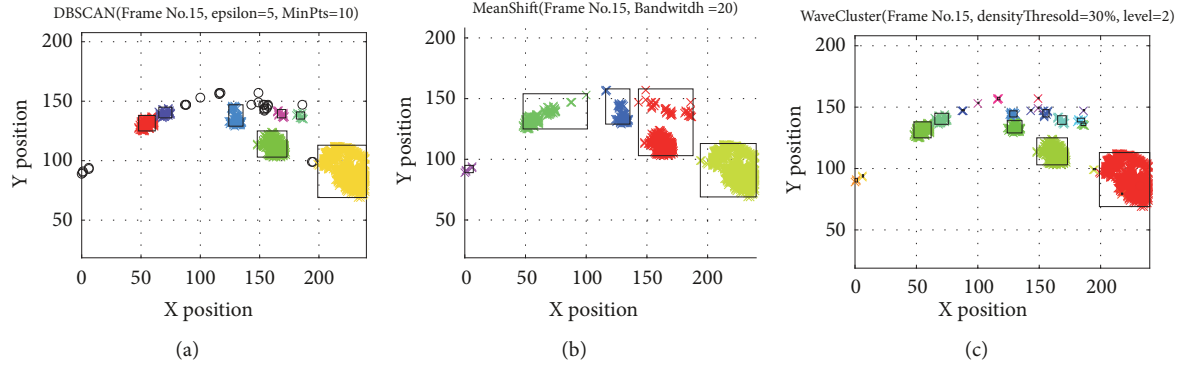


FIGURE 5: Detection-by-clustering results of (a) DBSCAN clustering applied to the events data accumulated at 20ms time interval. (b) MeanShift clustering applied to the events data accumulated at 20ms time interval. (c) WaveCluster clustering applied to the events data accumulated at 20ms time interval (best viewed in color).

TABLE 3: The performance of clustering algorithm.

Dataset	Tis	DBSCAN		MeanShift		WaveCluster	
		Recall	Precision	Recall	Precision	Recall	Precision
EventSeq-Vehicle1	10ms	53.1%	60.6%	44.5%	41.9%	44.4%	49.2%
EventSeq-Vehicle1	20ms	62.8%	64.5%	46.6%	40.7%	63.1%	64.4%
EventSeq-Vehicle1	30ms	61.9%	61.9%	44.3%	40.7%	61.8%	64.9%
EventSeq-Vehicle2	10ms	46.4%	51.9%	39.8%	38.0%	38.7%	41.4%
EventSeq-Vehicle2	20ms	52.3%	53.3%	38.3%	35.0%	51.9%	53.0%
EventSeq-Vehicle2	30ms	46.2%	47.2%	33.2%	33.5%	47.2%	52.1%
EventSeq-Vehicle3	10ms	41.1%	54.4%	35.1%	40.1%	32.1%	41.8%
EventSeq-Vehicle3	20ms	49.7%	59.4%	35.5%	36.8%	49.6%	58.0%
EventSeq-Vehicle3	30ms	47.7%	55.6%	33.5%	35.3%	49.0%	57.5%

clearly seen from Figures 6(k) and 6(l) that the same target is given different ID at different times, indicating that target lost occurs. Hence, the performance of our algorithms shows the limitations of the tracking-by-clustering system to some extent.

Table 4 shows the tracking performance metrics, i.e., MOTA, MOTP, MT, PT, ML, FP, FN, IDs, and FM for all the four trackers, i.e., SORT, GM-PHD filter, GM-CPHD filter, and the PDA filter for each 10ms, 20ms, and 30ms time intervals fed with EventSeq-Vehicle1. As the tracking component is highly dependent on the detection results, the number of times an ID-switched (IDs) is pretty large due to the inconsistent detection results. From the overall tracking performance evaluation results in Table 4, the value of MOTA and MOTP for four tracking algorithms is relatively higher. After applying these frame-by-frame-based tracking approaches, it is not surprising that we get large number of false detection, missed detection, ID switch, and fragmentation (FM). One possible way to decrease the number of missed detection, ID switch, and fragmentation (FM) is replacing the simple association metric in this paper to a more informed metric including motion information; it is able to track objects through longer periods of occlusions and disappearances. Tables 5 and 6 present the tracking performance metrics for EventSeq-Vehicle2 and EventSeq-Vehicle3, which are not as good as that of EventSeq-Vehicle1. It is especially

obvious in EventSeq-Vehicle2 with 30ms time interval, where the evaluation metrics of MOTA for tracking algorithms are very low. The main reason behind it lies in the occasional flash of huge amount of noise as shown in Figure 7, which would seriously obscure the tracking targets, resulting in periodic fluctuation in the performance of the algorithm. This “noise flash” phenomenon can attributed to the unstable working state of the sensor and variable environmental conditions. It also indicates that our three datasets are very representative and challenging. Such limitation of the neuromorphic vision sensor will also be discussed in Section 6.2.

As the first work of multitarget tracking based on neuromorphic vision sensor, we are not able to compare to state-of-the-art tracking algorithms. Instead, we provide our evaluation results as a baseline tracker for future neuromorphic vision based multiobject tracking methods.

**Runtime.** The experiment is carried out on a laptop with Intel Core™i7-6700HQ CPU with 2.60GHz quad core processor and 8.00 GB of RAM. Table 7 shows that the average FPS of the DBSCAN algorithm is 36, 17, and 8 for 10ms, 20ms, and 30ms time intervals, respectively. The decreasing frame rate is due to the increased number of events in the density search area, resulting in a more iterative process. Of course, the runtime performance is related to the selection of algorithms; for example, WaveCluster has almost the same frame rate at



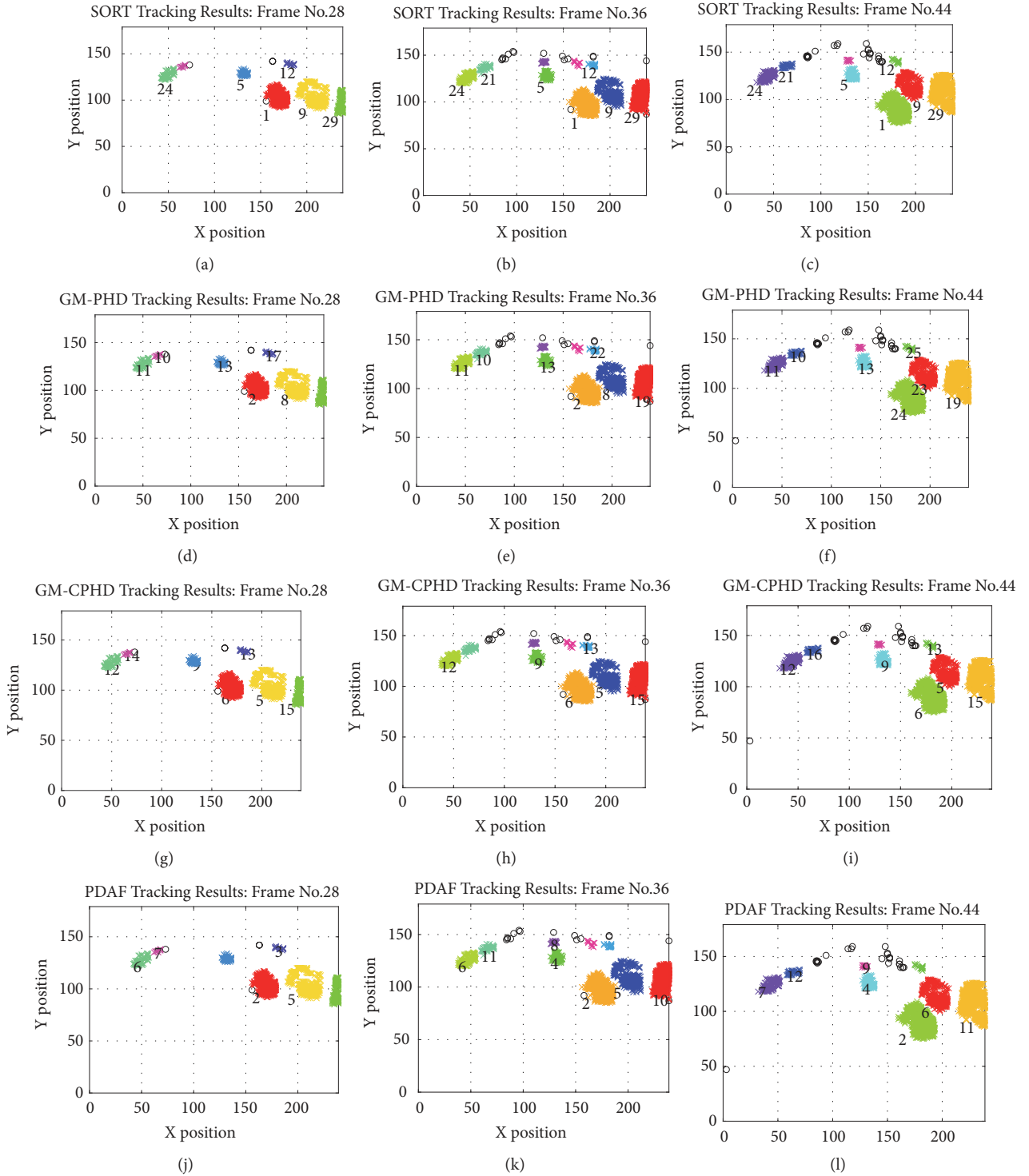


FIGURE 6: Tracking results from Frame 28 to Frame 44 for SORT, GM-PHD, GM-CPHD, and PDAF tracker. The numbers in black font near to cluster show the tracking ID (each object hypothesis is associated a tracker with a unique ID). The ID switching and lost IDs can also be witnessed from those sequences of input data (best viewed in color).

different time intervals. Additionally, in spite of the ordinary computer resources, MeanShift has a high running efficiency. For the tracking component, SORT is able to reach 552 FPS as shown in Table 8. Such a high frame rate indicates a promising

application of the sensors. According to the experimental results, our tracking-by-clustering system can run at a rate of more than 110 Hz when our tracking algorithms is combined with efficient detection algorithms, such as MeanShift. In

TABLE 4: The tracking performance for EventSeq-Vehicle1 using different tracking methods (SORT, GM-PHD, GM-CPHD, and PDAF) with detection through DBSCAN clustering. The impact of time intervals from 10ms to 30ms was studied.

Tis	Tracker	MOTA↑	MOTP↑	MT↑	PT↑	ML↓	FP ↓	FN↓	IDs↓	FM↓
10ms	SORT	36.2%	69.2%	8	79	20	2891	16369	146	1302
10ms	GM-PHD	24.0%	69.1%	1	85	21	4924	16649	1541	4097
10ms	GM-CPHD	21.1%	69.2%	3	89	15	7480	15621	900	3616
10ms	PDAF	20.9%	69.1%	0	86	21	5653	18228	158	4678
20ms	SORT	35.0%	70.2%	18	71	18	2905	6893	92	444
20ms	GM-PHD	35.1%	70.6%	18	70	19	2523	7019	323	770
20ms	GM-CPHD	25.7%	70.5%	12	75	20	3974	7180	152	716
20ms	PDAF	24.5%	70.4%	4	80	23	3576	7815	95	1371
30ms	SORT	28.5%	70.4%	12	69	26	1950	5190	94	265
30ms	GM-PHD	23.6%	70.8%	14	67	26	2323	5224	190	478
30ms	GM-CPHD	18.3%	70.7%	8	76	23	2870	5259	135	481
30ms	PDAF	19.3%	70.5%	1	76	30	2402	5701	66	900

TABLE 5: The tracking performance for EventSeq-Vehicle2 using different tracking methods (SORT, GM-PHD, GM-CPHD, and PDAF) with detection through DBSCAN clustering. The impact of time intervals from 10ms to 30ms was studied.

Tis	Tracker	MOTA↑	MOTP↑	MT↑	PT↑	ML↓	FP ↓	FN↓	IDs↓	FM↓
10ms	SORT	24.4%	70.2%	3	53	29	2170	14929	183	942
10ms	GM-PHD	13.4%	69.4%	0	60	25	3807	14994	1000	2524
10ms	GM-CPHD	7.8%	69.7%	2	69	14	7452	13084	528	2524
10ms	PDAF	13.8%	69.8%	0	57	28	4403	15206	100	3050
20ms	SORT	5.7%	68.1%	7	49	28	3304	7393	81	331
20ms	GM-PHD	15.6%	70.6%	11	52	21	2839	6524	290	729
20ms	GM-CPHD	11.3%	70.6%	10	57	17	3824	6196	118	655
20ms	PDAF	11.5%	70.5%	4	58	22	3091	6948	76	995
30ms	SORT	0%	67.3%	3	45	37	2069	5496	53	183
30ms	GM-PHD	7.6%	70.3%	5	50	30	1888	5001	149	328
30ms	GM-CPHD	-0.7%	70.1%	4	57	24	2872	4694	100	389
30ms	PDAF	5%	69.9%	3	55	27	1969	5216	49	542

comparison, DeepSort method [45] only reaches a runtime speed of 40 Hz despite the use of high performance GPU.

## 6. Conclusion and Discussion

**6.1. Conclusion.** In this paper, the first neuromorphic vision based multivehicle detection and tracking system in ITS is proposed. We provide our datasets and approaches as a baseline tracker for future neuromorphic vision based multiobject tracking methods. A variety of algorithms to perform the tracking task are presented, of which different combinations can be chosen for different accuracy and rate requirements. Hopefully, our preliminary study can motivate further research in this field, considering that the sparse stream of event data from the sensor captures only motion and salient information, which is perfect for the intelligent infrastructure systems. The proposed event-based online multiple target tracking-by-clustering system utilizes strikingly simple algorithms while it achieves good detection and tracking performance with respect to runtime requirement.

Specifically, three clustering algorithms, i.e., DBSCAN, MeanShift, and WaveCluster, were explored to deal with

the sparse data from neuromorphic sensor. After studying the detection results, the DBSCAN was selected for further detection stage due to its more robust and accurate outcome. Based on the detection results from DBSCAN, four different trackers were studied and their results were compared. The selected trackers were SORT, GM-PHD filter, GM-CPHD filter, and the PDA filter. From the experimental results, the tracking algorithm combined with DBSCAN can achieve higher accuracy, while combined with MeanShift can achieve higher frame rate of more than 110Hz. Different combinations of algorithms can be applied depending on different requirements of accuracy and real-time performance.

**6.2. Discussion.** To the best of our knowledge, the presented system is the first application of neuromorphic vision sensor on ITS which makes it well suited as a baseline, allowing for new researcher to work on intersection of the neuroscience and intelligent system. In our future work, different event encoding methods will be tried, adaptive algorithms will be explored, and the benchmark will be extended to pedestrian detection and tracking. Different filters, other than the basic

TABLE 6: The tracking performance for EventSeq-Vehicle3 using different tracking methods (SORT, GM-PHD, GM-CPHD, and PDAF) with detection through DBSCAN clustering. The impact of time intervals from 10ms to 30ms was studied.

Tis	Tracker	MOTA↑	MOTP↑	MT↑	PT↑	ML↓	FP ↓	FN↓	IDs↓	FM↓
10ms	SORT	24.6%	69.5%	1	34	24	1258	10838	109	611
10ms	GM-PHD	12.9%	68.9%	1	42	16	2484	10956	666	1903
10ms	GM-CPHD	12.5%	69.1%	2	41	16	3844	9957	363	1646
10ms	PDAF	13.8%	69.0%	0	38	21	2710	11173	70	2166
20ms	SORT	10.1%	69.3%	5	33	21	1990	5225	64	235
20ms	GM-PHD	21.4%	70.1%	5	38	16	1475	4699	188	490
20ms	GM-CPHD	13.4%	70.4%	5	36	18	2218	4698	94	402
20ms	PDAF	17.4%	70.2%	1	40	18	1682	4948	59	707
30ms	SORT	4.0%	69.0%	1	32	26	1329	3815	41	141
30ms	GM-PHD	14.3%	70.3%	7	35	17	1199	3311	120	285
30ms	GM-CPHD	6.1%	70.4%	4	34	21	1712	3300	58	240
30ms	PDAF	12.9%	70.6%	3	34	22	1112	3548	46	406

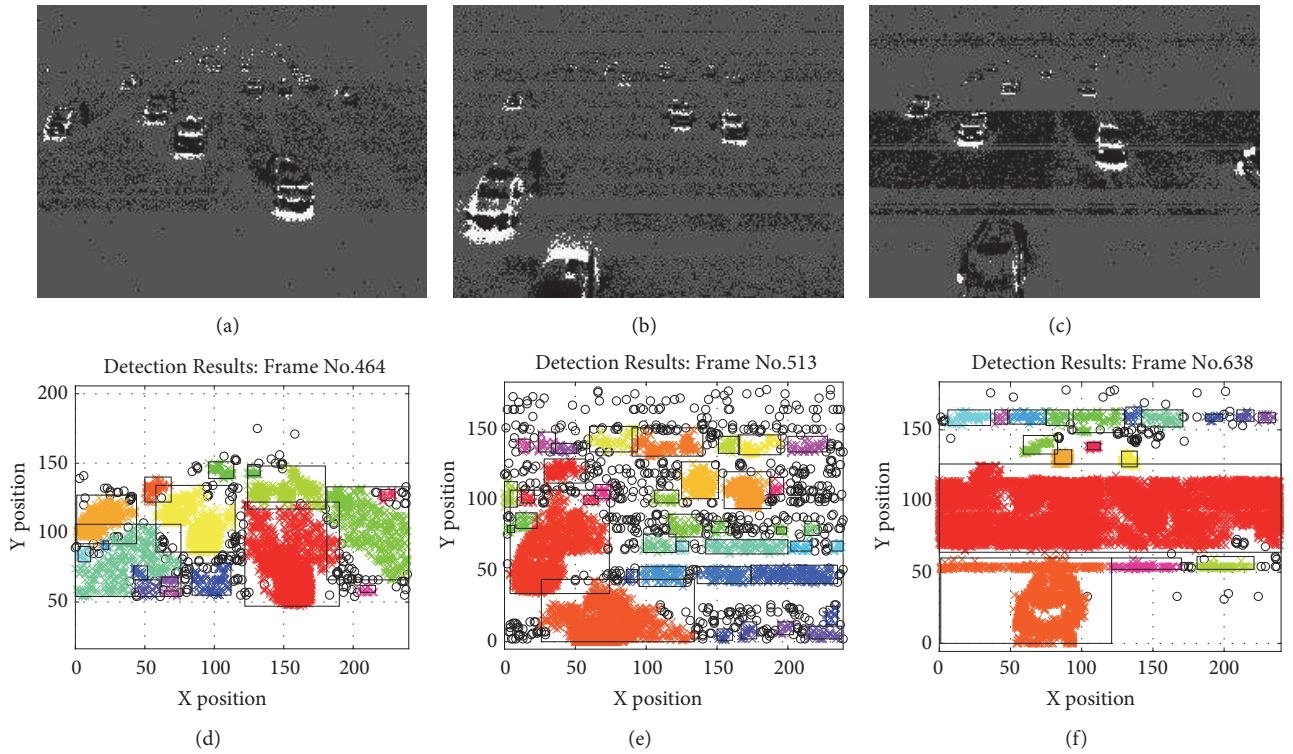


FIGURE 7: (a), (b), and (c) are the original images with the noise; (d), (e), and (f) are the results of the detection (best viewed in color).

activity filter, can be exploited to filter out the noise from the input data received from neuromorphic vision sensor. As a baseline, new approaches including recent deep learning based methods are supposed to improve the detection and tracking performance, especially the ability to identify vehicle types, such as trucks and cars and different pedestrians such as the elderly, children, etc.

*Limitation.* Admittedly, our algorithm still has some shortcomings. As can be seen in Figure 7, with noise becoming severer, the tracking system will make errors, such

as missed detection, multiple targets detection as one, the false detection of noise points as the target, and so on. The reason mainly lies in the immaturity of neuromorphic sensor technology. To be specific, the inherent defects of the current neuromorphic sensor lead to instability in collecting event information, which affects the quality of data and thus degrades the performance of the algorithms. Hence, a development of the sensor is indispensable before wide application of it in intelligent transportation system (ITS). It is also significant to note that, in order to take full advantage of event data, completely new neuromorphic vision algorithms are

TABLE 7: The FPS for EventSeq-Vehicle1 using different detection methods (DBSCAN, MeanShift, and WaveCluster). The impact of time intervals from 10ms to 30ms was studied.

Detector	Tis	FPS
DBSCAN	10ms	36
MeanShift	10ms	160
WaveCluster	10ms	17
DBSCAN	20ms	18
MeanShift	20ms	107
WaveCluster	20ms	19
DBSCAN	30ms	8
MeanShift	30ms	71
WaveCluster	30ms	15

TABLE 8: The FPS for EventSeq-Vehicle1 using different tracking methods (SORT, GM-PHD, GM-CPHD, and PDAF).

Tracker	FPS
SORT	552
GMPHD	3
GMCPHD	4
PDAF	46

required instead of extending existing methods of computer vision, taking into account the brand new information stream and the extremely high frame rate of the neuromorphic vision sensor.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the German Research Foundation (DFG) and the Technical University of Munich within the Open Access Publishing Funding Program. Part of the research has received funding from the European Unions Horizon 2020 Research and Innovation Program under Grant Agreement no. 785907(HBP SGA2) and from the Shanghai Automotive Industry Sci-Tech Development Program under Grant Agreement no. 1838. The authors would like to thank Zhongnan Qu for technical assistance and the help of data acquisition.

## References

- [1] P. Lichtsteiner, C. Posch, and T. Delbruck, "A  $128 \times 128$  120 dB 15  $\mu$ s latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.
- [2] E. Mueggler, C. Bartolozzi, and D. Scaramuzza, "Fast event-based corner detection," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.
- [3] X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, "Asynchronous event-based multikernel algorithm for high-speed visual features tracking," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 8, pp. 1710–1720, 2015.
- [4] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, AAAI Press, 1996, <http://dl.acm.org/citation.cfm>.
- [6] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wavecluster: A multi-resolution clustering approach for very large spatial databases in," in *Proceedings of the 24rd International Conference on Very Large Data Bases*, vol. 98, pp. 428–439, 1998.
- [7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, Phoenix, AZ, USA, September 2016.
- [8] B. Vo and W. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.
- [9] B. Vo, B. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, part 2, pp. 3553–3567, 2007.
- [10] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter: estimation in the presence of measurement origin uncertainty," *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, 2009.
- [11] G. Hinz, G. Chen, M. Aafaque et al., "Online Multi-object Tracking-by-Clustering for Intelligent Transportation System with Neuromorphic Vision Sensor," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 10505, pp. 142–154, 2017.
- [12] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A Survey of Vision-Based Traffic Monitoring of Road Intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2681–2698, 2016.
- [13] J. C. Rubio, J. Serrat, A. M. López, and D. Ponsa, "Multiple-target tracking for intelligent headlights control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 594–605, 2012.
- [14] X. Zhang, S. Hu, H. Zhang, and X. Hu, "A real-time multiple vehicle tracking method for traffic congestion identification," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 6, pp. 2483–2503, 2016.
- [15] J. Zhou, D. Gao, and D. Zhang, "Moving vehicle detection for automatic traffic monitoring," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 1, pp. 51–59, 2007.
- [16] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, 2013.



- [17] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," in *Proceedings of the 1999 IEEE/IEEEJ/SAI International Conference on Intelligent Transportation Systems*, pp. 758–763, October 1999.
- [18] M.-C. Huang and S.-H. Yen, "A real-time and color-based computer vision for traffic monitoring system," in *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 2119–2122, Taipei, Taiwan.
- [19] W. Zhang, Q. M. J. Wu, and H. B. Yin, "Moving vehicles detection based on adaptive motion histogram," *Digital Signal Processing*, vol. 20, no. 3, pp. 793–805, 2010.
- [20] B.-F. Wu, S.-P. Lin, and Y.-H. Chen, "A real-time multiple-vehicle detection and tracking system with prior occlusion detection and resolution," in *Proceedings of the 5th IEEE International Symposium on Signal Processing and Information Technology*, pp. 311–316, Greece, December 2005.
- [21] B. Aytekin and E. Altuğ, "Increasing driving safety with a multiple vehicle detection and tracking system using ongoing vehicle shadow information," in *Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics, SMC 2010*, pp. 3650–3656, Turkey, October 2010.
- [22] M. Betke, E. Haritaoglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Machine Vision and Applications*, vol. 12, no. 2, pp. 69–83, 2000.
- [23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 3354–3361, Providence, RI, USA, June 2012.
- [24] L. Wen, D. Du, Z. Cai et al., "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Computer Vision and Pattern Recognition*, 2015.
- [25] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *Computer Vision and Pattern Recognition*, 2015.
- [26] D. P. Moeys, F. Corradi, E. Kerr et al., "Steering a predator robot using a mixed frame/event-driven convolutional neural network," in *Proceedings of the 2nd International Conference on Event-Based Control, Communication, and Signal Processing, EBCCSP 2016*, Poland, June 2016.
- [27] Z. Jiang, Z. Bing, K. Huang, G. Chen, L. Cheng, and A. Knoll, "Event-Based Target Tracking Control for a Snake Robot Using a Dynamic Vision Sensor," in *Neural Information Processing*, vol. 10639 of *Lecture Notes in Computer Science*, pp. 111–121, Springer International Publishing, Cham, Switzerland, 2017.
- [28] G. Chen, Z. Bing, F. Roehrbein et al., "Efficient Brain-inspired Learning with the Neuromorphic Snake-like Robot and the Neurorobotic Platform," in *Proceedings of the IEEE Transactions on Cognitive and Developmental Systems*, p. 1, 2018.
- [29] H. Blum, A. Dietmüller, M. Milde, J. Conradt, G. Indiveri, and Y. Sandamirskaya, "A neuromorphic controller for a robotic vehicle equipped with a dynamic vision sensor," in *Proceedings of the Robotics: Science and Systems*, Cambridge, MA, USA, July 2017.
- [30] M. Litzenberger, A. N. Belbachir, and N. Donath, "Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor," in *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC '06)*, pp. 653–658, Toronto, Canada, September 2006.
- [31] M. Litzenberger, C. Posch, D. Bauer et al., "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *Proceedings of the 2006 IEEE 12th Digital Signal Processing Workshop and 4th IEEE Signal Processing Education Workshop, DSPWS*, pp. 173–178, USA, September 2006.
- [32] A. I. Maqueda, A. Loquercio, G. Gallego, N. Garca, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars in," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5419–5427, 2018.
- [33] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "Ddd17: End-to-end davis driving dataset, arXiv preprint," *Computer Vision and Pattern Recognition*, 2017.
- [34] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck, "DVS benchmark datasets for object tracking, action recognition, and object recognition," *Frontiers in Neuroscience*, vol. 10, 2016.
- [35] D. Tedaldi, G. Gallego, E. Mueggler, and D. Scaramuzza, "Feature detection and tracking with the dynamic and active-pixel vision sensor (DAVIS)," in *Proceedings of the 2nd International Conference on Event-Based Control, Communication, and Signal Processing, EBCCSP 2016*, Poland, June 2016.
- [36] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, "Low-latency visual odometry using event-based feature tracks," in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2016*, pp. 16–23, Republic of Korea, October 2016.
- [37] L. Everding and J. Conradt, "Low-latency line tracking using event-based dynamic vision sensors," *Frontiers in Neurobotics*, vol. 12, 2018.
- [38] Z. Ni, C. Pacoret, R. Benosman, S. Ieng, and S. Régnier, "Asynchronous event-based high speed vision for microparticle tracking," *Journal of Microscopy*, vol. 245, no. 3, pp. 236–244, 2012.
- [39] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A  $240 \times 180$  130 dB 3  $\mu$ s latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.
- [40] T. A. Biresaw, T. Nawaz, J. Ferryman, and A. I. Dell, "ViTBAT: Video tracking and behavior annotation tool," in *Proceedings of the 13th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2016*, pp. 295–301, USA, August 2016.
- [41] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.
- [42] G. Chen, F. Zhang, D. Clarke, and A. Knoll, "Learning to track multi-target online by boosting and scene layout," in *Proceedings of the 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013*, pp. 197–202, USA, December 2013.
- [43] L. Leal-Taixé, A. Milan, I. D. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *Computer Vision and Pattern Recognition*, 2015.
- [44] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [45] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, Beijing, China, September 2017.

