IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 10, 2022, accepted February 18, 2022, date of publication February 25, 2022, date of current version March 9, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3154895

Unsupervised Adaptive Multi-Object Tracking-by-Clustering Algorithm With a Bio-Inspired System

JULIO GUILLEN-GARCIA, DANIEL PALACIOS-ALONSO, ENRIQUE CABELLO, (Member, IEEE), AND CRISTINA CONDE

Escuela Técnica Superior de Ingeniería Informática—Universidad Rey Juan Carlos, Campus de Mostoles, 28933 Madrid, Spain

Corresponding author: Daniel Palacios-Alonso (daniel.palacios@urjc.es)

This work was supported by Universidad Rey Juan Carlos under Grant (COMPAD - 2022/00004/004).

ABSTRACT A problem of current interest is how to emulate nature by acquiring information in a neuromorphic-like fashion; namely, by using configurable hardware and electronic systems to emulate the information gathering and processing strategies of biological systems. In this paper, we introduce BioCAMSHIFT, an algorithm for a bio-inspired system that acquires information via a neuromorphic process and uses it to track multiple objects. The system consists of a silicon retina that simulates the behavior of the human eye together with a communication system that uses an Address-Event Representation protocol to transmit information in a way analogous to that of biological neural systems. An unsupervised procedure, based on the CAMSHIFT algorithm, is then used for multi-object tracking. It takes advantage of the retina's high event rate to adapt to the changing sizes of the objects in its field of view. The proposed system has been experimentally validated using a data set from Freeway 210 in Pasadena, California, demonstrating a significantly better improvement in terms of multi-vehicle detection and tracking performance over the current state of the art.

INDEX TERMS BioCAMSHIFT, address-event representation (AER), CAMSHIFT, bio-inspired system, bioinformatics, silicon retina devices, clustering algorithms, neuromorphic system, jAER.

I. INTRODUCTION

Neuromorphic systems [1], [2] attempt to emulate very specific biological functions, usually of a sensory type, whose structure and functionality have been analyzed in great detail. Their aim is to produce bio-inspired systems by using configurable hardware and electronic systems to emulate the ways of acting, information processing, and problem-resolution strategies of biological systems. The interest in these systems is two-fold: first, to study models that allow a better understanding of the neuronal functioning in nature and, second, to emulate biological functionality so as to obtain more effective artificial devices (e.g., silicon retinas).

Consider, for example, the particular field of Artificial Vision (AV). The typical approaches found in the literature [3] use computers to extract information from images of the physical world. Such images are usually presented as functions associating to every point in the image, a value relative to some property of the pixel or voxel it represents (e.g., brightness, hue, intensity, etc.). This representation can in turn correspond to a static image, a three-dimensional scene, a video sequence, views from multiple cameras, etc. Algorithms are then applied to these functions so as to filter, enhance and extract features that may be deemed important for achieving the particular task at hand. The main problem, however, is that this way of representing and processing images is configured to the architecture and functionality of conventional computer systems. Therefore, one may say that current AV approaches attempt to ''fit the problem to the tool'', a tool that operates in a way that differs vastly from the biological principles AV strives to emulate.

In contrast, the current consensus in neuroscience [1], [4] is that the functioning of a biological retina can be described as a matrix of light-sensitive neurons that emit electrical impulses when stimulated by the light they receive. Basically, a retina neuron receiving low-intensity light will show a low activity whereas a neuron being stimulated at a greater intensity will generate events at a higher rate. Over time, the amount of light

The associate editor coordinating the review of this manuscript and approving it for publication was Lefei Zhang.

received by a specific neuron can vary, and with it the rate of the impulses it generates, making such rate a function of time. Subsequently, after the luminous impulses are transformed into nerve impulses by the retina, the impulses reach through the optic nerve to the posterior region of the brain where they are interpreted by a complex mechanism involving millions of neurons

Analogously to the operation of a biological retina, a Temporal Difference Silicon Retina (TDSR) [5] is composed of a matrix of devices called pixels, each of which works asynchronously and independently of the others, capturing light and emitting pulses according to the difference in luminosity over time. The generated pulses are all of equal magnitude but are either positive or negative depending on whether the pixel has captured a change in luminosity that goes from dark to bright (positive) or vice versa (negative).

However, since bio-inspired systems must seek to emulate both the structure and functionality of biological systems, it is also mandatory to establish a communication protocol for transmitting the information provided by the neuromorphic retina in a manner that may be deemed analogous to that of biological neural systems. A protocol of representation by means of event addresses (Address-Event Representation or AER) was thus devised for this purpose. It was first proposed by Sivilotti [6] in 1991 and later extended by Mahowald [7].

This protocol arose from the need to interconnect the cells of one layer of a chip with those of other layers on another chip, so that this point-to-point connection could simplify the implementation of bio-inspired systems [6]. AER can be described as a mechanism for transferring the status of a cell set from one chip to another. The emitting chip consists of a collection of computer cells, called ''neurons'', that can send and receive signals. Each cell has an assigned address. To send a signal, a neuron transmits a pulse to an arbitrator, which retransmits it through an asynchronous handshake protocol. Such signal, called an ''event'', includes the time at which the address is transmitted as well as the value of that address. The first vision sensor to use AER [8] was created by Mahowald & Mead in 1994.

In turn, the history of the TDSR begins with Kramer [5], who presented a 48×48 array sensor that unfortunately struggled to operate properly when objects in its field of view moved at low speeds. In 2008, Lichtsteiner *et al.* [9] presented the Dynamic Vision Sensor (DVS) 128, an architecture much superior to Kramer's thanks to its particular photodiode design. This sensor's architecture, initially developed for the CAVIAR [40] [41] project, funded by the European Commission to develop a chip for a vision system based on the AER protocol, has continued to evolve. For example, Lenero-Bardallo *et al.* [10] further improved it by reducing latency times, and today there are DVS-type prototypes with 1280×720 [11] pixel arrays.

Alternative TDSR architectures include the Asynchronous Time-based Image Sensor (ATIS), which combines the information of two sensors for each pixel [12], as well as the Dynamic and Active pixel Vision Sensor (DAVIS), which provides global intensity information [13] and has further evolved to enable working in color [14].

The main contribution of this paper is that of introducing a bio-inspired framework addressing the problem of efficiently locating and tracking multiple objects traversing the field of view of a stationary silicon retina. Previous research in this area has either emphasized using a Mean Shift clustering algorithm without data buffering (an approach that excels in low memory usage and real time processing but does not take advantage of event history [15] and suffers from spurious events that interfere with the accuracy of the algorithm [16]), or has been based on creating data frames followed by applying classical clustering algorithms (something that does not take advantage of the silicon retina extremely-high time-resolution capability) [17].

This paper's approach differs from previous work in that it doesn't squander previous events but uses temporal information for more accurate tracking. Thus, by not discarding previous system activity, it is less prone to noise-generated events interfering with the accuracy of the algorithm. Also, BioCAMSHIFT begins operating from the get-go; in other words, unlike current methods that need certain activity for achieving steady-state operation, no initialization stage is required. Such property not only enables this procedure to respond promptly in rapidly-changing situations, but also provides the capability to instantaneously recover from interruptions caused by hardware or software mishaps, making it an extremely robust tracking tool. Additionally, unlike methods that encode events into frames, BioCAMSHIFT operates on individual events as they occur, allowing real-time processing of the scene. Finally, a life function is defined, which, combined with an event declaration filter, helps pruning irrelevant events and allowing minimal storage and processing requirements.

The paper is organized as follows. Section II presents an overview of both, the characteristics of the data provided by a silicon retina and of the current state-of-the-art regarding silicon retina-based tracking algorithms and their applications. Section III describes the BioCAMSHIFT algorithm and Section IV states and discusses the key experimental outcomes. Finally, Section V summarizes the results and presents the conclusions and future lines of research.

II. RELATED WORK

As mentioned in the Section I, this paper focuses on the problem of locating and tracking multiple objects traversing the field of view of a stationary silicon retina. This retina can be visualized as an array of imaging pixels where, rather than light intensity, each pixel samples the temporal-derivative of such intensity, outputting then solely the sign of such derivative. In other words, if the intensity of the light impinging on pixel (m, n) follows over time a function $f_{m,n}(t)$, then the pixel's output, $p_{m,n}(t)$, will be

$$p_{m,n}(t) = S_{\alpha,\beta} \frac{df_{m,n}(t)}{dt} \tag{1}$$

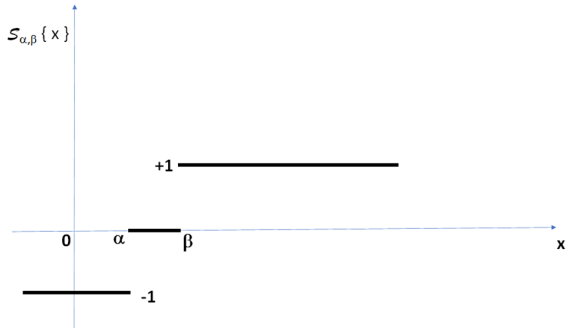

FIGURE 1. A depiction of the function $S_{\alpha,\beta}\{x\}$ of expression (2).

where $S_{\alpha,\beta}\{x\}$ is used here to denote a variant of the sign function [18] containing a threshold region intended for false alarm suppression purposes. Namely, as shown in Fig. 1,

$$S_{\alpha,\beta}\{x\} = \begin{cases} 1 & \text{if } x > \beta \\ 0 & \text{if } \alpha \leq x \leq \beta \\ -1 & \text{if } x < \alpha \end{cases} \tag{2}$$

where the real-valued scalars α and β are thresholds chosen by the user for each (m, n). Sensor outputs other than 0 (i.e., when $p_{m,n}(t) \neq 0$), are known as ''events''. The sensor's complete output in such cases consists of the event value (i.e., ± 1) and the array coordinates (m, n) of the pixel involved.

Observe that expression (1) corresponds to a temporal high-pass filter, signifying that rapid temporal variations sensed by a given pixel will be tagged as ''events'' while slowly-occurring changes will be ignored (another useful interpretation is that, after such filtering, the output of each pixel will tend to be temporally uncorrelated). This means that if the sensor itself is stationary, very-slow-moving or motionless objects in the retina's field of view will be substantially or completely suppressed, making this sensor suitable for temporal change-detection applications. Alternatively, with the silicon retina's current architecture, if the sensor itself were moving, immobile objects would seem to change location, substantially increasing false alarm rates in such change-detection and object-tracking applications. (We will assume here the use of a stationary retina sensor, as sensor alignment considerations fall outside the scope of this paper).

Observe too that although the outputs of the individual pixels may be temporally uncorrelated, they may not be uncorrelated spatio-temporally across pixels. Such inter-pixel correlation is the property that is generally exploited to cluster the retina's output into ''objects'' and track them.

Finally, note that by confining the filter outputs to the set $\{-1, 0, 1\}$, per Eq. (2), the sensor not only significantly simplifies the process of sampling and storing data at very-high temporal-resolution per-pixel rates, but also simplifies subsequent data processing. This is particularly so when the processing involves algorithms such as those used for distribution estimation (i.e., clustering/histogramming), sparse/compressed-sensing methods or 1-bit adaptive processing techniques.

Some of the original algorithms for tracking objects with neuromorphic retinas tagged the sensor array data as belonging to either of various regions of interest (ROIs). For instance, the algorithm of [19] and [20] is based on the detection of peaks of activity in predefined ROIs, corresponding to highway lanes for the intended application of vehicle detection, counting and speed measurement. Subsequently, clustering-based tracking algorithms appeared, the first of which [15], [21]–[23], were based on the Mean Shift approach [24]–[26], using distances to assign events to a predefined number of clusters.

Since the CAMSHIFT algorithm [27] we will be using in this paper is an extension of Mean Shift, it may be appropriate to briefly sketch the latter. Given discrete data sampled from a density function, Mean Shift is an iterative procedure for locating the function's maxima. The procedure begins with an initial estimate of the function's mean and, by weighting nearby points according to a pre-selected window or ''kernel'', the mean is re-estimated. The kernel is then centered at the new mean estimate and the procedure is repeated until convergence.

The Mean Shift cluster-tracking approach method has been utilized in multiple applications such as vehicle counting [19], human tracking [15] and even for a robotic goalie [23]. Delbruck and Lang [28] further enhanced the robotic goalie algorithm, achieving a fast self-calibrating robotic goalie with a 3 ms. reaction time. Schraml *et al.* [29] proposed a stereoscopic system, and later Schraml *et al.* [30] suggested a Mean Shift approach to track objects in this 3D AER system. Later on, Camunas-Mesa *et al.* [31] expanded the stereo visual tracking algorithm to solve object occlusion.

Another version of the Mean Shift cluster-tracking approach was developed by Barranco *et al.* [32], as they proposed an event-based Mean Shift clustering method using Kalman filters [33] for multi-target tracking. On the other hand, Gómez-Rodriguez *et al.* [34] proposed a complete hardware system where they tracked multiple objects by calculating their center of mass and speed, an approach similar to which was then used in [35] for the monitoring of particles in fluids.

Another proposed method was that of [36] which combined clusters based on distance and density to generate more robust tracking, while [37] presented a framework for tracking with DVS, that was later improved in [38]. Alternatively, [17] addressed the problem of vehicle detection and tracking by first comparing the outputs of clustering, MeanShift and DBSCAN [39] algorithms to perform object detection, followed by object-tracking by comparing the outputs of algorithms such as SORT [40], GM-PHD [41], GM-CPHD [42], and PDAF [43].

Research has also been done in applications involving the use of the retina sensor for the surveillance of individuals. For example, the method of Fu *et al.* [44] uses the centroid obtained from the average of sensor events to detects

falls, and Belbachir *et al.* [45] later expanded this work to stereoscopic environments. Schraml *et al.* [30] used a stereoscopic system to track people, while Piatkowska *et al.* [46] addressed the problem of tracking people in high-occlusion environments through the use of Gaussian Mixture Models (GMM) [47]; this clustering method is succesfully used in [48] to track pedestrians. Meanwhile, [49] addresses the pedestrian detection problem with an event-to-frame encoding method combined with Convolutional Neural Networks (CNN). Recently, the NeuroAED system has been presented; it aims to efficiently detect abnormal events in visual surveillance [50].

Additional methods of interest are those based on the Hough transform. These are particularly useful when objects follow rigid trajectories whose shape is known a priori, as the Hough transform can be made to match and highlight them. For example, in [51], [52] two sensors are used to aid in balancing a pencil by first estimating its position by means of a Gaussian in Hough space. In [53] the Hough transform was also used, this time for detecting microparticles.

Alternative methods of interest are those appearing in [14], [54]–[57]. In particular we would like to highlight the implementation in [57] of a noise sensor using a neuromorphic chip together with an ATIS sensor, for which they developed a Neural Network-Based Nearest Neighbor (''NeuNN'') filtering algorithm. In [58] the object tracking problem is accomplished by training a binary classifier with statistical bootstrapping. Recently, in [59] a spatial-temporal mixed particle filter (SMP Filter) is proposed to track LED-based rectangles. In [49], a Restricted Spatiotemporal Particle Filter (RSPF) tracking algorithm is presented, and evaluated tracking fingers. Lastly, in [60], a combined use of SNNs and silicon retina is proposed, applied to object tracking.

Finally, it must be underscored that one of the main problems that researchers have to face in bio-inspired systems is the lack of open-access widely-accepted databases for benchmarking algorithms. Tan *et al.* [61] enumerated some of the challenges involved in benchmarking metamorphic vision procedures. So, despite the fact that some limited data sets have been made available [17], [62], there is still a lack of comparative studies [62].

III. ALGORITHMIC METHODS

The proposed multi-object tracking algorithm does not follow the classic strategies mentioned above because of the particular way in which the sensor we use represents movement by pulses. Because of this, motion analysis is performed by processing the stream of events generated by the retina and not by the processing of frames, meaning that the system must be able to adapt continuously to the events present in the scene. To this end, we designed BioCAMSHIFT as an *ad hoc* unsupervised clustering-tracking procedure, based on the CAMSHIFT algorithm, capable of processing the flow of events received through the neuromorphic retina.

A. OBJECT MODEL

In order to be able to track objects consistently, the characteristics of the neuromorphic sensor and the protocol for the representation of events (call it ''Address-Event Representation'' or AER) must be taken into account to properly define the object.

We will define an event $\boldsymbol{\gamma}$ as the vector

$$\boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{x}_\gamma \\ t_\gamma \end{bmatrix} \tag{3}$$

where 2×1 vector $\boldsymbol{x}_\gamma$ contains the retina array coordinates of the event generated at instant t_γ (such t_γ is known as the event's ''timestamp''). Notice that we are not taking the sign of the sensor output into account, just the fact that a significant change has taken place.

In turn, an object $\boldsymbol{id_\theta}$ is modelled as a cluster of events that act in a correlated fashion, within a region of interest (ROI), for a time frame T. The size of such ROI is dynamically determined by the event activity in the region and during the time frame T. Thus, an object can be defined as the 5×1 vector

$$\boldsymbol{\theta} = \begin{bmatrix} id_\theta \\ \boldsymbol{x}_\theta \\ t_\theta \\ M_0 \end{bmatrix} \tag{4}$$

where id_θ is the object's label and 2×1 vector $\boldsymbol{x}_\theta$ and scalar M_0 respectively contain the centroid and the zero moment, in sensor array coordinates, of object θ at instant t_θ.

B. EVENT DECLARATION FILTER

The potential sources of error in the proposed system include system noise, peculiarities of the data processing and discretization method, the intrinsic characteristics of the AER protocol, etc. It must be recalled that since the processing embedded in the silicon retina involves taking derivatives (a high-pass filtering process), the high-frequency components of the data outputted by each pixel – including noise – will be enhanced, meaning that a large number of declared events will likely be spurious. Filtering such spurious activity would not only improve subsequent clustering and tracking processing stages, but would also benefit the system by reducing algorithm execution times and optimizing the visualization of the results.

Since an object has been defined as a cluster of correlated activity, one can remove noise-related outputs by filtering out any returns that behave in a manner that is inconsistent or uncorrelated across events. Only sets of returns whose activity correlates with that of others would thus be taken into account and declared events, while returns that pop-up arbitrarily would be ignored.

We use an M-out-of-N event-declaration process to filter out such spurious events. Namely, as shown in Fig. 2, when a nonzero output from the retina appears at a given pixel at time $t = t_g$, an $N = K \times L - 1$ pixel region around it is examined over a time interval $[t_g - T, t_g]$. The nonzero retinal output at the pixel under test is then declared an

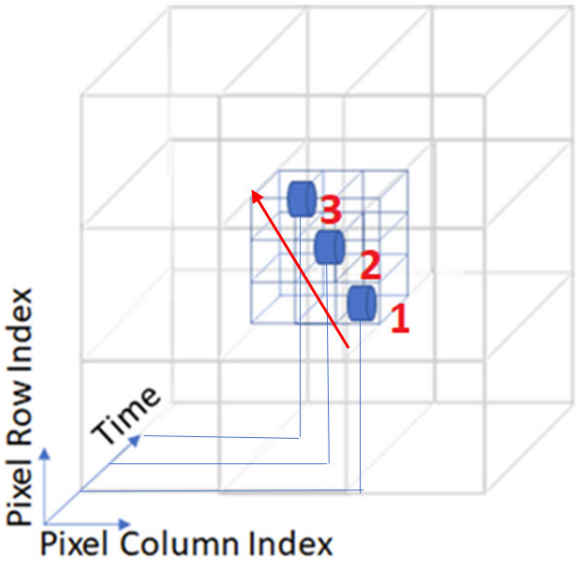

FIGURE 2. A non-zero retinal output will be declared an event only if at least M other such outputs appear within an $N = K \times L - 1$ cell spatio-temporal window of data surrounding it (M, K and L as selected by the user). The drawing in this Fig. illustrates the hypothetical case of a one-pixel object traveling through points 1, 2 and 3 in spatio-temporal space. Examining a window of 3 × 3 pixel-time data cells centered at the blue dot at point 2, one can observe that the two other such dots are also contained in it. Thus, the center dot will be declared an event for $M \leq 2$.

event only if at least M other such outputs appear within the region under examination. This process of declaring events by requiring the existence of multiple non-zero values within spatio-temporal proximity of each other can be shown to reduce the number of false declarations. Unfortunately, when used improperly, it will also reduce the number of true-event declarations.

For example, for purposes of illustration let's consider a simple model that assumes a Bernoulli process with a per-pixel probability p_s that a spurious non-zero retinal output occurs in a time interval T. If we define the probability p_{FE} of a false-event as that for the case when a spurious signal appears at the pixel under test and at least M other spurious outputs occur in an N-pixel window around it, we'll have

$$p_{FE} = p_s \sum_{m=M}^{N} \binom{N}{m} {p_s}^m (1 - p_m)^{N-m} \tag{5}$$

Fig. 3 plots the probability p_{FE} of false-event declarations, as a function of p_s, for various values of M in the case of a 3×3 spatial window (i.e., $N = 3 \times 3 - 1 = 8$). Observe that even for the least stringent case where $M = 1$, the probability p_{FE} of false event declaration decreases for basically all values of p_s. Such decrease is more marked for lower p_s values (let's say $p_s < 0.2$) than for high values of ($p_s > 0.5$), where it becomes negligible. Observe too that such reduction in p_{FE} becomes far steeper for larger values of M.

However, before selecting a high-M value for noise-filtering purposes, one must take into account that the exact

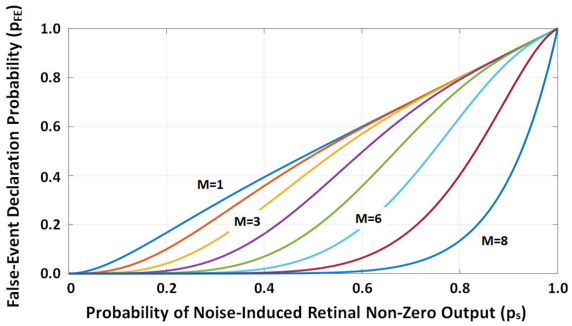

FIGURE 3. Computation of the probabilities of false-event declaration, p_{FE}, as function of per-pixel probability of non-zero retinal output, p_s, for various values of M. The M-out-of-N event-declaration process described in the text is used for the case of the 3 × 3 spatio-temporal window of Fig. 2.

same set of curves also apply for purposes of determining the probability of declaring a true-event (call it p_{TE}). In other words, if we let p_t be the probability that a non-zero retinal output at a given pixel indeed belongs to a signal of interest, the probability of declaring a true-event will again be given by expression (5), this time using p_t instead of p_s. Thus, selection of a high-M value would also reduce the probability of declaring actual events.

Since under practical conditions the system thresholds are generally set so that p_s is as low and p_t is as high as possible (i.e., $p_s \ll 0.5 < p_t$), the best compromise – and the one we selected for our purposes – is to set $M = 1$, as then p_{TE} will not be significantly affected while p_{FE} may be substantially reduced.

C. CREATION AND DESTRUCTION OF OBJECTS

The creation of an object occurs when there is sufficient correlated event activity in an area where no activity has been recorded recently; that is, where there are no objects in the vicinity of the coordinates of such events (Fig. 4a). Once an object has been created, an initial ROI is defined. During the lifetime of an object, the frequency with which events are generated in that ROI will define its size. If the event activity in the vicinity of an object happened to be below a preset threshold for a given period of time, such object would be deleted.

D. CALCULATION OF THE CENTROID

The life function $\phi(t, t_\gamma)$ of an event uses the event's timestamp, t_γ, to determine the event's influence level in a cluster. An example of such function is

$$\phi(t, t_\gamma) = \begin{cases} e^{-(t-t_\gamma)} & \text{if } t - t_\gamma \leq \Gamma \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The time limit Γ is a configurable parameter that represents the maximum time in nanoseconds that we consider an event to be alive. Fig. 5 depicts a graphical representation of events generated by a single object at an instant t_γ given a specific Γ.

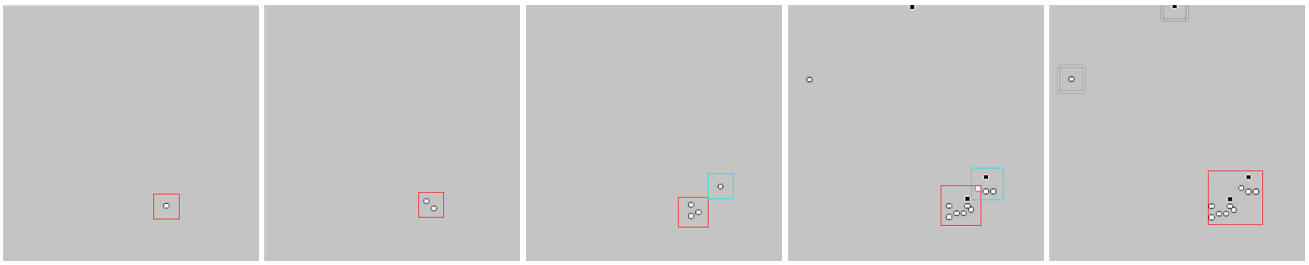

FIGURE 4. A (valid) new event in Fig. 4a generates a new cluster as there is no previous activity on the vicinity. Further activity in Fig. 4b modifies the cluster's centroid but still does not affect the cluster ROI. In Fig. 4c, more activity in the vicinity creates a new cluster as it hasn't happened on first cluster ROI. After more activity in both clusters, in Fig. 4d finally an event falls in both the first and second ROI clusters, so a merging has to be calculated. After the cluster merging, in Fig. 4e it can be seen that the first cluster prevails as it has more influence than the second one. More events create tentative clusters in different regions.

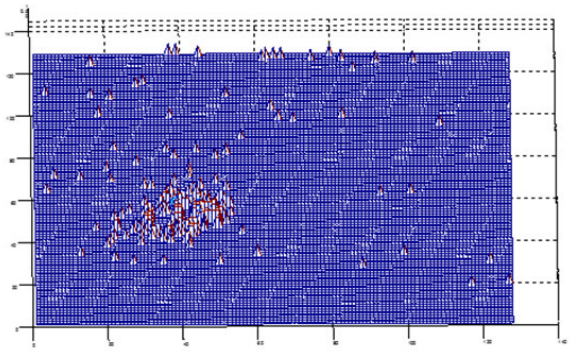

FIGURE 5. Representation of the life function at a given time t_γ. Each pixel shows the value of the life function in the form of a pyramid, the higher the value, the more recent the event. Thus, at any given instant there are a multitude of events to be taken into account for object detection and tracking.

This parameter allows us to discard old events and emphasize the most recent ones. As a result, events lose their impact on the object with the passing of time. This way, the centroid can be calculated as the weighted average of the Cartesian coordinates of the events in a region of interest, in proportion to their influence at a given instant.

In our case we'll use the center of mass of a cluster of events to define the center of coordinates of a distribution of influence in an ROI. Such center of mass is the cluster's 1st moment vector, $\boldsymbol{M}_1$, normalized by the zeroth moment of influence, M_0, where

$$M_0 = \sum_{\varphi \in ROI} \phi(t, t_\gamma) \tag{7}$$

Since M_0 is the sum of the life functions of the events located in the cluster's ROI, it can be interpreted as the total influence exerted by an object.

The 2×1 vector $\boldsymbol{M}_1$ of first moments along the events' x and y coordinates is

$$\boldsymbol{M}_1 = \sum_{\varphi \in ROI} \phi(t, t_\gamma) \boldsymbol{x}_\gamma \tag{8}$$

where $\boldsymbol{x}_\gamma$ contains the spatial coordinates of the event vector γ, as defined in (3).

Thus, the weighted centroid vector $\boldsymbol{x}_\theta$ of a cluster in a given ROI is

$$\boldsymbol{x}_\theta = \left(\frac{\boldsymbol{M}_1}{M_0} \right) \tag{9}$$

Once the centroid is calculated, a new ROI is established around it, and a new centroid vector $\boldsymbol{x}_\theta$ is calculated with respect to the new ROI. This procedure is repeated until the distance between said centroid $\boldsymbol{x}_\theta$ and the centroid of the previous iteration (call it $\boldsymbol{x}'_\theta$) is less than a previously established threshold δ_m; that is,

$$\| \boldsymbol{x}_\theta - \boldsymbol{x}'_c \|_2 < \delta_m \tag{10}$$

where $\| * \|_2$ denotes the 2-norm operation [63]. At such point, vector $\boldsymbol{x}_\theta$ is established as the new centroid of object $\boldsymbol{\theta}$ and the latest estimate of zeroth moment M_0 is its total influence.

E. CALCULATION OF THE DIMENSION OF AN OBJECT'S ROI

Once the centroid has been calculated, the system is extended by applying the ideas of [27]. The objective is to ensure that the search window (which matches the object's ROI) is adapted to the level of activity of correlated events appearing in its environment within a time frame Γ. The window thus becomes larger in extent if there is a large region of correlated activity and becomes smaller otherwise. In this way the window adapts to the size of the object, understanding it as a correlated activity set in a given region (Fig. 4b).

The dimensions of an object's ROI are calculated based on the estimate of its zeroth moment, M_0. This estimate is normalized by an *ad hoc* parameter selected by the user so that the resulting units correspond to those of the intended ROI.

The number of events generated by an object can cause problems when calculating an ROI, as the density of events depends on object parameters such as size, speed, aspect, etc. Thus, a large but slow object can produce a similar number of events as a fast but smaller object, even though their ROIs are very different. The latter leaves a trail of events that in a time frame Γ can model a ROI larger than what the object really

is and can cause unwanted merging of multiple small objects in the vicinity, resulting in the declaration of a single massive object rather than in a constellation of smaller ones.

On the other hand, a large object can be mistaken for a smaller one and its ROI never reach the size needed to frame the object completely. This could result in mistakenly declaring as smaller objects what really would be the local maxima of nearby events, in turn leading to the erroneous conclusion that several small objects are moving closely to each other.

Assuming that ROIs are square in shape, of size $l_{ROI} \times l_{ROI}$, a way to alleviate these issues is to use

$$l_{ROI} = \tau \left(\frac{M_0}{f} \right)^{1/2} \tag{11}$$

where τ and f are scalar parameters.

At each iteration of a centroid calculation, a maximum local density value is obtained. Setting f in (11) equal to this maximum, with $\tau = 2$, offers a good overall compromise for estimating l_{ROI} [27]. An adequate ROI is considered to have been obtained whenever the magnitude of the difference between the current estimate of M_0 and the previous one (call it M'_0) is smaller than a preset threshold δ; that is,

$$|M_0 - M'_0| < \delta \tag{12}$$

It is interesting to note that unlike classic systems, the proposed framework can handle objects that move at a very high speed, when frames are blurred and it is difficult to calculate a ROI in an accurate fashion.

If one wants to track objects that have a certain shape, one can set a specific proportionality constant for each axis. For example, to track human models (Fig. 6) one could use specific axis values like

$$l_{ROI}(x) = \frac{l_{ROI}}{2} \tag{13a}$$

$$l_{ROI}(y) = 2 l_{ROI} \tag{13b}$$

Furthermore, if the goal were to track objects undergoing rotations and/or stretching, one could compute the second order statistics of the ROI events and use the eigenvectors and eigenvalues of the resulting covariance matrix to estimate such changes over time. Since we are talking about 2×2 covariance matrices, closed-form expressions exist for such eigenvectors and eigenvalues, making their calculation straightforward and inexpensive.

The pseudocode encompassing the functioning of the algorithm described in subsections III-D and III-E is summarized in Algorithm 1. The input of the system at a given time t, comprises a θ object, the relevant set of activity (according to expression 6), and custom threshold values δ and δ_m.

F. MERGING AND DIVISION OF OBJECTS

As the objective of the proposed algorithm is to track multiple freely-moving objects, situations may occur in which two or more objects collide, intersect, split or overlap as, for example, when

$$\| \boldsymbol{x}_{\theta_1} - \boldsymbol{x}_{\theta_2} \| \leq \frac{l_{ROI_1} + l_{ROI_2}}{2} \tag{14}$$

where $\boldsymbol{x}_{\theta_1}$ and $\boldsymbol{x}_{\theta_2}$ are the centroids of objects $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ (see expression (4)).

Two main situations have now to be faced. One is when two objects satisfy (14); the other is when, in addition to satisfying (14), the centroid of one object lies within the ROI of the other object (Fig. 4d); that is,

$$\boldsymbol{x}_{\theta_1} \subset ROI_{\theta_2} \tag{15a}$$

$$M_{0_{\theta_1}} < M_{0_{\theta_2}} \tag{15b}$$

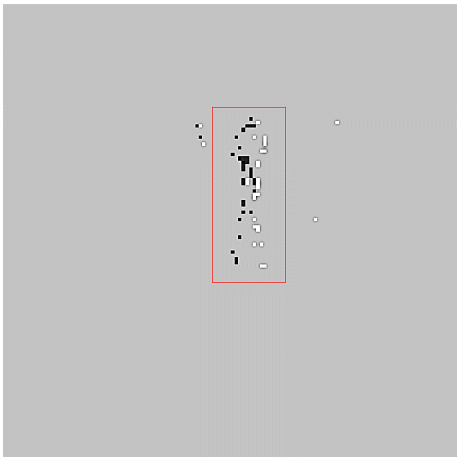

FIGURE 6. Events generated by a pedestrian. By adjusting the value of l_{ROI}, arbitrary shaped objects, such as this human silhouette, can be efficiently detected and tracked.

Algorithm 1 Calculation of the Centroid and the Dimension of the ROI of an Object

Input: $\boldsymbol{\theta} \begin{bmatrix} id_\theta \\ \boldsymbol{x}_\theta \\ t_\theta \\ M_0^\theta \end{bmatrix}, \{\boldsymbol{\gamma} \begin{bmatrix} \boldsymbol{x}_\gamma \\ t_\gamma \end{bmatrix} : \phi(t, t_\gamma) > 0\}, \delta_m, \delta, t$

$\boldsymbol{x}_c \leftarrow \boldsymbol{x}_\theta$
$M_0 \leftarrow M_0^\theta$
repeat
  $M'_0 \leftarrow M_0$
  repeat
    $\boldsymbol{x}'_c \leftarrow \boldsymbol{x}_c$
    $\boldsymbol{M}_0 = 0$
    for all $\gamma \in ROI_\theta$ **do**
      $M_0 = M_0 + \phi(t, t_\gamma)$
      $\boldsymbol{M}_1 = \boldsymbol{M}_1 + \phi(t, t_\gamma)\boldsymbol{x}_\gamma$
    end for
    $\boldsymbol{x}_c \leftarrow (\frac{\boldsymbol{M}_1}{M_0})$
  until $\| \boldsymbol{x}_\theta - \boldsymbol{x}'_c \|_2 < \delta_m$
until $|M_0 - M'_0| < \delta$
$\boldsymbol{x}_\theta \leftarrow \boldsymbol{x}_c$
$M_0^\theta \leftarrow M_0$

FIGURE 7. On the left is the high-speed movement of an object on the silicon retina. The centroid trajectory of the object in the last few moments is shown in red. On the right is the correspondence of this object on a frame-based sensor. It can be seen that the frame-based sensor struggles when fast moving objects are in the scene. Meanwhile, silicon retina can easily detect the object position and trajectory.

This second case usually occurs when certain activity is emerging in an object (new events are appearing) while the ROI is being formed. As events arrive in a disorderly fashion, small local minima may arise (Fig. 4c) that will gradually be absorbed by the object that ends up being the dominant one. In such cases, convention establishes that the object with the greatest influence (largest zeroth-order moment M_0) will be the one that prevails, eliminating the rest (Fig. 4e).

Alternatively, when an object is divided into multiple ones, the very nature of the algorithm ends up generating an object almost instantaneously as the splinter object moves away from the original one. Since no a priori information is known about the objects, it cannot be guaranteed that their individual identities will be recognized in this situation.

IV. RESULTS

To validate the performance of BioCAMSHIFT, multiple tests were carried out in different scenarios, in order to test the various aspects of the algorithm.

A. STUDY CASE

1) EXPERIMENT 1: SIMPLE OBJECTS

Survey data was collected using a hybrid vision system that involved a silicon retina paired with a frame-based sensor consisting of a 320 × 240 pixel array. The images obtained from the conventional frame-based sensor were used to determine the ground truth, while the silicon retina was employed to generate a series of data files involving single and multiple objects with particular conditions such as lighting fluctuations, abrupt speed changes, and merging, division and occlusion of objects (Fig. 7).

The main objective of this experimentation was to determine the behavior of the algorithm, evaluate its performance in various situations, analyze potential problems and pitfalls, and for early benchmarking purposes. It was specifically designed for tracking simple objects in a static environment.

A first test was carried out to analyze the behavior of the algorithm. The objective was to perform detection and tracking of a single object in the scene. It was verified that the behavior of the algorithm was adequate, a correctly detecting and tracking of the object. Likewise, as shown in Fig. 8a, after an initial high computational time peak, once a cluster was defined, the algorithm was generally stable, with a low use of the processor, except when burst of spikes due to noise or abrupt changes in speed which generated events to be processed.

A second test was conducted with the objective of calculating the impact of noise in algorithm performance. It consisted of multiple objects moving in a noisy environment. An event filter was defined as in III-B and as can be seen from comparing raw (Fig. 8b) and filtered data (Fig. 8c) it significantly reduced the average computation time in each iteration, and increased the overall precision and recall by ignoring noise-generated events occurring near the object. Figure 9, shows how the filter is capable of eliminating noisy data in an efficient fashion. Figure 8b shows the higher impact of execution time per iteration in the noisy scenario versus the filtered one in 8c, where despite processing the same amount of events, the requirements in nanoseconds per event and in overall system execution time, are significantly lower in the latter.

Finally, a third test involved tracking an object in a low light environment, the main objective being evaluating system performance when abrupt high-speed and direction changes take place. As shown in Fig. 8d, thanks to the particular ability of the silicon retina to capture intensity changes, the change in illumination did not significantly affect the performance of the algorithm. Furthermore, despite moving the object at a higher speed, the algorithm was able to track it efficiently. It was also observed that BioCAMSHIFT achieved better performance by adjusting Γ in equation (6) to counteract the greater number of events generated, discarding valid but older activity.

This first phase of experimentation allowed valuable conclusions to be drawn about the algorithm. The most important value for ensuring a proper tracking is the Γ used when calculating the life function $\phi(t, t_\gamma)$ in (6), as it allows managing the activity of an object and calibrating the size of the ROI. It enables adjusting the clusters to suit the activity in an ROI, be it that of a fast object that generates many events or of a slow, large object that generates a similar response activity. The other parameter to take into account is τ in (11), as it allows adjusting the ROI to the shape formed by the events' activity in the search window.

2) EXPERIMENT 2: CARS PASSING UNDER BRIDGE

We chose the open access DVS09 data set [64] to benchmark BioCHAMSHIFT, particularly so with respect to object-tracking. For this purpose we selected a sample of cars passing under a bridge over Freeway 210 in Pasadena, California, under late afternoon natural lightning conditions, as this data provides a real-life situation. The sequence was manually analyzed, with every vehicle position annotated for performance evaluation.

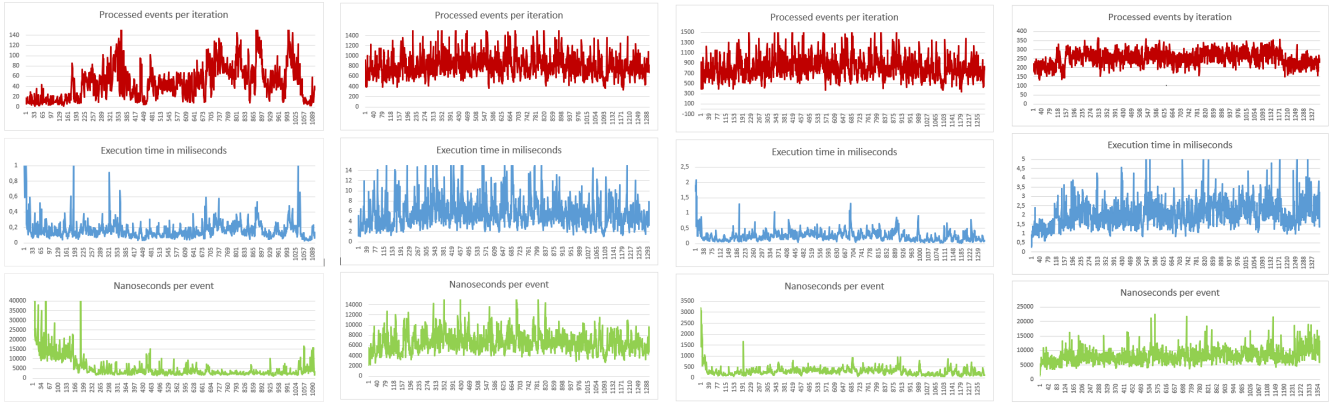

FIGURE 8. Benchmarking graphs of the case studies in Experiment 1. Test 1: Detection and tracking of a single object in standard noise/lightning conditions (Fig. 8a). Test 2: Detection and tracking of multiple objects in (Fig. 8b) noisy conditions and (Fig. 8c) applying the $1-out-of-N$ consistent-information filter (section III-B). Test 3: Detection and tracking of a single object in noisy conditions, with low illumination and abrupt speed changes (Fig. 8d).

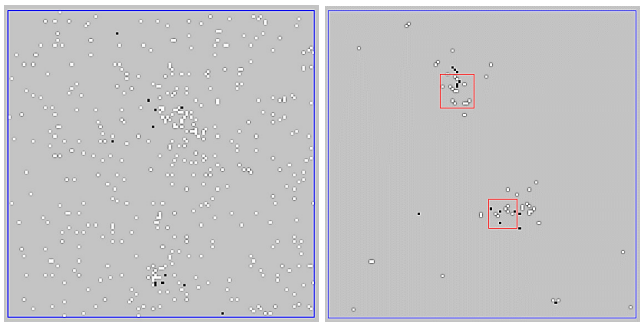

FIGURE 9. The image on the left depicts the movement of two objects in a noisy environment. The image on the right shows the noise is suppressed after applying the $1-out-of-N$ consistent information filter (III-B).

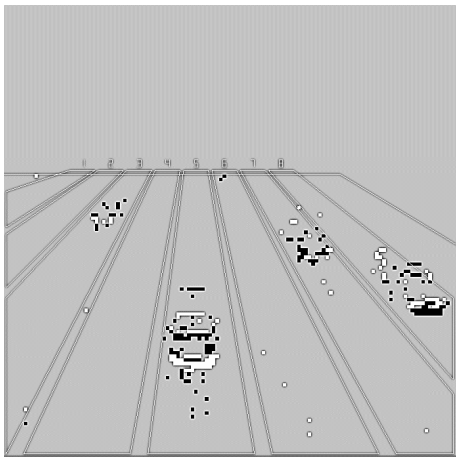

FIGURE 10. In order to specify a ground truth, vehicle trajectories in the DVS09 data set were analyzed and a series of regions of interest were defined to help identify the different vehicles transiting along the roadway and thus enable manually labelling and annotating objects.

In the absence of either ground truth or frame-based video corresponding to the neuromorphic view, multiple ROIs were defined manually on the screen, limiting the highway to lanes where vehicles could be distinguished (Fig. 10). Vehicles were labeled by both, type (automobile, van, truck) and lane in which they were circulating (Fig. 13). The right-hand side and the top portions of the images were excluded, as we were not able to visually tag these labels with the desired precision. The overall aim was to enable identifying and addressing specific issues that could arise in particularly challenging situations, such as when vehicles are changing lanes.

We used the F_1 score, defined as the harmonic mean of the model's precision and recall, to measure algorithm performance. Recall is defined as the number of true positives divided by number of true positives plus the number of false negatives (i.e., the number of true positives per actual, real positive). Precision, on the other hand, is defined as the number of true positives divided by the number of true positives plus the number of false positives (i.e., the number of true positives for each predicted positive).

The results in Table 1 show high values of detection and tracking of vehicles in the scene. Cars have a very high precision indicating that there are hardly any false positives. On the other hand, vans and trucks suffer from a higher presence of false positives, which results in a somewhat lower precision. Likewise, the number of false negatives evens out between different types of vehicles. Such false negatives tend to occur when the vehicles are in the distance, as they appear extremely small to the camera and thus fail to generate events due to their size. Their level of activity is thus not considered sufficient by the algorithm to generate new clusters defining a valid object in the scene. However, as the vehicles approach the sensor, their size increases, generating a larger number of events that allow the algorithm to define them as objects and do so with a high degree of certainty.

B. DISCUSSION

The results obtained from our case studies show that Bio-CAMSHIFT is capable of consistently detecting and tracking

TABLE 1. Experimental results.

Sequence	Recall	Precision	F_1 score
Cars	89.31%	**98.30%**	**93.59%**
Trucks & Vans	**93.68%**	84.04%	88.60%
Complete	**90.15%**	**95.06%**	**92.54%**

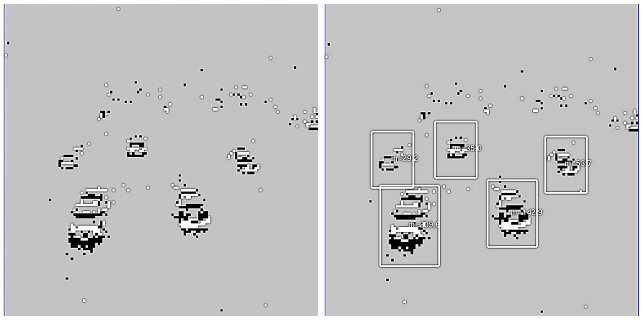

FIGURE 11. The image on the left shows a scene from the DVS09 database at a given instant. On the right, the result of running BioCAMSHIFT is presented. Under conventional circumstances, BioCAMSHIFT is able to effectively and consistently detect multiple vehicles on the road.

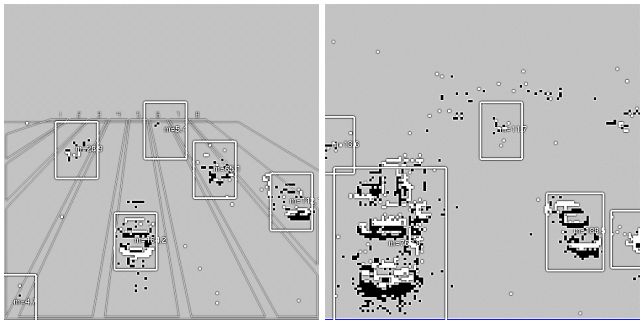

FIGURE 12. In the left panel, the cluster at the bottom-left corner corresponds to the exhaust fumes from a vehicle outside the frame; enough activity was created by such fumes to elicit the creation of such object. In the right panel, at the bottom-left, a truck is hauling a container of uniform intensity. The high-pass filtering action of the silicon retina effectively erases such container, stopping the generation of events in the central area, causing the BioCAMSHIFT algorithm to separately declare an object for the cab (the bottom-left rectangle) and another for the rear of the container (the rectangle at the center of the panel). In addition, the size of the truck cab partially obscures a vehicle traveling close to it in the adjacent lane (left-hand-side rectangle).

multiple objects in environments where the camera is stationary. Despite not being specifically modeled, vehicles are identified as such with a high rate of correctness (Fig. 11).

The aspects that most affect the results are the specification of the object of interest and the shape of its ROI. Regarding specifying the object, we opted for a generic approach and obtained good results with cars. However, the algorithm struggles with vehicles where long non-changing surfaces prevent the sensor from generating new events, such as truck trailers or the roofs of vans, as the high-pass filtering function of the silicon retina sensor will suppress them. Since our object definition process implicitly assumes that events generated by an object are at a reasonable distance from its centroid, large objects of a uniform color, such as truck trailers, will not generate events as they won't exhibit changes in intensity (Fig. 12). The detection of the corresponding object ends up being conditioned by the presence of events solely at the front or at the edges of the object. This can be seen in Table 1, where trucks and vans have somewhat lower F_1 scores, limited as they are by sensor characteristics and the generality of the object model. Also, due to its size, a truck may partially block smaller vehicles in adjacent lanes, causing the algorithm to detect them as a single object or alternatively to indeed consider them as separate objects but defining the visible portion of the smaller vehicle as a complete object in itself (Fig. 12). A trade-off study is thus required regarding the cost/benefits of using a simple, basic, general object model versus more complicated versions and/or their combination.

Another aspect that conditions the object detection and tracking results is the ROI shape used to define an object boundary. In this paper we chose rectangles for the simple reason that they adapt to the shape of the object based on the number of events and their arrangement on the plane. However, since the sides of these rectangles are defined parallel to those of the image, whenever a vehicle uses a side lane it will not be properly aligned with them, generating a larger region of interest. Hence, when two vehicles are very close to each other in the external lanes, the estimated ROI for any given vehicle becomes distorted and ends up encompassing both cars.

The algorithm could thus be combined with other solutions in order to solve the car tracking problem in particular. For example, it could use information about the lane in which the car is traveling to better segment the events or to select shapes other than rectangles (e.g., adaptively-constructed trapezoids) to define the ROIs. But the spirit of this research was to test the strengths of BioCAMSHIFT as a generic application-independent algorithm and to analyze its behavior.

More fundamental however may be the inclusion of means for improving performance by avoiding creating false objects that the system will have later to discard. The appearance of spurious events is inevitable and although the number of those due to random noise can be reduced with the $1-out-of-N$ consistent-information filter, the silicon retina can detect other elements (''clutter'') that are not necessarily part of a vehicle. For instance, some vehicles had ''fog-like'' events in their trail that we speculate were due to vehicle exhaust (Fig. 12). These ''false events'' interfere with the correct demarcation of an object, particularly when it comes to discerning whether a vehicle has completely abandoned the image limits. Means for the general characterization and suppression of such ''clutter'' should thus be considered and inserted at various points - from the detection to the high-level recognition and response-management stages – of a bioinspired processing string. Of course, these would constitute highly application-dependent processes that could nonetheless begin by marrying raw sensor capabilities with a simple, general statistical model as that of this paper (note that by

TABLE 2. Comparison of bioCAMSHIFT results and those of other methods found in the literature for analogous object detection applications.

Algorithm	Recall	Precision	F_1 score	Dataset	Objects
BioCAMSHIFT	0.9015	0.9506	0.9254	DVS09	vehicles
GMM [46]	0.9862	0.7872	0.8755	*ad hoc* (1)	pedestrians
DBSCAN [17]	0.628	0.645	0.6364	*ad hoc* (2)	vehicles
Meanshift [17]	0.466	0.407	0.4316	*ad hoc* (2)	vehicles
WaveCluster [17]	0.631	0.644	0.6374	*ad hoc* (2)	vehicles

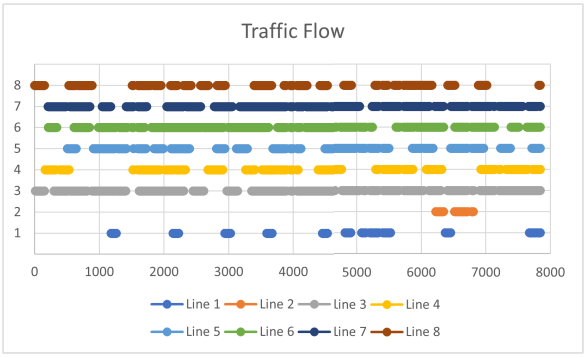

FIGURE 13. The figure shows the occupancy of vehicles in the lanes as shown in Figure 10 as a function of time. It can be seen how the left lanes, lanes 1 and 2, are the least traveled. Also, it can be seen that there is a lot of vehicle traffic, and that BioCAMSHIFT has had to be constantly detecting multiple objects in the scene.

virtue of its high-pass filtering capability, the silicon retina implicitly performs a clutter-filtering operation: the elimination of stationary events from consideration).

Finally, it must be underscored that one of the main problems researchers face when evaluating/benchmarking bio-inspired systems is the lack of widely-accepted large, annotated, open-access neuromorphic -vision databases [61]. Also, despite some data sets having been made available [16], [17], there is still a lack of comparative studies [62].

Results of other studies carried along lines of research closest to the work presented here are shown in Table 2. It can be seen that the performance of the proposed algorithms was lower than that of BioCAMSHIFT. On one hand, [46] forms clusters that can be modeled by Gaussian Mixtures (GMM), but suffers from higher false positive rates. On the other hand, [17] segments the continuous flow of events into frames in order to apply conventional algorithms of classical computer vision. They evaluate three classical clustering approaches: mean-shift clustering (MeanShift), density based spatial clustering of applications with noise (DBSCAN), and WaveCluster. They acknowledge that their work has some shortcomings, especially when noisy scenes are involved, and stress the importance of developing new vision algorithms that take direct advantage of the benefits of neuromorphic sensors.

V. CONCLUSION

Event-based computer vision offers multiple possibilities to researchers and an interesting alternative to conventional computer vision. The high temporal resolution at which information is obtained, as well as the absence of redundant information, make it possible to create solutions capable of operating efficiently in real time. In addition, using embedded systems such as silicon retinas, allows processing to take place in the sensor itself and to do so with low memory and power consumption requirements.

The main challenge nowadays is the paradigm shift of event-based computer vision with respect to classical computer vision as it forces to redesign and implement new algorithmic solutions to solve problems widely discussed in the literature. Part of the challenge may arise from the need of custom sensors and the lack of widely accepted freely accessible databases. Besides, although there are some databases available, they lack sufficient standardization to become benchmarks to ensure any progress in the field. In addition, the lack of publication of results prevents an evaluation process of the different algorithms being proposed. Data corpora, feature extraction, classification algorithms and results are too dispersed, and it is very difficult to make comparisons among them to check the progress in the state of the art.

The main contribution of this paper is the cluster formation and tracking algorithm we have called BioCAMSHIFT. Most cluster-tracking algorithms described in the literature process events in a per-frame fashion. Here we have proposed instead a new event-based object-tracking algorithm, evaluating it with data from a bio-inspired silicon retina. Compared to conventional frame-based tracking-by-clustering algorithms, the proposed bio-inspired system does not use redundant information (it is eliminated by the silicon retina's built-in high-pass temporal filtering function), it exploits the retina's high temporal resolution (basically operates on a temporal continuum), and does not need specific lighting. The proposed algorithm provided good detection and tracking performance, doing so efficiently by benefiting from the sparse, non-redundant activity generated by the silicon retina.

BioCAMSHIFT is the first proposal that has been made to develop a CAMSHIFT-type method that is purely bio-inspired, following a biological approach from the first step (silicon retina), through the whole process and up to the final result. In particular, BioCAMSHIFT differs from other current state-of-the-art algorithms in that it does not discard past information that may prove valuable for understanding an object's behavior (e.g., to analyze and determine complex trajectories under partial occlusion). Being an unsupervised algorithm able to adapt to the size and number of objects being tracked, it is capable of detecting and correcting

situations such as the merging and splitting of objects, quickly and efficiently. Finally, it has the added benefit of its simplicity in terms of the parameters being customized, requiring few modifications to optimize them for a specific situation.

The algorithm has proven to be robust in tracking multiple objects, even with abrupt changes in speed and illumination and in the presence of noise. This was demonstrated by detecting and tracking road vehicles using real data from a freely available database [56]. BioCAMSHIFT was able to deliver excellent results. Moreover, since unlike current procedures where the temporal image data is divided into frames, BioCAMSHIFT operates on individual ''events'' as they appear, it is able to work directly on the timestream continuum with minimal storage and processing requirements. The results obtained by BioCAMSHIFT have been compared with others in the literature [17], [46], obtaining promising results with an F_1 score of 92.54% and far surpassing (Table 2) the results obtained in analogous works.

Event cameras have proven their usefulness in a multitude of applications such as object tracking, surveillance and monitoring, object recognition and gesture control [16]. BioCAMSHIFT is presented as an alternative to current solutions. Part of its contribution is its ability to take advantage of previous events in the scene to quickly generate clusters without the need to be aware of the evolution of the activity and thus achieve a quick understanding of the scene at a given time. This improves object detection and tracking performance over that of currently used methods such as those in [46] and [17]. Perhaps more importantly, this paper diverges from current current methods in that it does not operate on ''frames''. In such frame-based methods, events occurring during a time window are operated on as if they were part of a single, instantaneous data-snapshot, irrespective of their actual order of appearance. In this paper we operate instead on the actual event data as it appears, thus maximally exploiting temporal information, doing it so to speak, in ''real-time''. This capability is enabled by the introduction of the life function of expression 6, which allows managing the data by discarding events whose life has extended beyond some expiration date.

Future research will involve making BioCAMSHIFT fully automatic in the sense of enabling it to autonomously adapt not only to changes in object size, orientation and perspective (e.g., a truck changing from frontal to side view), but to be able to identify and adapt to different event-flow conditions (e.g., so as to differentiate large lumbering objects from small nimble ones). Future research will also explore the introduction and application of ''life-functions'' at various stages of the BioCAMSHIFT processing string so as to properly exploit the opportunities granted by operating in the time continuum. At a minimum, this could enable incorporating multiple conventional computer vision algorithms without losing information on the timing of events, as has been the case up to now. Most importantly, however, it could enable the extraction and implicit preservation of significant past (e.g., ''track-like'') information, allowing BioCAMSHIFT to better analyze the behavior of objects and their evolution over time in cases of their occlusion, merging or division. Dynamic Programming-like concepts will be investigated for this purpose, as they can operate directly on the time continuum and perform the processing at the most primitive signal level [65]. This would enable maintaining all the processing within the sensor, thus providing a conceptual simplicity more aligned with that of a bio-inspired system than would be if one had used instead high-level data-driven processes such as Kalman filters or Maximum-Likelihood classifiers.

In conclusion, the paradigm-shift introduced by going from classical to event-based computer vision processing may enable addressing important computer vision problems that have remained intractable through conventional means. Achieving this potential will involve the development and implementation of new concepts and models from where to specify and design the proper sensors and algorithms. Bio-inspired sensors and processes will certainly play a significant role here. And of course, we will also need to create widely-accepted, freely accessible databases to enable comparing the various sensors and algorithms that may be slowly coming of age. The few such databases that are currently available lack the sufficient level of standardization required to be used as benchmarks, making it very difficult to establish comparisons and do proper evaluations. Hopefully new progress will be made in all these areas to promote the advancement of this very interesting, innovative and highly-promising field.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] C. Mead, ''Neuromorphic electronic systems,'' *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.
[2] G. Indiveri and T. K. Horiuchi, ''Frontiers in neuromorphic engineering,'' *Frontiers Neurosci.*, vol. 5, p. 118, Oct. 2011. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2011.00118
[3] D. Marr, *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. New York, NY, USA: Henry Holt and Company, 1982.
[4] C. A. Mead and M. A. Mahowald, ''A silicon model of early visual processing,'' *Neural Netw.*, vol. 1, no. 1, pp. 91–97, 1988.
[5] J. Kramer, ''An on/off transient imager with event-driven, asynchronous read-out,'' in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2, May 2002, p. 2.
[6] M. A. Sivilotti, ''Wiring considerations in analog vlsi systems, with application to field-programmable networks,'' Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, 1991.
[7] M. Mahowald, ''Vlsi analogs of neuronal visual processing: A synthesis of form and function,'' Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, 1992.
[8] M. Mahowald, ''Analog VLSI chip for stereocorrespondence,'' in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 6, May/Jun. 1994, pp. 347–350.
[9] P. Lichtsteiner, C. Posch, and T. Delbruck, ''A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor,'' *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Jan. 2008.
[10] J. A. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, ''A 3.6 μs latency asynchronous frame-free event-driven dynamic-vision-sensor,'' *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1443–1455, Jun. 2011.

[11] T. Finateu, A. Niwa, D. Matolin, K. Tsuchimoto, A. Mascheroni, E. Reynaud, P. Mostafalu, F. Brady, L. Chotard, F. LeGoff, H. Takahashi, H. Wakabayashi, Y. Oike, and C. Posch, "A 1280×720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μm pixels, 1.066 GEPS readout, programmable event-rate controller and compressive data-formatting pipeline," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 112–114.

[12] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB dynamic range asynchronous address-event PWM dynamic image sensor with lossless pixel-level video compression," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2010, pp. 400–401.

[13] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 130 dB 3 μs latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, Oct. 2014.

[14] H. Li, J. Pei, and G. Li, "Real-time tracking based on neuromrophic vision," in *Proc. 15th Non-Volatile Memory Technol. Symp. (NVMTS)*, Oct. 2015, pp. 1–7.

[15] M. Litzenberger, C. Posch, D. Bauer, A. N. Belbachir, P. Schon, B. Kohn, and H. Garn, "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *Proc. IEEE 12th Digit. Signal Process. Workshop, 4th IEEE Signal Process. Educ. Workshop*, Sep. 2006, pp. 173–178.

[16] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022, doi: 10.1109/TPAMI.2020.3008413.

[17] G. Chen, H. Cao, M. Aafaque, J. Chen, C. Ye, F. Röhrbein, J. Conradt, K. Chen, Z. Bing, X. Liu, G. Hinz, W. Stechele, and A. Knoll, "Neuromorphic vision based multivehicle detection and tracking for intelligent transportation system," *J. Adv. Transp.*, vol. 2018, pp. 1–13, Dec. 2018.

[18] R. N. Bracewell, *The Fourier Transform and Its Applications*. New York, NY, USA: McGraw-Hill, 2000.

[19] M. Litzenberger, D. Bauer, N. Donath, H. Garn, B. Kohn, C. Posch, and P. Schön, "Embedded vehicle counting system with 'silicon retina' optical sensor," *AIP Conf.*, vol. 860, no. 1, p. 360, 2006. [Online]. Available: https://www.overleaf.com/project/5dd8005a5d14370001d332c9

[20] M. Litzenberger, B. Kohn, A. N. Belbachir, N. Donath, G. Gritsch, H. Garn, C. Posch, and S. Schraml, "Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2006, pp. 653–658.

[21] M. Litzenberger, A. N. Belbachir, P. Schön, and C. Posch, "Embedded smart camera for high speed vision," in *Proc. 1st ACM/IEEE Int. Conf. Distrib. Smart Cameras*, Sep. 2007, pp. 81–86.

[22] M. Litzenberger, B. Kohn, G. Gritsch, N. Donath, C. Posch, N. A. Belbachir, and H. Garn, "Vehicle counting with an embedded traffic data system using an optical transient sensor," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2007, pp. 36–40.

[23] T. Delbruck and P. Lichtsteiner, "Fast sensory motor control based on event-based hybrid neuromorphic-procedural system," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 845–848.

[24] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Inf. Theory*, vol. IT-21, no. 1, pp. 32–40, Jan. 1975.

[25] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

[26] D. Comaniciu and V. Ramesh, "Mean shift and optimal prediction for efficient object tracking," in *Proc. Int. Conf. Image Process.*, vol. 3, Sep. 2000, pp. 70–73.

[27] G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in *Proc. 4th IEEE Workshop Appl. Comput. Vis. (WACV)*, Oct. 1998, pp. 214–219.

[28] T. Delbrück and M. Lang, "Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor," *Front. Neurosci.*, vol. 7, p. 223, Nov. 2013.

[29] S. Schraml, P. Schön, and N. Milosevic, "Smartcam for real-time stereo vision—Address-event based embedded system," in *Proc. 2nd Int. Conf. Comput. Vis. Theory Appl.*, vol. 2, 2007, pp. 466–471.

[30] S. Schraml, A. N. Belbachir, N. Milosevic, and P. Schön, "Dynamic stereo vision system for real-time tracking," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2010, pp. 1409–1412.

[31] L. A. Camuñas-Mesa, T. Serrano-Gotarredona, S.-H. Ieng, R. Benosman, and B. Linares-Barranco, "Event-driven stereo visual tracking algorithm to solve object occlusion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4223–4237, Sep. 2018.

[32] F. Barranco, C. Fermuller, and E. Ros, "Real-time clustering and multi-target tracking using event-based sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 5764–5769.

[33] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME-J. Basic Eng.*, vol. 82, pp. 35–45, 1960.

[34] F. Gómez-Rodríguez, L. Miró-Amarante, F. Diaz-del-Rio, A. Linares-Barranco, and G. J. Robotics, "Real time multiple objects tracking based on a bio-inspired processing cascade architecture," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 1399–1402.

[35] D. Drazen, P. Lichtsteiner, P. Häfliger, T. Delbrück, and A. Jensen, "Toward real-time particle tracking using an event-based dynamic vision sensor," *Exp. Fluids*, vol. 51, no. 5, p. 1465, 2011.

[36] S. Schraml and A. N. Belbachir, "A spatio-temporal clustering method using real-time motion analysis on event-based 3D vision," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.-Workshops*, Jun. 2010, pp. 57–63.

[37] A. Linares-Barranco, F. Gómez-Rodríguez, V. Villanueva, L. Longinotti, and T. Delbrück, "A USB3.0 FPGA event-based filtering and tracking framework for dynamic vision sensors," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 2417–2420.

[38] A. Linares-Barranco, F. Perez-Peña, D. P. Moeys, F. Gomez-Rodriguez, G. Jimenez-Moreno, S.-C. Liu, and T. Delbruck, "Low latency event-based filtering and feature extraction for dynamic vision sensors in real-time FPGA applications," *IEEE Access*, vol. 7, pp. 134926–134942, 2019.

[39] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, vol. 96, no. 34, pp. 226–231.

[40] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.

[41] B. N. Vo and W. K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.

[42] B. T. Vo, B. N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3553–3567, Jul. 2007.

[43] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Syst.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.

[44] Z. Fu, T. Delbrück, P. Lichtsteiner, and E. Culurciello, "An address-event fall detector for assisted living applications," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, no. 2, pp. 88–96, Jun. 2008.

[45] A. N. Belbachir, S. Schraml, and A. Nowakowska, "Event-driven stereo vision for fall detection," in *Proc. CVPR WORKSHOPS*, Jun. 2011, pp. 78–83.

[46] E. Piatkowska, A. N. Belbachir, S. Schraml, and M. Gelautz, "Spatiotemporal multiple persons tracking using dynamic vision sensor," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 35–40.

[47] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 381–396, Mar. 2002.

[48] G. Chen, L. Hong, J. Dong, P. Liu, J. Conradt, and A. Knoll, "EDDD: Event-based drowsiness driving detection through facial motion analysis with neuromorphic vision sensor," *IEEE Sensors J.*, vol. 20, no. 11, pp. 6170–6181, Jun. 2020.

[49] G. Chen, Z. Xu, Z. Li, H. Tang, S. Qu, K. Ren, and A. Knoll, "A novel illumination-robust hand gesture recognition system with event-based neuromorphic vision sensor," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 2, pp. 508–520, Apr. 2021.

[50] G. Chen, P. Liu, Z. Liu, H. Tang, L. Hong, J. Dong, J. Conradt, and A. Knoll, "NeuroAED: Towards efficient abnormal event detection in visual surveillance with neuromorphic vision sensor," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 923–936, 2020.

[51] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. J. Douglas, and T. Delbruck, "A pencil balancing robot using a pair of AER dynamic vision sensors," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2009, pp. 781–784.

[52] J. Conradt, R. Berner, M. Cook, and T. Delbruck, "An embedded AER dynamic vision sensor for low-latency pole balancing," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops, ICCV Workshops*, Sep. 2009, pp. 780–785.
[53] Z. Ni, C. Pacoret, R. Benosman, S. Ieng, and S. Régnier, "Asynchronous event-based high speed vision for microparticle tracking," *J. Microsc.*, vol. 245, no. 3, pp. 236–244, Mar. 2012.
[54] X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, "Asynchronous event-based multikernel algorithm for high-speed visual features tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1710–1720, Aug. 2015.
[55] D. R. Valeiras, X. Lagorce, X. Clady, C. Bartolozzi, S.-H. Ieng, and R. Benosman, "An asynchronous neuromorphic event-driven visual part-based shape tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3045–3059, Mar. 2015.
[56] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "HATS: Histograms of averaged time surfaces for robust event-based object classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1731–1740.
[57] V. Padala, A. Basu, and G. Orchard, "A noise filtering algorithm for event-based asynchronous change detection image sensors on truenorth and its implementation on truenorth," *Frontiers Neurosci.*, vol. 12, p. 118, Mar. 2018.
[58] B. Ramesh, H. Yang, G. Orchard, N. A. Le Thi, S. Zhang, and C. Xiang, "DART: Distribution aware retinal transform for event-based cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 11, pp. 2767–2780, Nov. 2020.
[59] B. Li, H. Cao, Z. Qu, Y. Hu, Z. Wang, and Z. Liang, "Event-based robotic grasping detection with neuromorphic vision sensor and event-grasping dataset," *Frontiers Neurorobotics*, vol. 14, p. 51, Oct. 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnbot.2020.00051
[60] A. Renner, M. Evanusa, and Y. Sandamirskaya, "Event-based attention and tracking on neuromorphic hardware," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–8.
[61] C. Tan, S. Lallee, and G. Orchard, "Benchmarking neuromorphic vision: Lessons learnt from computer vision," *Frontiers Neurosci.*, vol. 9, p. 374, Oct. 2015.
[62] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck, "DVS benchmark datasets for object tracking, action recognition, and object recognition," *Frontiers Neurosci.*, vol. 10, p. 405, Aug. 2016.
[63] C. F. Van Loan and G. Golub, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.
[64] T. Delbruck, "Frame-free dynamic digital vision," in *Proc. Int. Symp. Secure-Life Electron.*, vol. 1, no. 1. Tokyo, Japan: Univ. Tokyo, Mar. 2008, pp. 21–26, doi: 10.5167/uzh-17620.
[65] M. F. Fernandez, A. Aridgides, and D. Bray, "Detecting and tracking low-observable targets using IR," *Proc. SPIE*, vol. 1305, pp. 193–206, Oct. 1990.

JULIO GUILLEN-GARCIA was born in Madrid, Spain. He received the B.S. degree in computer science from Universidad Nacional de Educación a Distancia (UNED), in 2012, and the M.S. degree in computer vision from Universidad Rey Juan Carlos (URJC), in 2014, where he is currently pursuing the Ph.D. degree in information and communication technologies. He has worked for several years in the private sector as a Consultant, and in the public sector, in projects at URJC and the Universidad Pública de Navarra (UPN), as a Researcher. He is currently an Assistant Professor with URJC, where he is also a member of the Face Recognition and Artificial Vision (FRAV) Research Group. His research interests include bioinspired systems, dendritic computation, multiplayer environments, artificial intelligence, and video games.

DANIEL PALACIOS-ALONSO was born in Madrid, Spain. He received the B.S. and M.S. degrees in computer science and the Ph.D. degree in advanced computation from Universidad Politécnica de Madrid (UPM), in 2009 and 2017, respectively. He worked as a Team Leader at a technological consulting firm for five years. Since 2013, he is a member of the Neuromorphic Speech Processing Laboratory, Center for Biomedical Technology. He is an Associate Professor with Universidad Rey Juan Carlos (URJC). He is currently the Head of the Bioinspired Systems and Applications Group (SA-BIO). His research interests include stress and emotional states, neurodegenerative diseases, such as Parkinson's, ALS, Alzheimer's, among others, artificial vision, pattern recognition, and biomedical signal processing. He is a reviewer of national and international journals. He was a recipient of several Best Paper Awards, including ICPRS-2016, BIOSIGNALS-2019, and JID-2020, and the Doctoral Consortium Award of the Spanish Association of Artificial Intelligence, in 2013.

ENRIQUE CABELLO (Member, IEEE) received the B.S. degree in physics (electronics) from the University of Salamanca and the Ph.D. degree from the Polytechnic University of Madrid. In 1990, he joined the Computer Science Department at the University of Salamanca. He joined Universidad Rey Juan Carlos, in 1998, where he has been the Head of the Face Recognition and Artificial Vision Group, since 2001. He is currently the Head of the Computer Science and Statistics Department. His research interests include image and video analysis, pattern recognition, and machine learning, using classic and bio-inspired approaches.

CRISTINA CONDE received the B.S. degree in physics (electronics) from the University Complutense of Madrid, in 1999, and the Ph.D. degree from University Rey Juan Carlos, Madrid, in 2006. She has worked for several years in the private sector and she joined as an Assistant Professor at University Rey Juan Carlos, in 2001. For seven years, she was the Vice-Dean of studies at the Computer Science School. She has coordinated several national and European projects. Her research interests include image and video analysis, pattern recognition, and machine learning in both, classical and biologically inspired computation.