# Event-Based Visual Flow

Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi

***Abstract*— This paper introduces a new methodology to compute dense visual flow using the precise timings of spikes from an asynchronous event-based retina. Biological retinas, and their artificial counterparts, are totally asynchronous and data-driven and rely on a paradigm of light acquisition radically different from most of the currently used frame-grabber technologies. This paper introduces a framework to estimate visual flow from the local properties of events' spatiotemporal space. We will show that precise visual flow orientation and amplitude can be estimated using a local differential approach on the surface defined by coactive events. Experimental results are presented; they show the method adequacy with high data sparseness and temporal resolution of event-based acquisition that allows the computation of motion flow with microsecond accuracy and at very low computational cost.**

***Index Terms*— Event-based vision, event-based visual motion flow, neuromorphic sensors, real time.**

## I. Introduction

RECENT work in this paper of neural activity has shown that each spike arrival time is reliable [1]–[4]. However, the extent to which the precise timing of neural spikes down to millisecond precision is significant for computation is still a matter of debate. In this paper, we address this issue by focusing on the computational principles that could be operated by motion-sensitive neurons of the visual system to compute the visual flow. We bring together new experimental sensors delivering truly naturalistic precise timed visual outputs and a new mathematical method that allows to compute event-based visual motion flow using each incoming spike's timing as main computation feature. This presented method does not rely on gray levels, nor on the integration of activity over long time intervals. It uses each relative timing of changes of individual pixel's responses to visual stimuli as a computational input.

This paper builds on recent achievements of neuromorphic engineering, exploiting the unique characteristics of a novel family of asynchronous image sensors [5]–[8]. The increasing availability and, most importantly, the steadily improving quality of these sensors open up the potential to introduce a shift in the methodology of acquiring and processing visual information in various demanding, machine vision applications [9]–[15]. The microsecond temporal resolution and the inherent redundancy suppression of the frame-free, event-driven acquisition and subsequent representation of visual information employed by these cameras enables to derive a novel methodology to process the visual information at a very high speed and with low computational cost.

Visual flow is a topic of several research fields that has been intensively studied since the early days of computational neuroscience. It is widely used in artificial vision and essential in navigation. Visual flow is known to be an ill-posed noisy visual measure limited by the aperture problem. Its use in real-time applications on natural scenes is generally difficult. It is usually computed sparsely on high salient points.

Visual flow techniques are commonly classified under one of the four major categories.

1) Energy-based or frequency-based methods estimate optical flow from the output of the velocity-tuned filters designed in the Fourier domain [16]–[18].
2) Phase-based methods estimate image velocity in terms of band-pass filter outputs [19].
3) Correlation-based or region-matching methods search for a best match of small spatial neighborhoods between adjacent frames [20]–[25].
4) Gradient-based or differential methods use spatiotemporal image intensity derivatives and an assumption of brightness constancy [26]–[28].

Energy-based techniques are slow [19] and are not adequate for real-time applications where gradient-based approaches perform better, as they rely on correlations. Visual flow is generally slow and does not exceed several Hertz on dense input. There are existing solutions to speed up the computation according to a tradeoff between accuracy and efficiency [29]. Preprocessing stages and kernel differentiation are often needed but they affect drastically real-time performance. In this case, accuracy is linked to the size of kernels that inevitably influences the execution time. If temporal kernels are used, then the buffering of images needed to perform computation dramatically increases the amount of stored data and introduces additional time delay in the computation.

The high computational cost of all of the approaches described previously are not suitable for real-time applications. Frame-based flow computation using large temporal windows is not compatible with the temporal precision of biological sensors that respond with 1 ms precision. The same observation applies to artificial vision that intrinsically remains linked to the frequency of the available cameras, generally not exceeding 60 Hz. Most of the developed techniques are computationally expensive and are mostly used OFF-line.

Manuscript received June 19, 2012; revised March 25, 2013; accepted June 22, 2013. Date of publication September 5, 2013; date of current version January 10, 2014. This work was supported by the European Grant eMorph under Grant ICT-FET 231467.

R. Benosman is with the Institut de la Vision, University Pierre and Marie Curie, Paris 75252 cedex 05, France, and also with the DIST, University of Genova, Genova 16121, Italy (e-mail: ryad.benosman@upmc.fr).

C. Clercq and C. Bartolozzi are with the Department of Robotics, Istituto Italiano di Tecnologia, Genoa 16163, Italy (e-mail: charles.clercq@iit.it; chiara.bartolozzi@iit.it).

X. Lagorce and S.-H. Ieng are with the Insitut de la Vision, University Pierre and Marie Curie, Paris 75252 cedex 05, France (e-mail: sio-hoi.ieng@upmc.fr; Xavier.Lagorce@crans.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2013.2273537

Comparing to the event-based optical flow calculation introduced in [10], the method introduced in this paper offers a new formulation which is entirely and only based on events' timing. The optical flow in [10] is obtained by adapting the differential flow brightness consistency constraint to event-based. Image intensities are then approximated by events' summations, as the dynamic vision sensor (DVS) does not provide absolute intensities. This paper introduces an alternative to flow computation that does not necessitate solving dynamic equations. It offers a pure event-based time-oriented computation of the motion flow within the focal plane.

## II. Neuromorphic Silicon Retina

Biological retinas, unlike frame-based cameras, transmit less-redundant information about a visual scene in an asynchronous manner. The various functionalities of the retina have been incorporated into neuromorphic vision sensors because the late 1980s in the pioneering paper of Mahowald [30]. Since then, the most interesting achievements in neuromorphic retinas have been the development of activity-driven sensing. The event-based vision sensors output compressed digital data in the form of events, removing redundancy, reducing latency, and increasing dynamic range as compared with conventional imagers. A complete review of the history and existing sensors can be found in [31]. The DVS used in this paper is an Address-Event Representation (AER) silicon retina with 128 × 128 pixels [5]. The DVS output consists of asynchronous address-events that signal scene reflectance changes at the times they occur. Each pixel is independent and detects changes in log intensity larger than a threshold since the last emitted event (typically 15% contrast). As shown in Fig. 1, when the change in log intensity exceeds a set threshold, an ON or OFF event is generated by the pixel depending on whether the log intensity increased or decreased. The advantages of such a sensor, over conventional clocked cameras, are that only moving objects produce data thus reducing the load of postprocessing. Additionally, the timing of events can be conveyed with very low latency and accurate temporal resolution of 1 $\mu$s. Thus the equivalent frame rate is typically several kiloHertz.

The encoding of log intensity of light change implements a form of local gain adaptation which allows them to work over scene illuminations that range from 2 lux to >100 klux. When events are transmitted off-chip, they are time-stamped and then transmitted to a computer using a standard universal serial port (USB) connection, in this case packets of events are received approximately every millisecond.

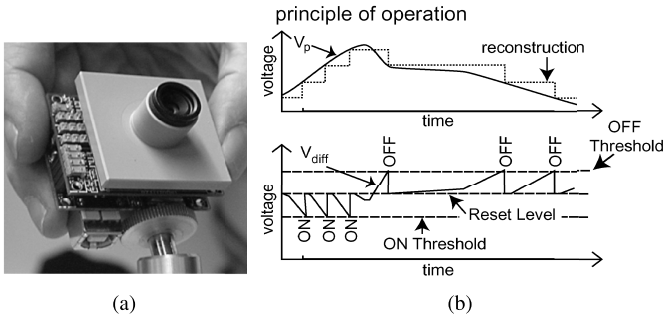



Fig. 1. (a) First generation DVS sensor with 128 × 128 pixels [5]. (b) Principle of ON and OFF spikes generation of DVS pixels, adapted from Lichtsteiner *et al.* [5]. Top: the evolution of pixel's voltage $V_p$ proportional to the log intensity. Bottom: the corresponding generation of ON (voltage increases above change threshold) and OFF (voltage decreases) events, from which the evolution of $V_p$ can be reconstructed.

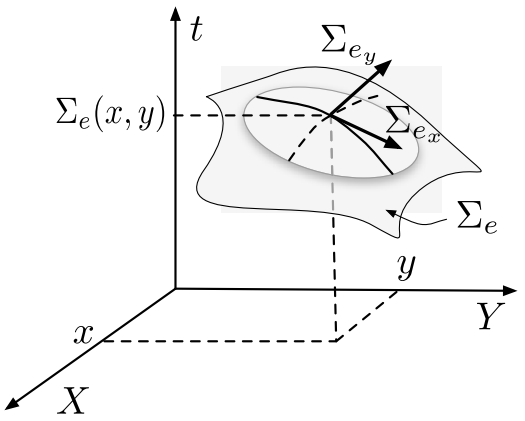



Fig. 2. General principle of visual flow computation. The surface of active events $\Sigma_e$ is derived to provide an estimation of orientation and amplitude of motion.

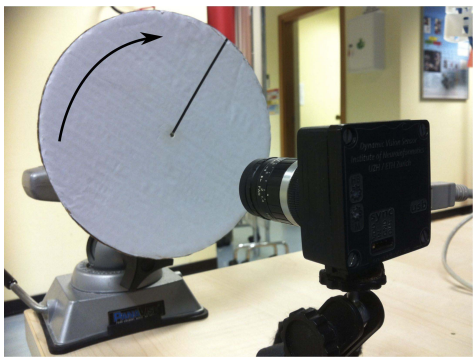


Fig. 3. Experimental setup consisting of a black bar printed on a white disk, driven by a dc motor.

## III. Event-Based Visual Motion Flow

### A. Flow Definition

The stream of events from the silicon retina can be mathematically defined as follows: let $e(\mathbf{p}, t) = (\mathbf{p}, t)^T$ a triplet giving the position $\mathbf{p} = (x, y)^T$ and the time $t$ of an event. We can then define locally the function $\Sigma_e$ that maps to each $\mathbf{p}$ (Fig. 2), the time $t$

$$\begin{array}{rcl} \Sigma_e : \mathbb{N}^2 & \rightarrow & \mathbb{R} \\ \mathbf{p} & \mapsto & \Sigma_e(\mathbf{p}) = t. \end{array} \tag{1}$$

Time is an increasing physical quantity, $\Sigma_e$ as defined here, is a monotonically increasing function of space.

We then set the first partial derivatives with respect to the parameters as: $\Sigma_{e_x} = \partial \Sigma_e / \partial x$ and $\Sigma_{e_y} = \partial \Sigma_e / \partial y$. We can then write $\Sigma_e$ as

$$\Sigma_e(\mathbf{p} + \Delta\mathbf{p}) = \Sigma_e(\mathbf{p}) + \nabla \Sigma_e^T \Delta\mathbf{p} + o(||\Delta\mathbf{p}||) \tag{2}$$

with $\nabla \Sigma_e = (\partial \Sigma_e / \partial x, \partial \Sigma_e / \partial y)^T$.

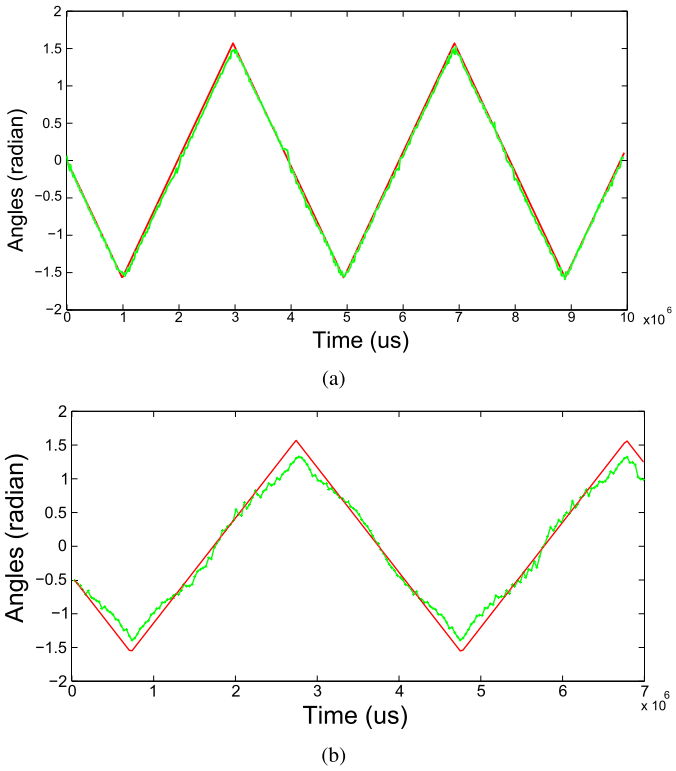



Fig. 4. (a) Ground truth and computed orientation for the stimulus in Fig. 3. Red line: the real orientation of the bar. Green line: the estimated one for the event-based method. (b) Angles estimated using frame-based Horn and Schunck [27]. Red line: real angles. Green line: estimated angles.

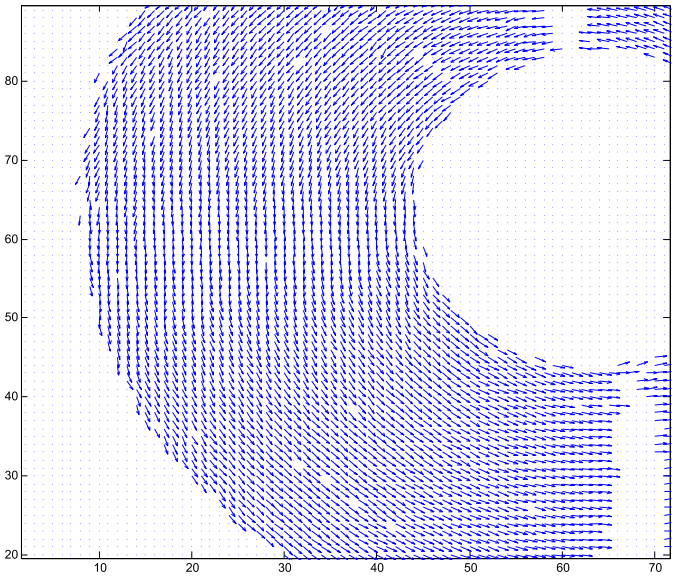


Fig. 5. Sample of computed event-based flow for the stimulus in Fig. 3 shown for a cumulated period of 5 ms for a portion of the focal plane for better readability.

The partial functions of $\Sigma_e$ are functions of a single variable whether $x$ or $y$. For time being a strictly increasing function, $\Sigma_e$ is a nonzero derivatives surface at any point. It is then possible to use the inverse function theorem to write around

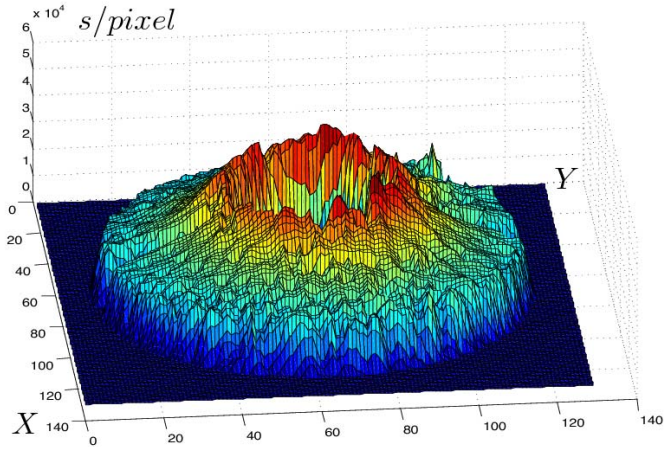
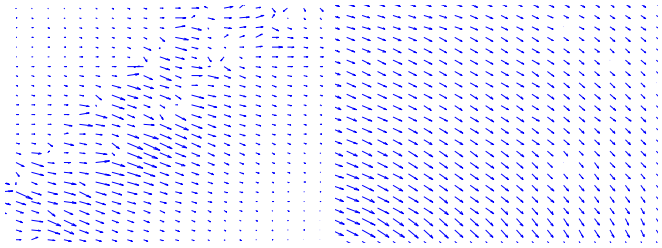



Fig. 6. Amplitude of the computed motion flow for the stimulus in Fig. 3.



(a) (b)

Fig. 7. (a) Frame-based optical flow using Horn–Schunck on a 3 ms accumulation frame. (b) Accumulation of event-based optical flow over a same duration of 3 ms.

a location $\mathbf{p} = (x, y)^T$

$$\frac{\partial \Sigma_e}{\partial x}(x, y_0) = \frac{\mathrm{d}\,\Sigma_e|_{y_0}}{\mathrm{d}x}(x) = \frac{1}{v_x(x, y_0)}$$
$$\frac{\partial \Sigma_e}{\partial y}(x_0, y) = \frac{\mathrm{d}\,\Sigma_e|_{x_0}}{\mathrm{d}y}(y) = \frac{1}{v_y(x_0, y)} \tag{3}$$

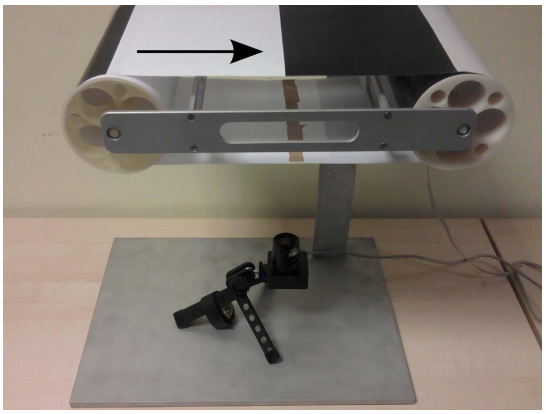
$\Sigma_e|_{x_0}$, $\Sigma_e|_{y_0}$ being $\Sigma_e$ restricted respectively to $y = y_0$ and $x = x_0$. The gradient $\nabla \Sigma_e$ can then be written as

$$\nabla \Sigma_e = \left(\frac{1}{v_x}, \frac{1}{v_y}\right)^T. \tag{4}$$

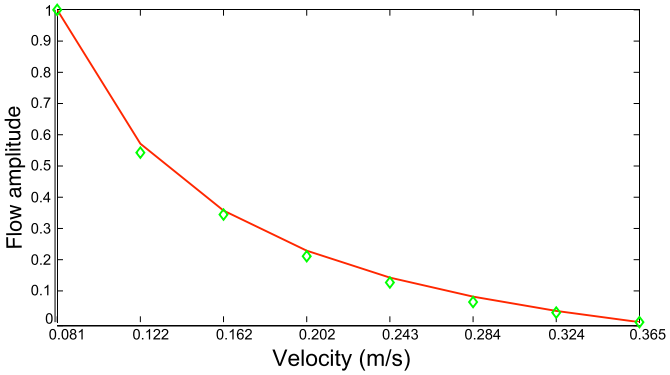
The vector $\nabla \Sigma_e$ measures the rate and the direction of change of time with respect to the space, and its components are also the inverse of the components of the velocity vector estimated at $\mathbf{p}$.

### *B. Flow Regularization*

The flow definition given by 4 is sensitive to noise because it consists of estimating the partial derivatives of $\Sigma_e$ at each individual event. One way of making the flow estimation robust against noise is to add a regularization process to the estimation. To achieve this, we assume a local velocity constancy. This hypothesis is satisfied in practice for small sets of spatiotemporaly clustered events. It is then equivalent to assume $\Sigma_e$ being locally planar because its partial spatial derivatives are the inverse of the speed, hence constant velocities produce constant spatial rate of change in $\Sigma_e$.



(a)




(b)

Fig. 8. (a) Belt driven by a dc motor, controlled in closed loop. (b) Estimated amplitude (green diamond) is shown with the real one (red line).

Finally, the slope of the fitted plane with respect to time axis is directly proportional to the motion velocity. The regularization also compensates for absent events in the neighborhood of active events where motion is being computed. The plane fitting provides an approximation of the timing of nonactive spatial locations because of the nonidealities and the asynchronous nature of the sensor.

A robust plane fitting is applied to each event arriving at time $t$ over a spatiotemporal window of dimensions $L \times L \times 2\Delta t$, centered on the event. In practice, a spatiotemporal window is selected empirically for the fitting, in what follows $L = 3$ and $\Delta t \sim 1$ ms.

Any event $e(\mathbf{p}, t)$ belongs to a plane of parameters $\Pi = \begin{pmatrix} a & b & c & d \end{pmatrix}^T$ if the following equality is satisfied:

$$\Pi^T \begin{pmatrix} \mathbf{p} \\ t \\ 1 \end{pmatrix} = 0. \tag{5}$$

According to this equality, the regularization operation can be performed as detailed in Algorithm 1 that provides the whole approach of computing motion flow.

The threshold in step 4 can be set to $th_1 =$ 1e-5, it is usually the magnitude of accuracy we get from this iterative estimation algorithm. The second threshold in step 5 is also set to $th_2 = 0.05$, according to the experimental results. Moreover, the error usually converges in just 1–2 iterations. The resulting algorithm is robust and consumes less computational resources.

**Algorithm 1** Local planes fitting algorithm on incoming events.

1: **for all** event $e(\mathbf{p}, t)$ **do**
2: Define a spatiotemporal neighborhood $\Omega_e$, centered on $e$ of spatial dimensions $L \times L$ and duration $[t - \Delta t, t + \Delta t]$.
3: Initialization:
- apply a least square minimization to estimate the plane $\Pi = \begin{pmatrix} a & b & c & d \end{pmatrix}^T$ fitting all events $\tilde{e}_i(\mathbf{p}_i, t_i) \in \Omega_e$:

$$\tilde{\Pi}_0 = \underset{\Pi \in \mathbb{R}^4}{\operatorname{argmin}} \sum_i \left| \Pi^T \begin{pmatrix} \mathbf{p}_i \\ t_i \\ 1 \end{pmatrix} \right|^2 \tag{6}$$

- set $\epsilon$ to some arbitrarly high value ($\sim 10e6$).

4: **while** $\epsilon > th_1$ **do**
5: Reject the $\tilde{e}_i \in \Omega_e$ if $|\tilde{\Pi}_0^T \begin{pmatrix} \mathbf{p}_i \\ t_i \\ 1 \end{pmatrix}| > th_2$ (i.e. the event is too far from the plane) and apply Eq. 6 to estimate $\tilde{\Pi}$ with the non rejected $\tilde{e}_i$ in $\Omega_e$.
6: Set $\epsilon = ||\tilde{\Pi} - \tilde{\Pi}_0||$, then set $\tilde{\Pi}_0 = \tilde{\Pi}$.
7: **end while**
8: Attribute to $e$ the velocity defined by the fitted plane.
9: **end for**
10: **return** $v_x(e)$, $v_y(e)$.

## IV. Results and Validation

The input used for the validation of the proposed method is provided by a DVS camera connected to a PC via a USB port. The motion flow algorithm has been implemented in Java using jAER [32] and in C using a linux driver of the DVS. The results obtained with the event-based method are compared with frame-based algorithms such as Horn and Shunck [27], and Lucas and Kanade [33]. For Horn and Schunck, following [19], the smoothness term $\lambda$ is set to 0.5, with a number of iteration $<100$. The event-based flow computation is performed according to Algorithm 1. The flow is regularized as detailed in Section III-B, but $\Sigma_e$ is never updated by the regularization operation. In the experiments, events are also clustered according to their polarities. The ON and the OFF events are then processed separately and the final results shown are merging both the pathways.

### A. Orientation

In the first experiment, the motion flow is computed for a black bar painted on a white disk, rotating with a constant angular velocity $\omega = 1.59$ rad $\cdot$ s$^{-1}$ as shown in Fig. 3. The event-based optical flow is computed in real time directly using the output train of events. The frame-based optical flow is computed for a 30 fps using a conventional frame-based camera where images are rescaled to $128 \times 128$ pixels to allow for comparison with the DVS.

Fig. 4 shows the orientation of the bar computed from the flow plotted versus the ground truth orientation of the bar for the event-based and Horn and Schunck algorithms. The mean error corresponding to the event-based algorithm is

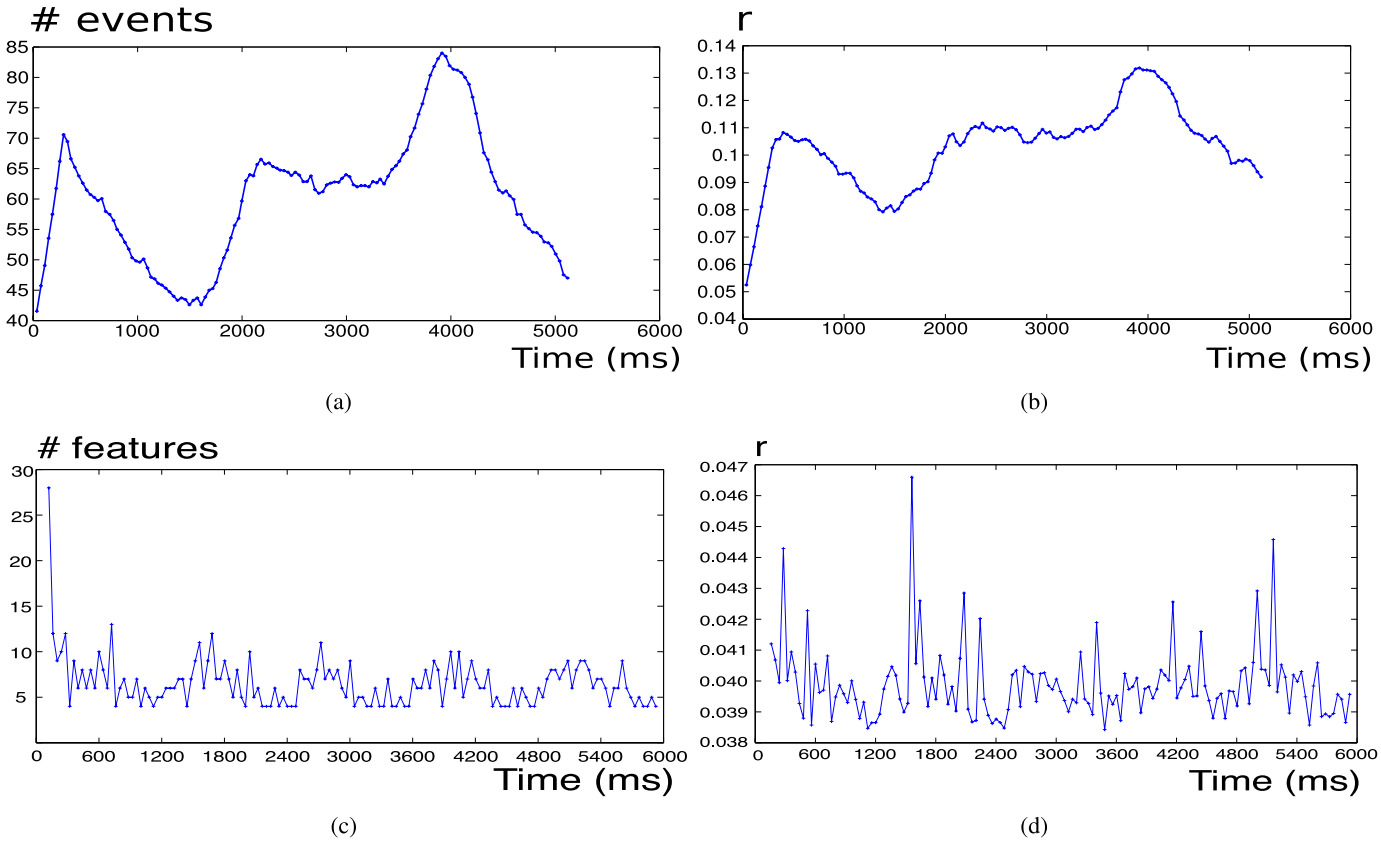



Fig. 9. (a) Number of events per time bins generated by the rotating bar shown in Fig. 3. (b) Processing time ratio of the event-based technique: the mean processing ratio is equal to 0.1, for a mean number of events equal to 60. (c) Number of features using cumulated frames generated by the optimized Lucas–Kanade's algorithm. The algorithm selects the features on which the optical flow is computed. (d) Processing time ratio of the Lucas–Kanade's technique: the mean processing ratio is equal to 0.0395 and the mean number of features is 6.

0.037 rad, with a standard deviation of 0.025 rad. The mean error corresponding to the Horn and Schunck [27] frame-based implementation is 0.11282 rad, with a standard deviation of 0.07816 rad. This is three times higher than the event-based method. The event-based computation takes full advantage of the high temporal accuracy of the sensor, by providing a smoother and more accurate estimation of the rotation angles in real time. A sample of the computed flow is shown in Fig. 5, highlighting the quality of the estimated motion, the motion flow is shown for a period of 5 ms. The lower accuracy of the flow estimated by the Horn and Shunk algorithm on reconstructed images is largely because of the lower temporal resolution of frames. Increasing the frame-rate would of course improve the frame-based performance, but at the cost of an increase of the needed computational resources and, as shown later, at the cost of real-time performances.

### *B. Amplitude*

The cumulative amplitude of the visual motion flow for the previous experiment is shown in Fig. 6. Each spatial location is associated with its estimated flow amplitude after a single rotation. As the computed component of the flow are inverted, high amplitudes correspond to slow motions, and vice versa. As expected, the velocity in pixel of the rotating bar increases with radius. The expected theoretical ratio of the velocity between the outer and the inner rim is 3.09, the estimated ratio after computation is 3.07.

A sample of flows computed from the same sequence on the same time period of 3 ms for frame-based and event-based methods is shown in Fig. 7. Frames have been generated by cumulating events to simulate a frame rate of 3 ms to ensure clarity in the display of results and allow comparison on a short timescale. In the absence of a fast camera, and because of the simplicity of the stimulus, cumulating events has shown to efficiently approximate the frames [10]. In Fig. 7, frame-based computation produces a wide variety of amplitude responses because of the static nature of frame-based acquisition. It is usually a nonsmooth vector field because of the nondiscontinuities introduced by frames' sampling. The method used to improve the smoothness of the vector field as appearing in [27] applies a general smoothing paradigm that induces additional artifacts. Computed event-based amplitudes in Fig. 7 are smooth and closely correspond to the expected amplitudes because of the use of the high temporal resolution of the DVS.

In second experiment, the amplitudes of the estimated motion flow are computed for a moving pattern of bars presented on a moving conveyor belt, whose translational speed can be accurately set by adjusting the supply voltage of a dc motor [Fig. 8(a)].

The amplitudes of the visual flow are computed for different known belt speeds obtained by setting the dc motor with a range of velocities 0.081 $\mathrm{m}\cdot\mathrm{s}^{-1}$ to 0.365 $\mathrm{m}\cdot\mathrm{s}^{-1}$. To compare the real and the computed amplitudes, we normalize the

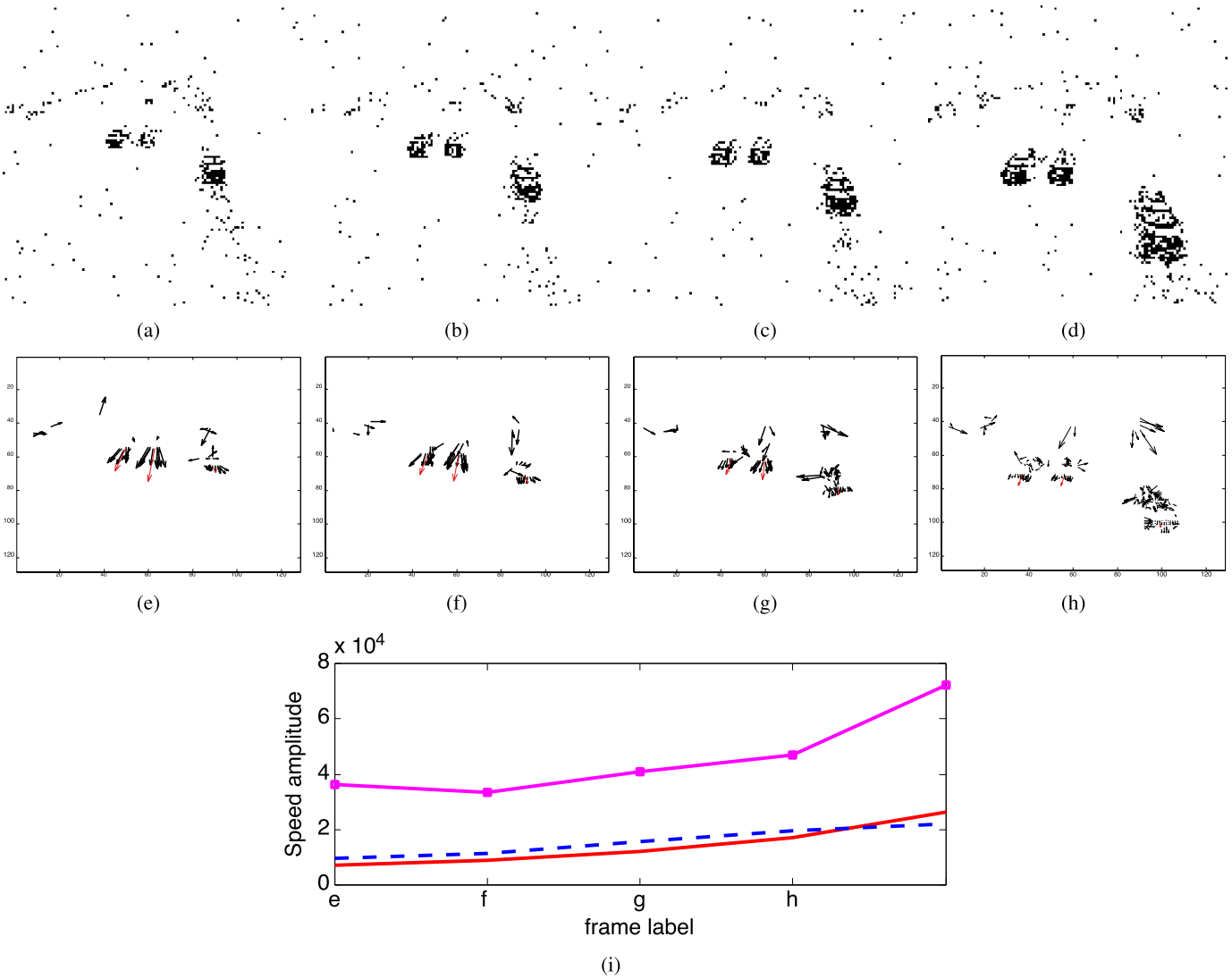



Fig. 10. (a–d) Outdoor scene acquired by the event-based retina showing cars on a highway using cumulated events on the focal plane. (e–h) Mean motion flow of each vehicle (arrows) increases as the cars get closer to the retina. This is a direct effect of the perspective projection. The mean velocities amplitudes are shown in (i) for each of the three cars.

estimated amplitudes and the velocities of the moving patterns with their respective maximum values. The normalized results, shown in Fig. 8(b), correspond to the estimated velocity of the motion flow, and the ground truth ranges in the interval $[0.081, 0.365]$ m $\cdot$ s$^{-1}$. The ground truth and the estimated values coincide with a mean error of 9%.

### C. Computational Time

We define $\Delta t_b$ as the time interval within which the visual signal is observed, and it is equal to the inverse of the frame-rate of the standard perspective camera (i.e., $\Delta t_b \sim 33$ ms if the camera captures at 30 fps). For a meaningful comparison, this time duration is chosen to evaluate the frame-based and the event-based techniques. Let $\Delta t_c$ be the processing time consumed for the flow estimation from the signal acquired during $\Delta t_b$. We then define the ratio $r$ such as: $r = \Delta t_c / \Delta t_b$. If $r \leq 1$ then the computation can be performed in real time. For the experiments, the computation of the frame-based optical-flow uses the OPEN-CV implementation Lucas and Kanade's [33] algorithm. It relies on a preprocessing of incoming images to select features [34] on which the motion visual motion flow is then estimated. Fig. 9(a) and Fig. 9(c) shows the number of events and features used to compute the flow for the rotating bar. As shown in Fig. 9(c), the maximum ratio in the case of event-based algorithm is $r = 0.1$, showing that the visual flow is estimated in real time. The frame-based implementation processing time ratio is $r = 0.0395$ as shown in Fig. 9(d). It also allows real-time estimation, assuming only a few image features are detected.

The mean number of events and the mean processing ratio allow to estimate the mean processing time of a single event: $33 \times 0.1/60 \sim 0.055$ ms (mean number of events per millisecond $= 60$ and mean processing ratio r $= 0.1$). With the same consideration, the mean time to process one image feature with the frame-based optimized implementation is equal to $33 \times 0.0395/6 \sim 0.217$ ms (mean number of feature per millisecond $= 6$ and mean processing ratio per feature r $= 0.035$). This means that within the time slot of 33 ms, around 600 events can be processed in real time with the first

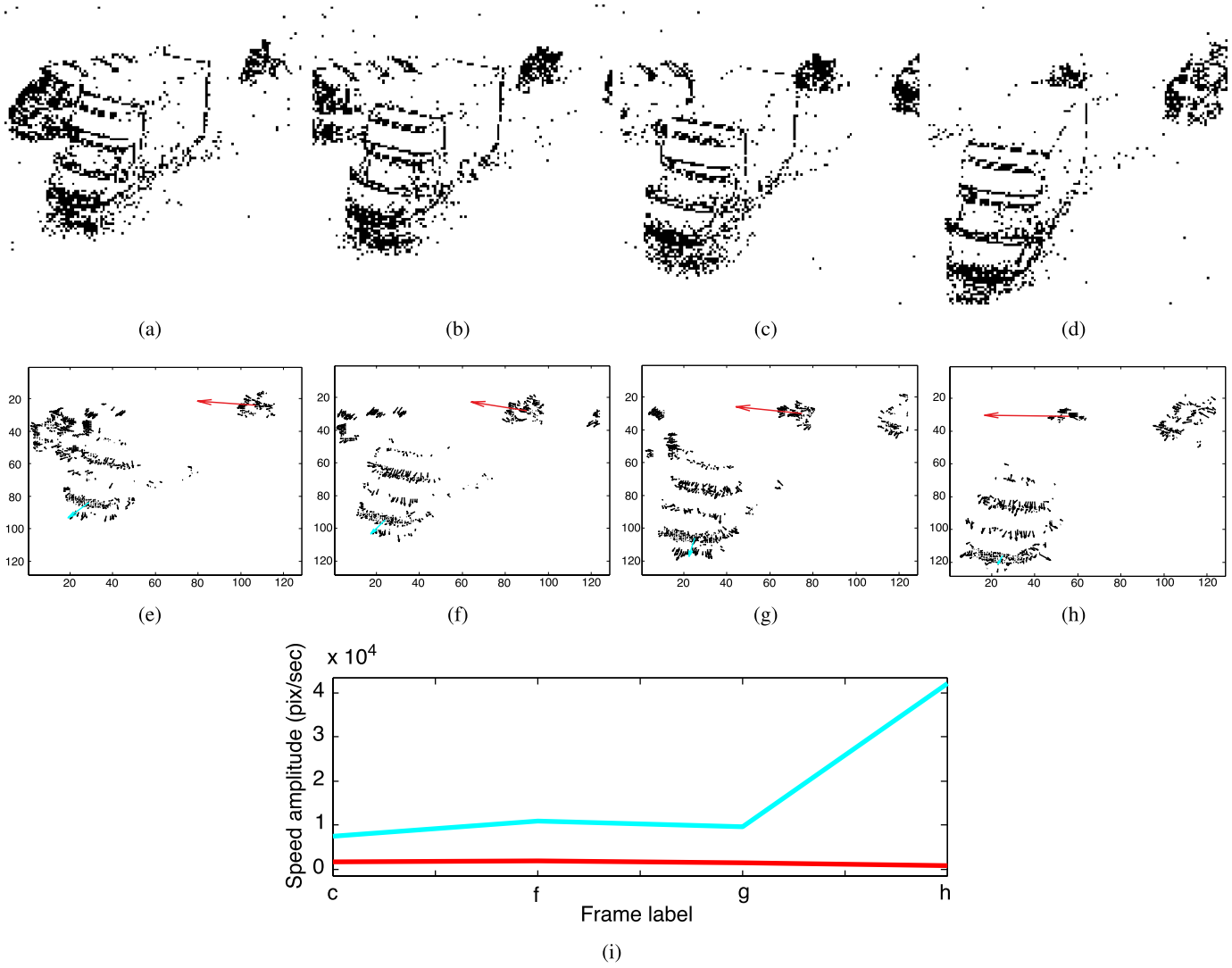



Fig. 11. (a–d) Cumulative events on the focal plane of an outdoor traffic scene. In (c–h) the car moving from left to right (in the background) has an almost constant depth. The truck moving toward the retina is showing an increasing velocity amplitude because of the perspective projection. In (i) the mean velocities amplitude of each vehicle is shown: bottom curve is reflecting the almost constant velocities of the car, while the increasing one provides the velocity of the truck moving toward the retina.

technique, while only 150 features can be processed in real time in the second technique. The Lucas–Kanade C++ OpenCV implementation is highly optimized if compared to the Java implementation of the event-based algorithm. Presented results can then be seen as a lower bound of the event-based algorithm performances. Event-based motion flow is expected to run even faster with an adequate algorithm optimization, because its computational principle is simple.

### *D. Remarks*

We followed in this paper the same set of experiments as in [10] using the same recordings. Comparing both the methods, we found that they perform similarly. The only difference lies in the computational performances, and the presented work lowers the amount of computational load by an order of 15%. Perhaps the presented method is more easy to tune that the work presented in [10] was sensitive to the summation period of events to produce a local contrast patch. This parameters needs to be tuned sharply according to the scene's content. The presented method does not necessitate such a fine tuning, a coarse value for the plane fitting is sufficient as long as a sufficient number of events is collected (2–5 ms for all type of scenes). While in [10], beyond a certain summation time interval because of the refractory period of the pixels, the summation of events tend to produce wrong contrast patches, thus providing imprecise orientations.

### *E. Natural Scenes*

In the case of natural scenes, the flow estimation is harder to evaluate as the ground truth is not available. However, it is still possible to show the coherence of the computed flows with the scene's content. In the first set of data shown in Fig. 10(a–d), the velocity vectors amplitudes of cars moving along highway lanes using the event-based retina are shown. Velocities are increasing as the cars are getting closer to the retina. This is a direct effect of the perspective projection. A second sequence acquired by the retina shown in Fig. 11 confirms the same observations.

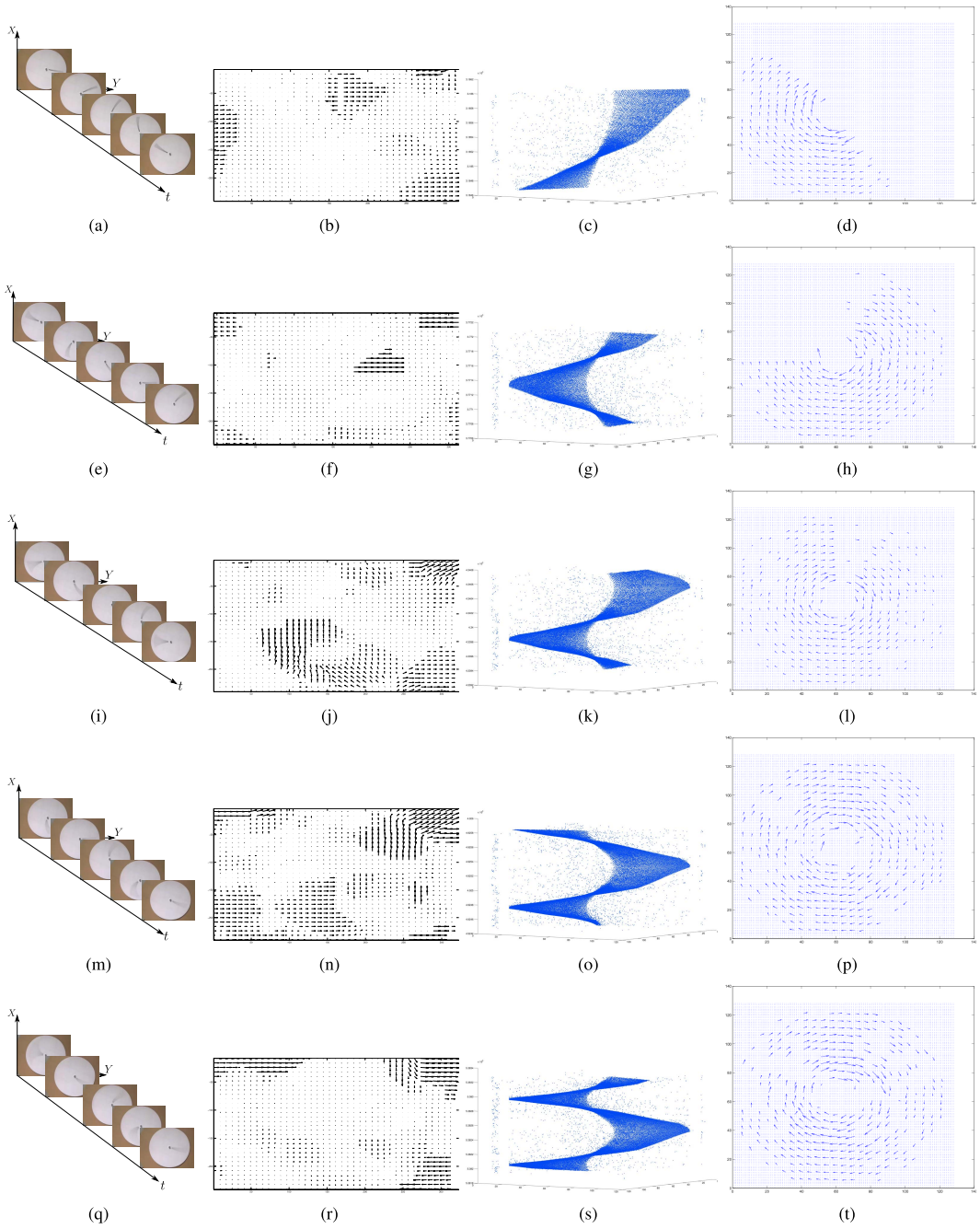



Fig. 12. Comparison of the flows computed with a 30 fps frame-based camera and the DVS. The disk is rotated at a speed ranging from 450 to 5000 rpm with an elementary step of 500 rpm. The first column [(a), (e), (i), (m), (q)] shows the samples of images acquired by a conventional camera at 30 fps. The second column [(b), (f), (j), (n), (r)] shows the results using the Horn and Schunck's algorithm. As expected, frames are not suitable for these high-speed motion as the motion blur is preventing flow estimation, motion results are chaotic and irregular and not fitting the general rotational motion of the bar. The third column [(c), (g), (k), (o), (s)] shows the events generated by the rotating bar in the spatiotemporal space of events. One can notice that as the bar rotates faster, more events are collected and longer portions of the motion are acquired. The last column [(d), (h), (l), (p), (t)] provides the flow estimation using the event-based algorithm; the flows are accurately estimated in real time for all rotational speeds.

### *F. Limitations*

The retina is subject to several nonidealities that are causing latencies in the signal acquisition. These factors are limiting the accuracy of the computed flow. In this experiment, we consider a rotating moving bar, as shown in Fig. 12. The bar, is observed by both the spiking retina and a conventional camera. It has an angular speed increasing progressively from 450 to 5000 rpm.

The first column [(a), (e), (i), (m), (q)] shows the samples of images acquired by a conventional camera at 30 fps. The second column [(b), (f), (j), (n), (r)] shows the results using the Horn and Schunck's algorithm. As expected, frames are not suitable for these high speed motion as the motion blur is preventing flow estimation, motion results are chaotic and irregular and not fitting the general rotational motion of the bar. The motion flow is already inaccurate at the lowest speed as shown in Fig. 12(b).

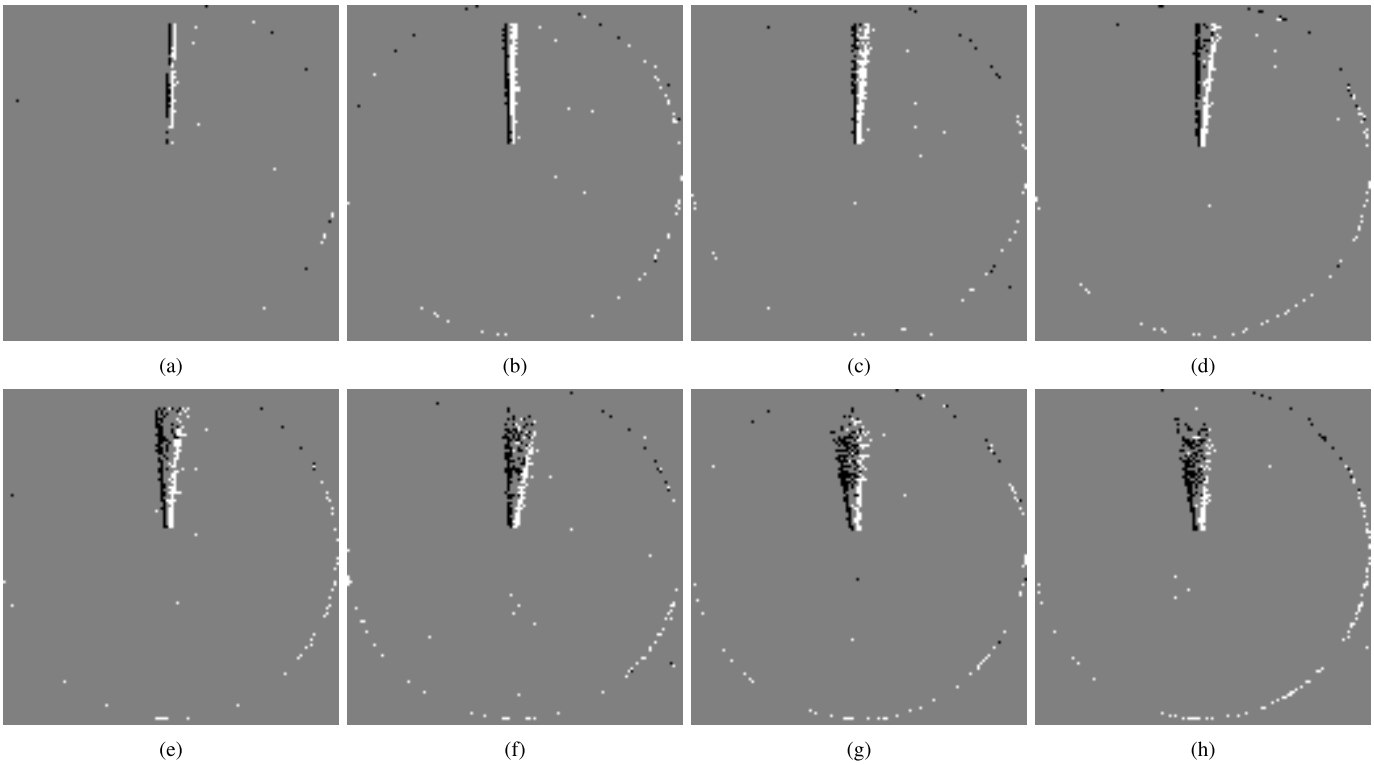



Fig. 13. Sequences of a disk rotating with an increasing speed from 450 rpm to 5000 rpm for the event-based retina. Cumulated events in the focal planes are shown for a fixed time period of 500 ms. Events underlining the edges are more and more scattered as the rotation speed increases; this is a typical effect of the motion blur.

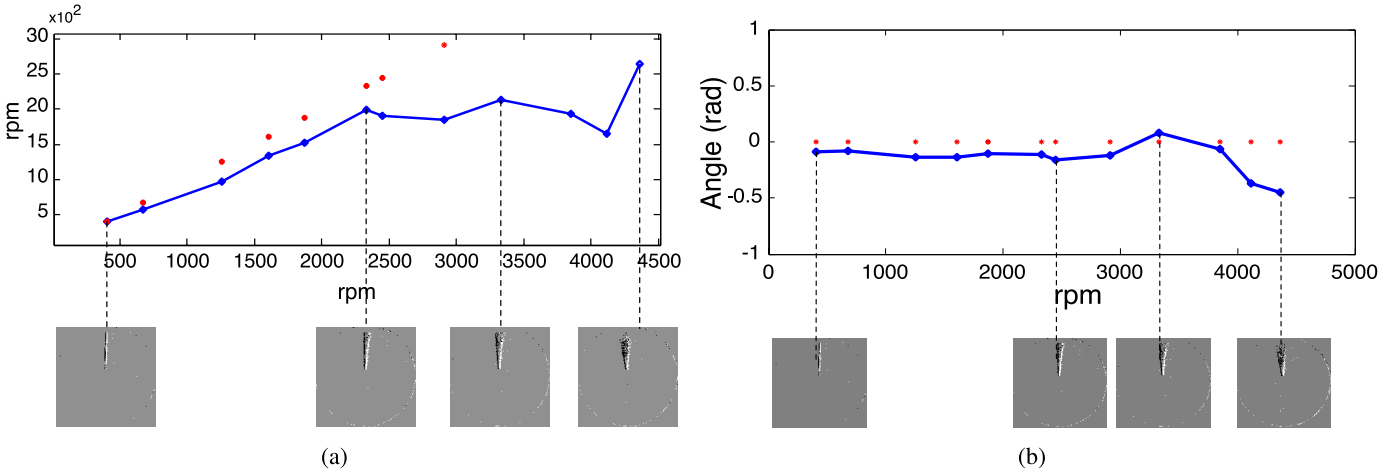



Fig. 14. In (a) the mean velocity of the estimated flow and in (b) the mean angle estimation are shown using the event-based retina from the event flow of the rotating bar. The angular speed increases from 450 to 5000 rpm. Estimations are given each time the bar reaches its initial starting position. The red dots provide the ground truth. The estimation of the motion parameters is stable up to 2500 rpm. Beyond the orientation seems to stay stable up to 400 rpm. Less events are collected at high speed, thus affecting the slope of fitted plane but not its direction. This is the limit of the sensor for the used lighting conditions.

The third column [(c), (g), (k), (o), (s)] shows the events generated by the rotating bar in the spatiotemporal space of events. One can notice that as the bar rotates faster, more events are collected and longer portions of the motion are acquired. The last column [(d), (h), (l), (p), (t)] provides the flow estimation using the event-based algorithm. The flows are accurately estimated in real-time for all rotational speeds. It is possible to notice that as the rotational speed increases the motion flow in the event-based case tends to be sparser, Fig. 13 shows the motion blurs, resulting from the retina's latencies when capturing light. This is the main limiting element of the retina in estimating accurately the flow. With the increasing speed of the rotation, motion blurs induces clusters of events instead of sharp edges. The retina is also not generating a sufficient amount of events form all spatial locations, some pixels are not activated by the moving bar. The fitting of $\Sigma_e$ is then inevitably affected. As shown in Fig. 14, the estimated amplitude and orientation of the bar follow closely the ground-truth up to a rotation velocity of 2500 rpm. Beyond this velocity, the orientation seems to stay stable up to 4000 rpm.

Less events are collected at high speed thus affecting the slope of fitted plane but not its direction. This is the limit of the sensor. The use of better lighting conditions will lower these effects.

## V. Conclusion

This paper shows that the space of coactive events allows to directly derive information about the direction of a visual stimulus. The precise timing conveyed by the neuromorphic asynchronous event-based vision sensor is fully used to determine locally for each incoming event its direction and amount of motion. Timing is the essential computational element, the whole computation is based on its precision. The method complies with the concepts of event-based computation, processing is performed on each incoming event instead of a time interval. The presented work differs from existing frame-based techniques that consider temporal window frames (33–500 ms) and induce unnatural high computational costs. The developed method can be applied equally on other modalities such as tactile that shares common characteristics with vision, both rely on a spatial grid of sensors (pressure: mechanoreceptor, light: photoreceptor) that input to a chain of processing precise neural responses.

The presented approach did not impose any mathematical model, results show that the natural properties of spatiotemporal spaces provide the inverse of the velocity. This observation sets the estimation of motion on time instead of space. The precise timing properties of the method comply with the dynamic properties of natural environments and the importance of time measurement in the brain, Actions are driven by time, motor control depends on processes that have to determine when exactly to perform action. This then adds to the ongoing debate around whether perception and movement share common time lines or are organized on separate clocks and coding. Event-based acquisition provides a fast common mechanism that allows perception to be linked to motion by a common computationally inexpensive timing system. This could then provide an efficient way of controlling sensory-dependent behavior and anticipating changes in the environment.

## Acknowledgment

The authors would like to thank both the CapoCaccia Cognitive Neuromorphic Engineering Workshop and the NSF Telluride Neuromorphic Cognition workshops.

## References

[1] D. H. Perkel and T. H. Bullock, "Neural coding," *Neurosci. Res. Program Bull.*, vol. 6, no. 3, pp. 221–348, 1968.

[2] H. B. Barlow and W. R. Levick, "Changes in the maintained discharge with adaptation level in the cat retina," *J. Physiol.*, vol. 200, no. 3, pp. 1–24, 1969.

[3] W. Bialek, F. Rieke, R. de Ruyter van Steveninck, and D. Warland, "Reading a neural code," *Science*, vol. 252, pp. 1854–1857, Jun. 1991.

[4] J. J. Atick, "Could information theory provide an ecological theory of sensory processing," *Network*, vol. 3, no. 2, pp. 213–251, 1992.

[5] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120dB 15$\mu$s latency asynchronous temporal contrast vision sensor," *IEEE J. Solid State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.

[6] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011.

[7] X. Guo, X. Qi, and J. Harris, "A time-to-first-spike cmos image sensor," *IEEE Sensors J.*, vol. 7, no. 8, pp. 1165–1175, Aug. 2007.

[8] J. Lenero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, "A 3.6 $\mu$s latency asynchronous frame-free event-driven dynamic-vision-sensor," *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1443–1455, Jun. 2011.

[9] J. Perez-Carrasco, C. Serrano, B. Acha, T. Serrano-Gotarredona, and B. Linares-Barranco, "Event based vision sensing and processing," in *Proc. 15th IEEE ICIP*, Oct. 2008, pp. 1392–1395.

[10] R. Benosman, S. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan, "Asynchronous frameless event-based optical flow," *Neural Netw.*, vol. 27, pp. 32–37, Mar. 2012.

[11] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S.-C. Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. Ballcels, T. Serrano-Gotarredona, A. Acosta-Jimenez, and B. Linares-Barranco, "CAVIAR: A 45k neuron, 5M synapse, 12G connects/s AER hardware sensory system for high-speed visual object recognition and tracking," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1417–1438, Sep. 2009.

[12] Z. Fu, T. Delbruck, 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.

[13] R. Benosman, S. Ieng, C. Posch, and P. Rogister, "Asynchronous event-based Hebbian epipolar geometry," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1723–1734, Nov. 2011.

[14] P. Rogister, R. Benosman, S. Ieng, P. Lichtsteiner, and T. Delbruck, "Asynchronous event-based binocular stereo matching," *IEEE Trans. Neural Netw. Learn Syst.*, vol. 23, no. 2, pp. 347–353, Feb. 2012.

[15] J. A. Pérez-Carrasco, B. Acha, C. Serrano, L. A. Camuñas-Mesa, T. Serrano-Gotarredona, and B. Linares-Barranco, "Fast vision through frameless event-based sensing and convolutional processing: Application to texture recognition," *IEEE Trans. Neural Netw.*, vol. 21, no. 4, pp. 609–620, Apr. 2010.

[16] A. B. Watson and A. J. Ahumada, *A Look at Motion in the Frequency Domain*. Washington, DC, USA: Nat. Aeronautics Space Admin., 1983, pp. 1–10.

[17] D. J. Heeger, "Optical flow using spatiotemporal filters," *Int. J. Comput. Vis.*, vol. 1, pp. 279–302, Feb. 1998.

[18] D. J. Heeger, "Model for the extraction of image flow," *J. Opt. Soc. Amer.*, vol. 4, no. 8, pp. 1455–1471, 1987.

[19] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 43–77, 1994.

[20] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *Int. J. Comput. Vis.*, vol. 2, no. 3, pp. 283–310, 1989.

[21] R. Kories and G. Zimmerman, "A versatile method for the estimation of displacement vector fields from image sequences," in *Proc. IEEE Workshop Motion, Represent. Anal.*, Feb. 1986, pp. 101–106.

[22] A. Singh, *Optic Flow Computation: A Unified Perspective*. New York, NY, USA: Wiley, 1992.

[23] M. A. Sutton, W. J. Walters, W. H. Peters, W. F. Ranson, and S. R. McNeil, "Determination of displacement using an improved digital correlation method," *Image Vis. Comput.*, vol. 1, no. 3, pp. 133–139, 1983.

[24] T. Camus, "Real-time quantized optical flow," *Real-Time Imag.*, vol. 3, no. 2, pp. 71–86, 1997.

[25] J. Banks and P. Corke, "Quantitative evaluation of matching methods and validity measures for stereo vision," *Int. J. Robot. Res.*, vol. 20, no. 7, pp. 512–32, 2001.

[26] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills, "Recovering motion fields: An evaluation of eight optical flow algorithms," in *Proc. 9th Brit. Mach. Vis. Conf.*, 1998, pp. 195–204.

[27] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 13, nos. 1–3, pp. 185–203, 1981.

[28] H. H. Nagel, "On the estimation of optical flow: Relations between different approaches and some new results," *Artif. Intell.*, vol. 33, no. 3, pp. 299–324, 1987.

[29] M. Bjorkman, "Real time motion and stereo cues for active visual observers," Ph.D. dissertation, Dept. Numer. Anal. Comput. Sci., R. Inst. Technol., Stockholm, Sweden, 2002.

[30] M. Mahowald, "Vlsi analogs of neuronal visual processing: A synthesis of form and function," Ph.D. dissertation, Dept. Comput. Neural Syst., California Inst. Technol., Pasadena, CA, USA, 1992.
[31] T. Delbruck, B. Linares-Barranco, E. Culurciello, and C. Posch, "Activity-driven, event-based vision sensors," in *Proc. ISCAS*, 2010, pp. 2426–2429.
[32] T. Delbruck, (2013). *jAER Open Source Project* [Online]. Available: http://sourceforge.net/p/jaer/wiki/Home/
[33] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (IJCAI)," in *Proc. 7th IJCAI*, Apr. 1981, pp. 674–679.
[34] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE CVPR*, Jun. 1994, pp. 593–600.

Authors' biographies and photographs not available at the time of publication.