Neural Networks 66 (2015) 91–106

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Asynchronous event-based corner detection and matching

Xavier Clady*, Sio-Hoi Ieng, Ryad Benosman

Vision Institute, Pierre and Marie Curie University, Paris, France

ARTICLE INFO

Article history:
Received 16 April 2014
Received in revised form 11 February 2015
Accepted 12 February 2015
Available online 11 March 2015

Keywords:
Neuromorphic sensor
Event-based vision
Corner detection and matching
Intersection of constraints

ABSTRACT

This paper introduces an event-based luminance-free method to detect and match corner events from the output of asynchronous event-based neuromorphic retinas. The method relies on the use of space–time properties of moving edges. Asynchronous event-based neuromorphic retinas are composed of autonomous pixels, each of them asynchronously generating "spiking" events that encode relative changes in pixels' illumination at high temporal resolutions. Corner events are defined as the spatiotemporal locations where the aperture problem can be solved using the intersection of several geometric constraints in events' spatiotemporal spaces. A regularization process provides the required constraints, i.e. the motion attributes of the edges with respect to their spatiotemporal locations using local geometric properties of visual events. Experimental results are presented on several real scenes showing the stability and robustness of the detection and matching.

1. Introduction

This article addresses the problem of corner detection from the output of an asynchronous neuromorphic retina formalized in an event-based framework. In the conventional frame-based machine vision paradigm, corner detectors are widely used to localize pixels of interest in images around which features can be extracted. The detection of pixels of interest in images is a fundamental low-level task for several higher level processing algorithms such as: image retrieval (Deselaers, Keysers, & Ney, 2008; Dickscheid, Schindler, & Förstner, 2011), action or object recognition (Mikolajczyk, Leibe, & Schiele, 2005; Weinland, Ronfard, & Boyer, 2011), texture classification (Lategahn, Gross, Steh, & Aach, 2010), robotic navigation and localization (Gil, Mozos, Ballesta, & Reinoso, 2010). Several detection algorithms have been developed (Li & Allinson, 2008; Tuytelaars & Mikolajczyk, 2007) and compared (Dickscheid et al., 2011; Gauglitz, Höllerer, & Turk, 2011; Gil et al., 2010; Mikolajczyk & Schmid, 2005; Mokhtarian & Mohanna, 2006; Moreels & Perona, 2007). The general principle of an interest point detector consists of sorting pixels according to some criterion defined over each pixel's neighborhood. Bottom-up approaches are mainly applying the following scheme: local features derived from edges, gradient, color, etc., are first extracted to build a measure which is then used to characterize pixels. In techniques such as corner detectors (Harris & Stephens, 1988; Mokhtarian & Suomela, 1998; Park, Ahmad, Seung-Hak, Han, & Park, 2004), the core operation is to determine points for which a measurement is locally optimal with respect to some specific criterion. This measure can be computed by a cumulative process (Park et al., 2004), using a self-similarity measure (Moravec, 1980) derived from mathematical analysis (e.g. contour's local curvature Mokhtarian & Suomela, 1998, relying on an eigenvalue decomposition of a second-moment matrix Harris & Stephens, 1988) or selected as the output from a machine learning process (Rosten & Drummond, 2006).

The asynchronous event-based acquisition mechanism provides visual information asynchronously sampled through a high temporal resolution luminance-free approach. Time being a valuable information of these sensors, features in spatiotemporal visual signals must then logically rely on it to derive an accurate detection algorithm. This paper is focused on the essential task of detecting corners in visual streams of events. Unlike detectors applied in conventional image processing which often treats separately detection and motion problems, we will formulate the problem as a unique problem relying on estimation of local velocities.

The rest of this Section presents the event-based formalism used to represent visual information in a neuromorphic vision sensor and the aperture problem. In Section 2, we introduce the event-based corner detection principle. A matching process using the estimated velocity, obtained for each detected corner event, is presented in Section 2.4. Section 3 presents experimental results using moving objects in natural unconstrained outdoor scenes. Finally, Section 4 provides a discussion and a conclusion on the biological plausibility and limitations of the proposed approach.

* Corresponding author.
E-mail address: xavier.clady@upmc.fr (X. Clady).

http://dx.doi.org/10.1016/j.neunet.2015.02.013
0893-6080/© 2015 Elsevier Ltd. All rights reserved.

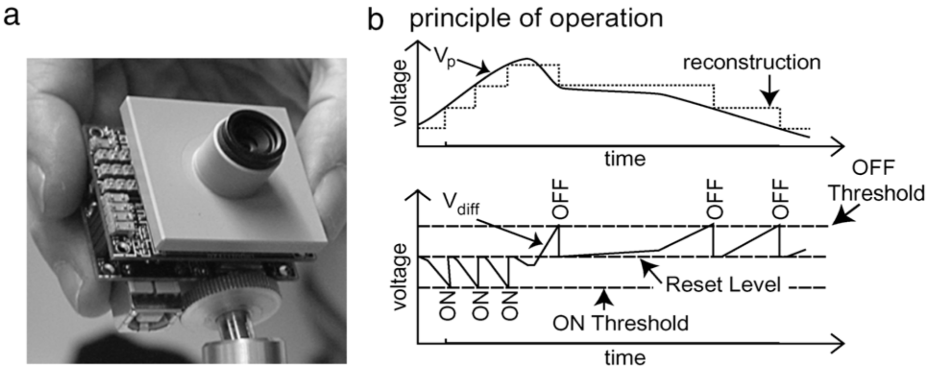


Fig. 1. (a) First generation DVS sensor with 128 by 128 pixels (Lichtsteiner et al., 2008). (b) Principle of ON and OFF spikes generation of DVS pixels, adapted from Lichtsteiner et al. (2008). On top, the evolution of pixel's voltage V_p. Below, the corresponding generation of ON (voltage increases above change threshold) and OFF (voltage decreases) events, from which the evolution of V_p can be reconstructed.

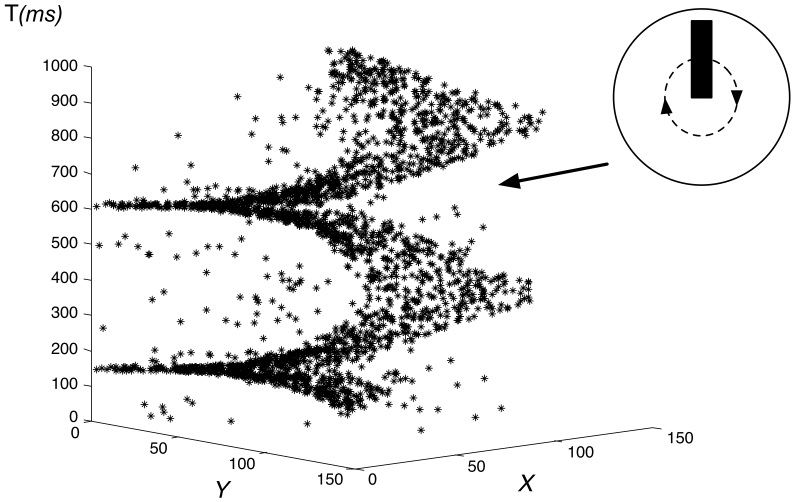


Fig. 2. Space–time representation of events generated in response to a rotating black bar. Each dot represents a visual event.

1.1. Neuromorphic event-based vision

Biological retinas unlike frame-based cameras, transmit extremely low redundant information about a visual scene in an asynchronous manner. The various functionalities of the retina have been reproduced by neuromorphic vision sensors since the late eighties in the pioneering work of Mahowald (1992). Since then, the most interesting achievements of the neuromorphic retinas' field has been the development of activity-driven sensing. The event-based vision sensors output compressed digital data in the form of events which encode light information with a much higher dynamic range than conventional imagers. A complete review of the history and existing sensors can be found in Delbruck, Linares-Barranco, Culurciello, and Posch (2010).

The Dynamic Vision Sensor (DVS) used in this work is an Address-Event Representation (AER) silicon retina with 128 × 128 pixels (Lichtsteiner, Posch, & Delbruck, 2008). It reproduces the transient responses of the retina (Roska & Werblin, 2003). Pixels asynchronously detect temporal changes of contrast as shown in Fig. 1. An ON/OFF event is generated by each pixel each time the contrast exceeds a predefined threshold.

The retina pixels also implement a local gain adaptation mechanism which allows them to work over scene illuminations that range from 2 lux to over 100 klux. Events can be transmitted off-ship, they are timestamped using off-chip digital components and then transmitted to a computer using a standard USB connection. Only moving objects produce data thus reducing the load of post-processing, thus very long streams can be recorded on the host computer at high speed.

The timing of events in the DVS can be conveyed with a very accurate temporal resolution of 1 μs. Unlike the processing relying on frames, the event-based processing of the visual event stream fully exploiting temporal information of the scene. A visual event output by the retina can be defined as a four components vector:

$$\mathbf{e} = (\mathbf{p}, t, pol)^T, \qquad (1)$$

where $\mathbf{p} = (x, y)^T$ is the spatial coordinate of the event, t, its timestamp and $pol \in \{-1, 1\}$ is the polarity. Fig. 2 shows an example of the event-based signal generated by a rotating bar and represented in a spatiotemporal coordinate frame (XYT).

1.2. Corner events from coherent visual pattern

Cameras have a finite aperture size, motion estimation is then possible only for directions orthogonal to edges. This problem is known in the literature as the aperture problem (Wallach, 1935; Wohlgemuth, 1935).

Fig. 3 shows the ambiguity due to the finite aperture. This can be written as follows: if $\mathbf{v}^n$ is the normal component of the velocity vector to an edge at time t at a location $\mathbf{p}$, then the real velocity vector is an element of the $\mathbb{R}^2$ subspace spanned by the unit vector $\mathbf{v}^t$, tangent to the edge at $\mathbf{p}$. This subspace is defined as $\mathcal{V}_1 = \{\mathbf{v} = \mathbf{v}^n + \alpha\mathbf{v}^t\}$ with $\alpha \in \mathbb{R}$. For a regular edge point, α can usually not be estimated.

If we assume another edge intersecting the previous one at the same time t, at the same location $\mathbf{p}$ then its velocity is also

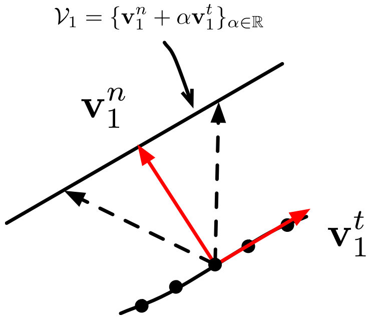

$\mathcal{V}_1 = \{\mathbf{v}_1^n + \alpha \mathbf{v}_1^t\}_{\alpha \in \mathbb{R}}$
$\mathbf{v}_1^n$
$\mathbf{v}_1^t$

Fig. 3. The aperture problem is allowing to estimate only the normal component $\mathbf{v}_1^n$ of the velocity of events generated by an edge. The tangential component $\mathbf{v}_1^t$ is not recoverable. Any motion with the same component $\mathbf{v}_1^n$ is inducing the same stimulus. Those motions define the real plane subspace $\mathcal{V}_1$.

an element of the $\mathbb{R}^2$ subspace $\mathcal{V}_2 = \{\mathbf{v}_2^n + \beta \mathbf{v}_2^t\}$. In that case, it is obvious that the real velocity is the one satisfying both constraints, namely the intersection of the two subspaces: $\mathcal{V}_1 \bigcap \mathcal{V}_2$ (see Fig. 4(a)). When two crossed moving gratings are superimposed to produce a coherent moving pattern, the velocity can be unambiguously estimated (Adelson & Movshon, 1982). In the case of event-based signals, a moving edge generates events at a rate directly related to the real motion velocity. This instantaneous velocity projected onto the focal plane will assign to each event a 2D velocity. This forms the well known optical flow.

For the same reason presented in the previous paragraphs, events generated at edges intersections can have their velocities estimated without any ambiguity and subsequently these events can be collected as corner events.

2. Corner events detection

To estimate the respective velocity subspace spanned by each local edge, a regularization is operated on the event stream in the (XYT) coordinate frame. The regularization consists of a local plane fitting on events generated within a spatiotemporal window. The plane fitting operation is a local linear regularization which extracts the normal component of the velocity vector as it is coplanar to the plane's gradient. An event is labeled as corner if it belongs to the intersection of two or more fitting planes. Fig. 4(b) shows the plane fitting operation allowing to estimate $\mathbf{v}_1^n, \mathbf{v}_2^n$. The tangential components $\mathbf{v}_1^t, \mathbf{v}_2^t$ are obvious since they are directly collinear to the respective edges. Thus, we have all the necessary elements to build the subspaces $\mathcal{V}_1$ and $\mathcal{V}_2$ and estimate the velocity $\mathbf{v}$ as the intersection of the two subspaces.

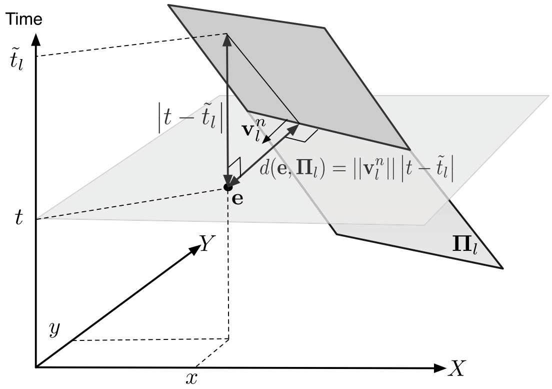


Fig. 5. Computation of the spatial distance $d(e, \mathbf{\Pi}_l)$ between the moving edge, modeled as the plane $\mathbf{\Pi}_l$, and the event $\mathbf{e}$ at time t.

2.1. Robust plane fitting

Let us consider the 2D function Σ_e that maps a time t to each spatial location $\mathbf{p}$:

$$\begin{aligned} \Sigma_{\mathbf{e}} &: \mathbb{R}^2 \to \mathbb{R}^+ \\ \mathbf{p} &\mapsto \Sigma_e(\mathbf{p}) = t. \end{aligned} \tag{2}$$

$\Sigma_{\mathbf{e}}$ is represented by a surface in the coordinate frame (XYT). We also assume the spatiotemporal window in which the signal is observed to be small enough to ensure that $\Sigma_{\mathbf{e}}$ is a single-valued function. If we consider a small spatiotemporal neighborhood around an incoming event $\mathbf{e}$, the selected events define locally a rigid edge. $\Sigma_{\mathbf{e}}$ can then be approximated locally by a plane with the following implicit equation:

$$\begin{aligned} &\Pi : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R} \\ &(\mathbf{p}, t) \mapsto \Pi(\mathbf{p}, t) = \mathbf{\Pi}^T \begin{pmatrix} \mathbf{p} \\ t \\ 1 \end{pmatrix} = 0. \end{aligned} \tag{3}$$

The real vector $\mathbf{\Pi} = (a, b, c, d)^T$ refers to the plane approximating locally $\Sigma_{\mathbf{e}}$. If $\nabla \mathbf{\Pi} = (a, b, c)^T$ is the gradient of Π, then the vector $(a, b)^T$ is collinear to $\mathbf{v}^n$. Solving Eq. (3) for $\mathbf{\Pi}$ produces however a solution defined up to a scale since the plane equation is homogeneous. To get the correct $\mathbf{v}^n$, the right scale is given by t and this is achieved just by normalizing $\mathbf{\Pi}$ by c. The velocity $\mathbf{v}$ can be obtained by intersecting several sets of $\mathcal{V}_i$.

The estimation of the plane is achieved as detailed in Algorithm 1. A robust plane fitting is applied to each event $\mathbf{e}$ and over a spatiotemporal window $\Omega(\mathbf{e})$. It is spatially defined as a disk of radius R (corresponding to a small receptive field) centered on the

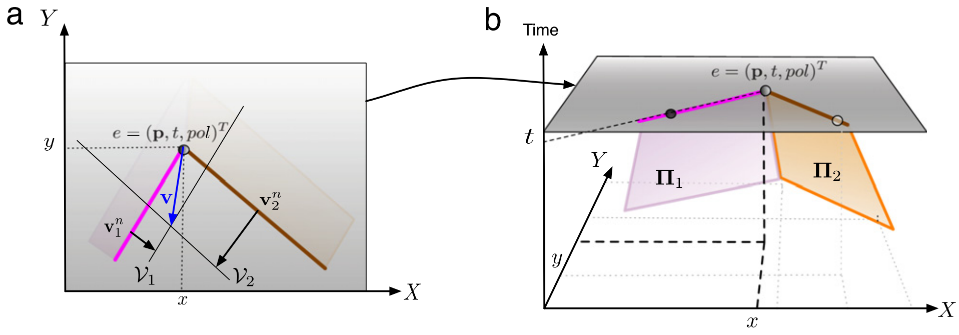


Fig. 4. (a) An event e occurs at spatial location $\mathbf{p}$ at time t where two edges intersect. This configuration provides sufficient constraints to estimate the velocity $\mathbf{v}$ at $\mathbf{p}$ from the normal velocity vector $\mathbf{v}_1^n$ and $\mathbf{v}_2^n$ provided by the two edges. The velocity subspaces $\mathcal{V}_1$ and $\mathcal{V}_2$ are derived from the normal vectors. (b) Vectors $\mathbf{v}_1^n$ and $\mathbf{v}_2^n$ are computed by locally fitting two planes $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ on the events forming each edge over a space–time neighborhood. As shown in Section 2.1, $\mathbf{v}_1^n$ and $\mathbf{v}_2^n$ are extracted from the slope of (respectively) $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ at $(\mathbf{p}, t)$.

spatiotemporal location $(\mathbf{p}, t)$. The temporal interval is adaptively defined according to the N events temporally closest to t. Only the planes fitted with at least 50% of events initially selected in the neighborhood (Step 2) – that are non-rejected (Step 5) – are preserved for the computation of constraints intersection. The resulting algorithm is robust and computationally inexpensive (it usually converges after 1 or 2 iterations). The reader interested in the computation and properties of event-based motion flow can refer to Benosman, Clercq, Lagorce, Ieng, and Bartolozzi (2014) for more details and evaluations. Note that planes are fitted for events having the same polarity.

Algorithm 1 Local plane fitting and normal velocity estimation

```
1: for all event $\mathbf{e} = (\mathbf{p}, t, pol)^T$ do
2:     Define a spatiotemporal neighborhood $\Omega(\mathbf{e})$, centered on $\mathbf{e}$.
3:     Initialization:
         • apply a least square minimization to estimate the plane $\mathbf{\Pi}$ fitting the $N$ events $\mathbf{e}_i = (\mathbf{p}_i, t_i, pol_i)^T \in \Omega(\mathbf{e})$ and with $pol_i = pol$:
           $$\tilde{\mathbf{\Pi}}_0 = \underset{\mathbf{\Pi}\in\mathbb{R}^3}{\operatorname{argmin}} \sum_i \left| \mathbf{\Pi}^T \begin{pmatrix} \mathbf{p}_i \\ t_i \\ 1 \end{pmatrix} \right|^2 \quad (4)$$
         • set $\epsilon$ to some arbitrarily high value ($\sim 10e6$).
4:     while $\epsilon > th_1$ do
5:         Reject the $\mathbf{e}_i$ that are satisfying the inequality $\left| \tilde{\mathbf{\Pi}}_0^T \begin{pmatrix} \mathbf{p}_i \\ t_i \\ 1 \end{pmatrix} \right| > th_2$ (the event is too far from the plane) and apply Eq(4) to estimate a new plane $\tilde{\mathbf{\Pi}}$ with the non rejected $\mathbf{e}_i$ in $\Omega(\mathbf{e})$.
6:         Set $\epsilon = |\tilde{\mathbf{\Pi}} - \tilde{\mathbf{\Pi}}_0|$, then $\tilde{\mathbf{\Pi}}_0 = \tilde{\mathbf{\Pi}}$.
7:     end while
8:     Set $\mathbf{v}^n$ to $\mathbf{e}$.
9: end for
```

2.2. Intersection of constraints

The plane fitting estimates the normal component of the velocity and the related constraint $\mathcal{V}_i$. Corner events are the ones for which the intersection of constraints exists, i.e. the velocity $\mathbf{v}$ at $\mathbf{e}$ satisfies $\mathbf{v} = \bigcap_i \mathcal{V}_i$. Since each constraint $\mathcal{V}_i$ spans a line, $\mathbf{v}$, as their intersection, satisfies the n equations of the following form: $a_i v_x + b_i v_y + c_i = 0$, with $\mathbf{v} = (v_x, v_y)^T$.

An event $\mathbf{e}_k$ is a corner event if it gives rise to a real solution to the equation:

$$A\mathbf{v} = -\mathbf{C}, \tag{5}$$

where

- $\mathbf{v}$ is the velocity vector,
- $A = \begin{pmatrix} a_1 & b_1 \\ \vdots & \vdots \\ a_n & b_n \end{pmatrix}$ and
- $\mathbf{C} = (c_1, \dots, c_n)^T$ are the lines parameters.

Eq. (5) can be solved using a least-square technique that minimizes the quantity $\|A\mathbf{v} + \mathbf{C}\|$, assuming that the system is overdetermined ($n > 2$). However this method is highly sensitive to error.

To minimize the influence of errors, we define a weighting matrix W to update Eq. (5) into:

$$WA\mathbf{v} = -W\mathbf{C}. \tag{6}$$

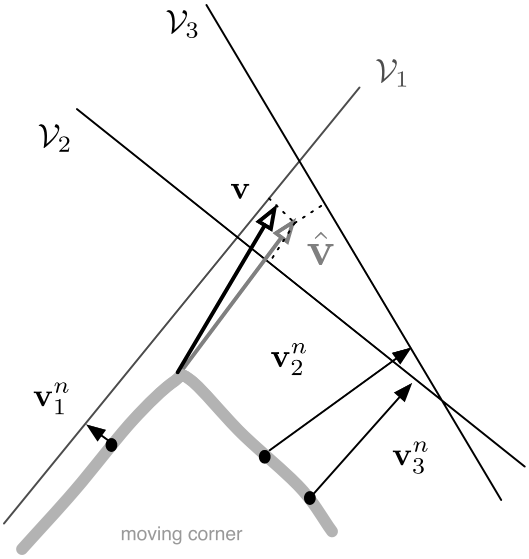


Fig. 6. Example of unweighted least-square technique based estimation of the motion, biased by the over-representation of similar constraints linked to the same line edge. The real motion is represented as vector $\mathbf{v}$. Vectors $\mathbf{v}_1^n$, $\mathbf{v}_2^n$ and $\mathbf{v}_3^n$ correspond to the normal motions estimated around events generated during a small time-window. The least-square minimization provides $\hat{\mathbf{v}}$.

W is the diagonal matrix defined as $\operatorname{diag}\left(\frac{w(\mathbf{\Pi}_1,\mathbf{e})}{\mathcal{L}(\mathbf{\Pi}_1,\mathbf{e})}, \dots, \frac{w(\mathbf{\Pi}_n,\mathbf{e})}{\mathcal{L}(\mathbf{\Pi}_n,\mathbf{e})}\right)$ where the numerator and denominator terms, w and $\mathcal{L}$ are designed to prevent different types of error.

The numerator term w discards wrong constraints introduced by planes computed on sets of events outdated (and not corresponding to the present motion) or generated by a moving edge, close to a corner but not intersecting at its location. Let us assume an event $\mathbf{e}$ and $\mathscr{P}(\mathbf{e})$, the set of planes fitted on previous events within a spatiotemporal neighborhood of $\mathbf{e}$. In this set, some planes can correspond to wrong constraints. The spatial distance $d(\mathbf{e}, \mathbf{\Pi}_l)$, at time t, between an event $\mathbf{e}$ and a plane $\mathbf{\Pi}_l$ (see Fig. 5) is used to discriminate the right constraints from the wrong ones. As illustrated in Fig. 5, this distance can be expressed as $\|\mathbf{v}_l^n\| \left|t - \tilde{t}_l\right|$, with $\tilde{t}_l$ the time at which the plane reaches the position $\mathbf{p}$. Mathematically, $\tilde{t}_l$ is defined such as $\mathbf{\Pi}_\mathbf{l}^T(\mathbf{p}, \tilde{t}_l, 1)^T = 0$.

If the distance $d(\mathbf{e}, \mathbf{\Pi}_l)$ is sufficiently low, the plane $\mathbf{\Pi}_l$ and the associated constraint $\mathcal{V}_l$ can be considered as reliable. The weighting term w is expressed as an exponentially decreasing function applied on this distance, such as:

$$w(\mathbf{\Pi}_l, \mathbf{e}) = \exp\left(-d(\mathbf{e}, \mathbf{\Pi}_l)\right) = \exp\left(-\|\mathbf{v}_l^n\| \left|t - \tilde{t}_l\right|\right). \tag{7}$$

This provides a measure of the reliability of each constraint (Eq. (6)). Note that introducing this condition as a weight in the system instead of fixing a threshold lowers the number of needed parameters.

Moreover, the least-square estimation can be biased if a constraint is over-represented by similar equations in comparison with the other constraint(s). This can happen if an intersecting edge generates more events (and furthermore induces more planes/constraints) than other(s) because the amplitude of its apparent motion is higher. In the example given in Fig. 6, a corner composed of two lines moving at velocity $\mathbf{v}$ is shown. Due to the motion direction and orientations of the intersecting edges, there are more events generated by the right line edge than by the left one. Vectors $\mathbf{v}_1^n$, $\mathbf{v}_2^n$ and $\mathbf{v}_3^n$ correspond to the normal motions estimated around events generated during a very small time-window. The unweighted least-square technique leads to estimate $\hat{\mathbf{v}}$. Geometrically, the peak of $\hat{\mathbf{v}}$ is equidistant to the constraints $\mathcal{V}_{i\in\{1,2,3\}}$. It is mostly influenced by the two similar constraints $\mathcal{V}_2$ and $\mathcal{V}_3$

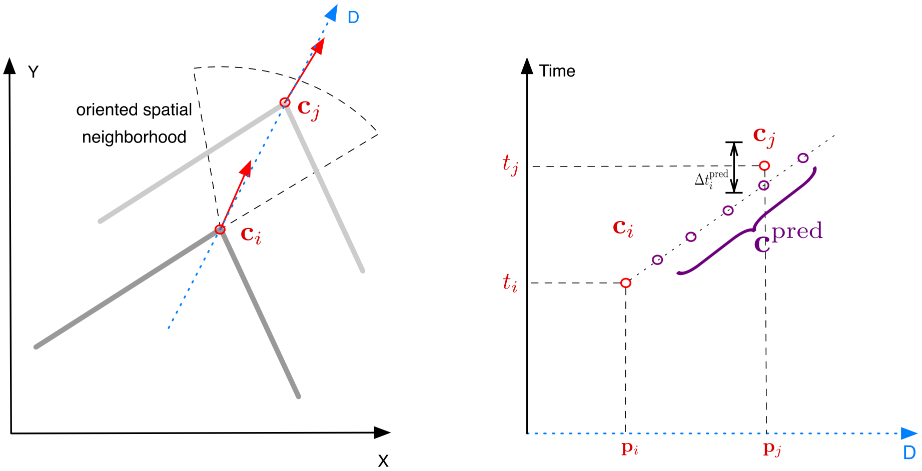


Fig. 7. Principle of the time-interval based matching approach. The left picture represents a moving corner from which 2 corner events, $\mathbf{c}_i$ and $\mathbf{c}_j$, have been estimated. The oriented spatial neighborhood is represented as a dotted shape. The line D shows the corner events trajectory. The right picture shows the predicted corner events, $\mathbf{c}^{\text{pred}}$, generated along the line D and the time interval-based matching between the corner event $\mathbf{c}_j$ and the predicted corner event occurring at the same spatial location.

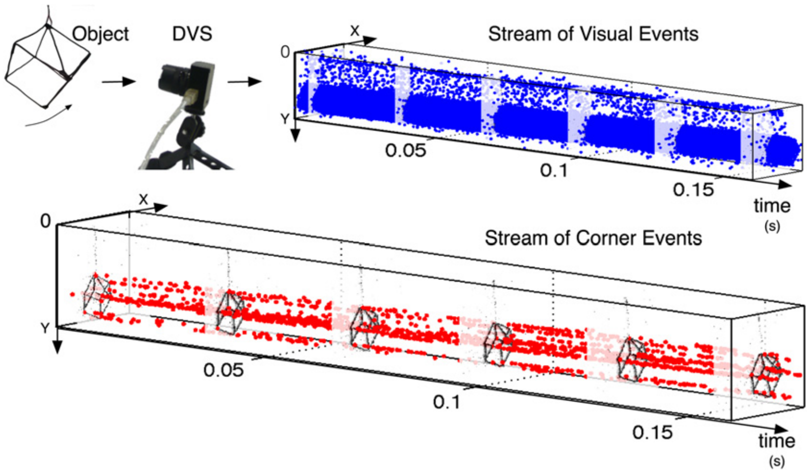


Fig. 8. Experiment 1: (top) streams of events provided by the DVS sensor, in response to a swinging 3D cube, and detected events (bottom). Each dot represents the spatiotemporal locations of an event and snapshots of artificially generated frames are shown every 30 ms. (See video in supplementary material, Appendix A.)

linked to the same edge. A way to reduce this error is to attribute more weight to the constraint $\mathcal{V}_1$, so that it has as much influence as the others lines, e.g., to attribute weight equal to the inverse of the number of similar constraints. In the example of Fig. 6, it means that $\mathcal{V}_2$ and $\mathcal{V}_3$ should be weighted with 1/2 and $\mathcal{V}_1$ with 1. This method requires to determine which constraints are similar.

A likelihood measure L between constraints can then be defined as:

$$L(\boldsymbol{\Pi}_l, \boldsymbol{\Pi}_k) = \delta(pol_l, pol_k)\mathcal{G}^{\tau}(\boldsymbol{\Pi}_l, \boldsymbol{\Pi}_k)\mathcal{G}^{\theta}(\boldsymbol{\Pi}_l, \boldsymbol{\Pi}_k), \tag{8}$$

where:

- $\boldsymbol{\Pi}_l$ and $\boldsymbol{\Pi}_k \in \mathcal{P}(\mathbf{e})$,
- δ is the Dirac function, it is used to consider only the planes computed around events with the same polarity,
- $\mathcal{G}^{\tau}(\boldsymbol{\Pi}_l, \boldsymbol{\Pi}_k) = \exp\left(-\frac{(\|\mathbf{v}_l^n\| - \|\mathbf{v}_k^n\|)^2}{\|\mathbf{v}_l^n\|\|\mathbf{v}_k^n\|}\right)$, corresponds to the velocity amplitude similarity. The normalization term is introduced to consider relative differences instead of absolute differences,
- $\mathcal{G}^{\theta}(\boldsymbol{\Pi}_l, \boldsymbol{\Pi}_k)$ is the velocity direction similarity defined as $\mathcal{G}^{\theta}(\boldsymbol{\Pi}_l, \boldsymbol{\Pi}_k) = \exp\left(-\frac{(\theta_l - \theta_k)^2}{\Theta^2}\right)$, where $\theta_{i\in\{l,k\}}$ is the respective velocity direction of $\mathbf{v}_{i\in\{l,k\}}^n$. Θ is a normalization term referring to the variance of $\theta_l - \theta_k$.

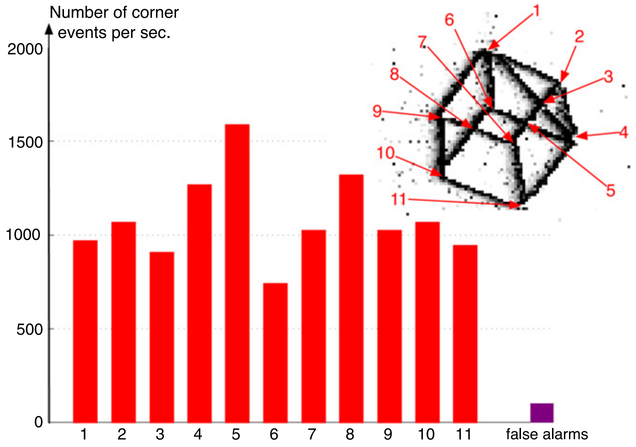


Fig. 9. Experiment 1, evaluation of mean frequencies: number of corner events per second for 11 manually labeled corners of a 3D wire cube (corresponding to the stream of visual events shown in Fig. 8).

Summing this measure for each plane $\boldsymbol{\Pi}_k \in \mathcal{P}(\mathbf{e})$ (weighted by $w(\boldsymbol{\Pi}_l, \mathbf{e})$), provides an estimation of the quantity of reliable

Table 1
Experiment 1, accuracy evaluation of the corner events related to used weighting matrices W: accuracy in terms of errors (compared to the manually labeled ground-truth, noted gt), in spatial distance, velocity direction, velocity magnitude and absolute difference of velocity vectors (first column) is given for different ratios (20%, 50% and 80%) of the population of corner events (second column), e.g. the angular error is mainly less than 9.8° (in bold) for 80% of corner events detected with the proposed method. Note that results are means over the 11 ground-truth's corners: each corner is associated to the spatially closest ground-truth corner's trajectory. Each set of corner events (associated to a ground-truth corner) is sorted according to one of the evaluation criteria (type of errors). The X%-most accurate (20%, 50%, 80%) corner events are then selected. Finally, the accuracy mean for this evaluation criterion is computed overall ground-truth's corners. Different weighting matrices are considered: the first evaluation, noted "W full", gives the accuracy results with W as described in Section 2.2. The second one, noted "W without $\mathcal{L}$-filter", is obtained without the likelihood based plane filter given in Remark 2. The third one, noted "W without $\mathcal{L}$", corresponds to a weighting matrix without dividing by the likelihood, but with the $\mathcal{L}$-filter. The last one considers only a weighting matrix composed by the weighting term w and no $\mathcal{L}$-filter. To obtain a fair evaluation, the thresholds applied to the different selection criteria (obtained with the different weighting matrices) have been set in order to detect the same number of corner events as with our approach (S set to 0.01).

Type of errors	% of Corner Evts	W full	W without $\mathcal{L}$-filter	W without $\mathcal{L}$	W without $\mathcal{L}$-filter and $\mathcal{L}$
Spatial	20%	<0.6 pix.	<7.5 pix.	<6.5 pix.	<7.6 pix.
Distance	50%	<0.9 pix.	<14.4 pix.	<15.1 pix.	<15.2 pix.
$\|\mathbf{p} - \mathbf{p}_{gt}\|$	80%	<1.5 pix.	<22.7 pix.	<23.1 pix.	<23.2 pix.
Velocity	20%	<3.0°	<3.4°	<3.7°	<18.3°
Direction	50%	<6.5°	<8.3°	<8.6°	<76.8°
$\lvert\theta - \theta_{gt}\rvert$	80%	<**9.8**°	<14.0°	<15.9°	<168.3°
Velocity	20%	<3.7%	<3.9%	<5.1%	<32.6%
Magnitude	50%	<7.5%	<10.5%	<13.9%	<63.6%
$\left\lvert\frac{\|\mathbf{v}\|-\|\mathbf{v}_{gt}\|}{\|\mathbf{v}_{gt}\|}\right\rvert$	80%	<11.9%	<21.6%	<29.2%	<95.7%
Abs. Diff.	20%	<9.9%	<13.2%	<14.7%	<54.6%
of Vectors	50%	<15.7%	<20.9%	<23.5%	<98.6%
$\frac{\|\mathbf{v}-\mathbf{v}_{gt}\|}{\|\mathbf{v}_{gt}\|}$	80%	<20.2%	<31.2%	<38.8%	<105.9%

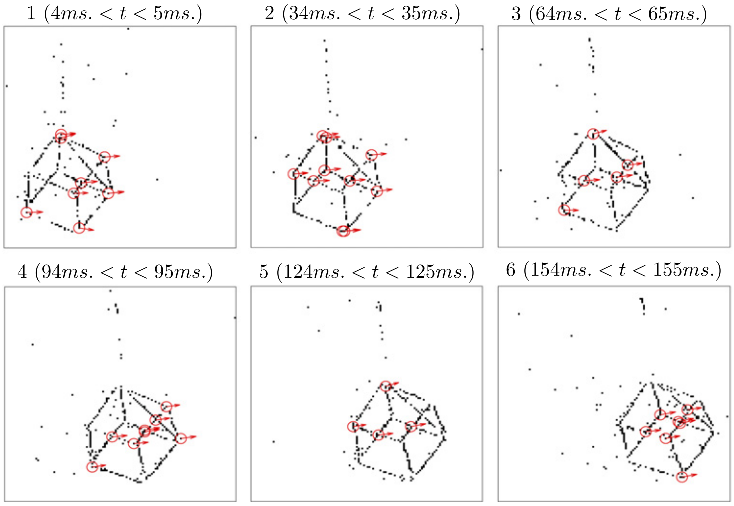


Fig. 10. Results of the Experiment 1: examples of corner events detected corresponding to the stream and snapshots shown in Fig. 8. During the considered time periods (1 ms), some corners are not detected because of the absence of visual events at their spatiotemporal locations: this is particularly visible in the last snapshot (6). In addition, the snapshot (1) illustrates the false alarms in relation to Fig. 9: they are mainly due to the string holding the cube, which introduces sometimes a corner not referenced in the ground-truth.

constraints in the neighborhood of $\mathbf{e}$ given by:

$$\mathcal{L}(\mathbf{\Pi}_l, \mathbf{e}) = \sum_{\mathbf{\Pi}_k \in \mathcal{P}(\mathbf{e})} w(\mathbf{\Pi}_k, \mathbf{e}) L(\mathbf{\Pi}_l, \mathbf{\Pi}_k). \tag{9}$$

Then the over-determined system can be solved if $M = (WA)^T WA$ has a full rank meaning that its two eigenvalues have to be significantly large. This significance is determined with the selection criterion established by Noble (1988):

$$\mathcal{C}(\mathbf{e}) = \frac{\det(M)}{\operatorname{trace}(M)}. \tag{10}$$

Eq. (6) is also solved with a least square minimization technique and solutions are considered as valid if $\mathcal{C}$ is greater than a threshold S usually experimentally set.

Finally, a stream $\mathcal{S}^c$ of corner events, noted $\mathbf{c} = (\mathbf{p}, \mathbf{v}, t)^T$, is obtained with the Algorithm 2 described below.

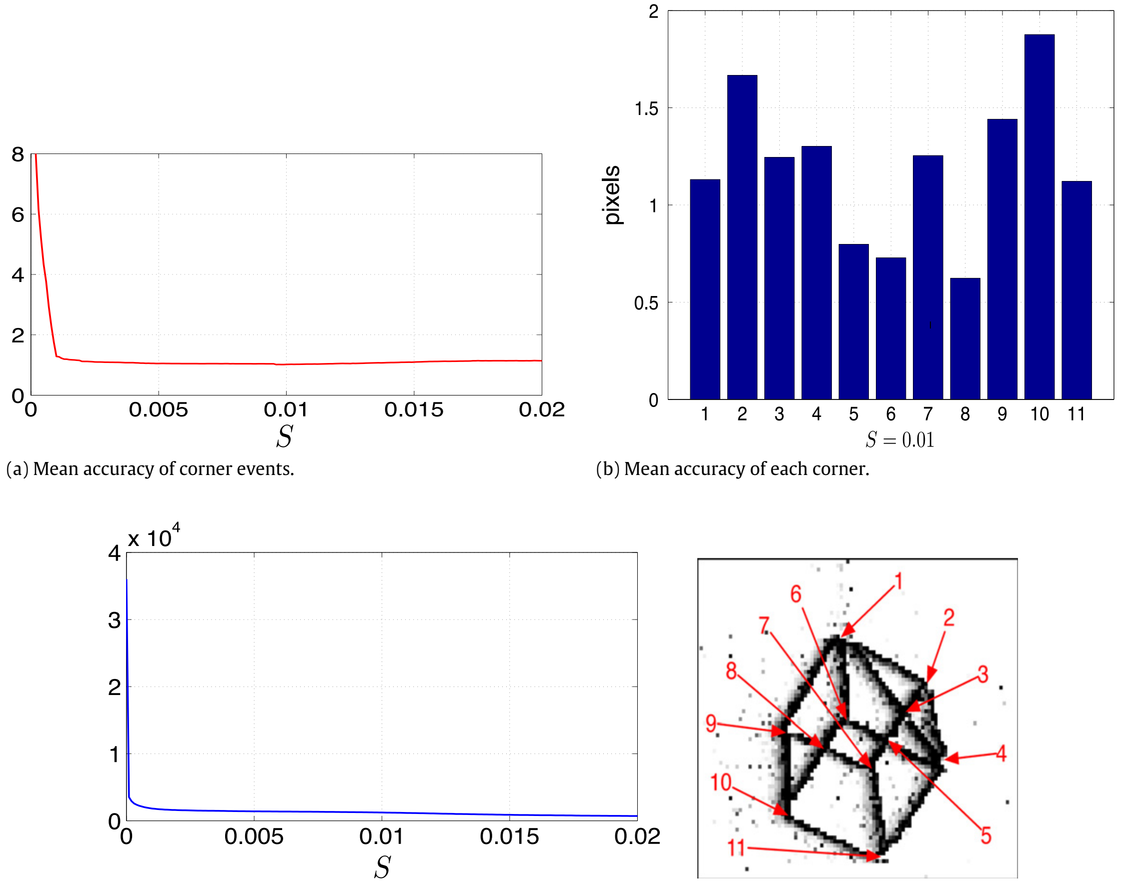
Remark 1. Let $\tau_l = 1/\|\mathbf{v}_l^n\|$, Eq. (7) can be written as:

$$w(\mathbf{\Pi}_l, \mathbf{e}) = \exp\left(-\frac{|t - \tilde{t}_l|}{\tau_l}\right), \tag{11}$$

which is similar to the inter-spikes synchrony measurement defined by van Rossum in van Rossum (2001). It is used to measure the similarity of neuronal responses when submitted to similar stimuli. Similarly, w links spatiotemporal neighboring events as they are likely generated by the same visual stimulus.

This implies that a relation of local connexity between visual events can be quantified mainly using a measure based on a time interval. This allows the presented method to be related to the computational context of temporal coding (Theunissen & Miller, 1995).

Remark 2. A likelihood function is also used to filter out the poorly fitted planes. This mainly happens for events close to edges intersection or for events generated by noise. Planes around



(a) Mean accuracy of corner events.

(b) Mean accuracy of each corner.

(c) Number of corner events.

(d) Labels of the 11 corners.

Fig. 11. Experiment 1, evaluation of spatial accuracy related to the threshold S: evolution related to the threshold S of the number of corner events (c) and of the mean spatial accuracy (a) for the 11 manually labeled corners of a 3D wire cube (d) compared to the ground-truth. The picture (b) gives the mean spatial accuracy for each corner (with S set to 0.01).

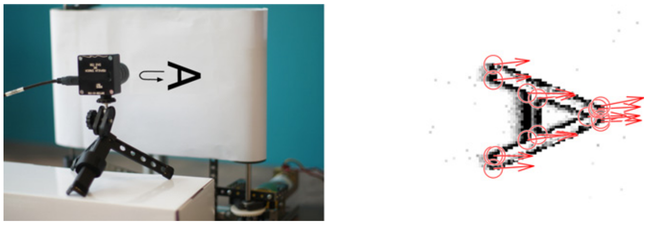


Fig. 12. Experiment 2, translating 'A' pattern: (Left) experimental setup, (right) corner events detected during a time windows of 1 ms (Right).

these events are usually the mean planes going through a set of events incoherent in space and time. Experimentally, only planes satisfying

$$\mathcal{L}(\boldsymbol{\Pi}_l, \mathbf{e}) > 0.5 \max_{\boldsymbol{\Pi}_k \in \mathcal{P}(\mathbf{e})} (\mathcal{L}(\boldsymbol{\Pi}_k, \mathbf{e})) \qquad (12)$$

are preserved.

This operation corrects a side effect of the weighting by the inverse of the likelihood in Eq. (6). It removes wrong constraints introduced when a small set of planes have the same weights of correct constraints. This filter is called $\mathcal{L}$-filter. Its efficiency is demonstrated in the experimental section (see Section 3.1, Table 1).

2.3. Implementation of the event-based detector

The intersection of the constraints is computed grouping planes according to their relative time with each visual event $\mathbf{e}$, expressed as the relative time-based distance w (see Eq. (7) and Remark 1).

In order to not devote too much computation time to a complex search of relevant planes in spatiotemporal neighborhoods with different time scales (because object's intersecting edges could have different dynamics), a stream $\mathcal{S}^n$ of temporary events, called "normal events" and noted $\mathbf{n}_j = (\mathbf{p}_j, \mathbf{v}_i^n, t_j)^T$, is generated for each event $\mathbf{e}_i = (\mathbf{p}_i, t_i, pol_i)^T$. These normal events record the parameter $\mathbf{v}_i^n$ of the plane $\boldsymbol{\Pi}_i$ computed for the visual event $\mathbf{e}_i$ and are defined such as:

$$\boldsymbol{\Pi}_i \begin{pmatrix} \mathbf{p}_j \\ t_j \\ 1 \end{pmatrix} = 0 \qquad (13)$$

where $\mathbf{p}_j$ belongs to a spatial neighborhood of the visual event. This processing is summarized by the Step 3 in the Algorithm 2.

Even if an efficient memory management is required on a conventional computing architecture, the search of relevant planes is reduced to determine which normal events in the stream $\mathcal{S}^n$ occur at the same spatial location and at (approximatively) the same time (i.e. in the time interval defined as $|t - t_l| < \Delta T$) as the visual event (Step 4 in Algorithm 2). Note that the normal events are qualified as temporary because the stream of visual events is causal and only the normal events $\mathbf{n}_j$ with $t_j > t - \Delta T$ are stored in the stream $\mathcal{S}^n$.

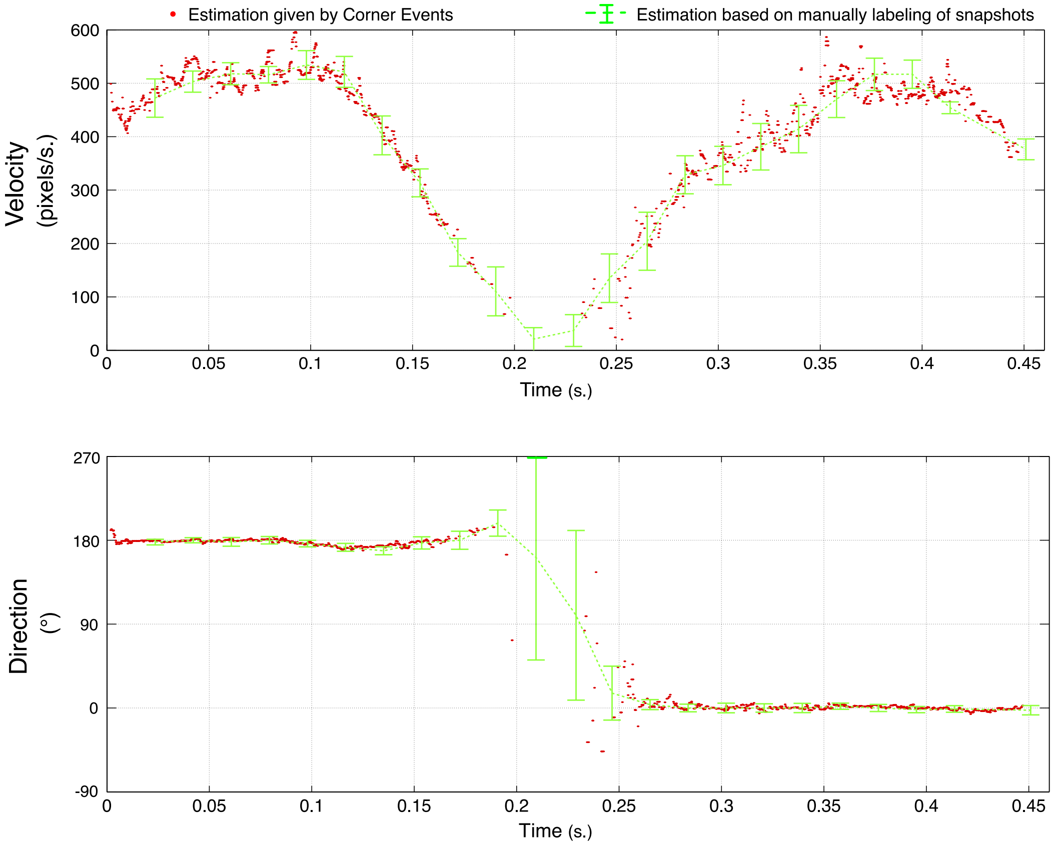


Fig. 13. Evaluation of the accuracy of velocity estimations for experiment 2. Each red point corresponds to the speed of the detected events. The green curves represent the ground-truth. The associated error bars represent the standard deviation error computed with respect to the ground-truth. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

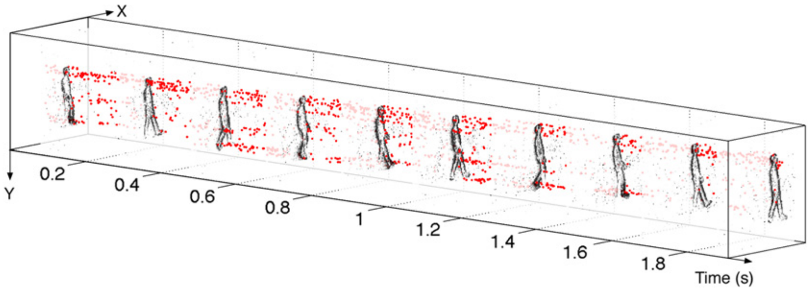


Fig. 14. A stream of corner events obtained in response to a walking person. Each red dot represents the spatiotemporal location of a corner event.

In addition, this algorithm is naturally compatible with recent neuromorphic hardware systems, able to transmit and store local streams of events, and then process them in parallel and in real time, such as those described in Brüderle et al. (2011) and Furber et al. (2013).

Remark 3. The method can be extended to detect corner events even if a visual event is not present at a close temporal location. Indeed a corner event is essentially defined by the spatiotemporal intersection of several planes into the same spatiotemporal line (see Fig. 4).

But such an approach implies a very expensive computation load for a conventional computing architecture; it would require to evaluate systematically all normal events and could generate "illusory corners" (corner events at spatiotemporal locations where object's edge does not exist).

2.4. Time-interval based corner event matching

Two corner events, $\mathbf{c}_i$ and $\mathbf{c}_j$ with $t_j > t_i$, can be matched considering their spatiotemporal relations and respective velocities. In the proposed approach, they are considered as belonging to the same corner if the following conditions are satisfied:

$$\begin{cases} \|\mathbf{p}_j - \mathbf{p}_i\| < \Delta\mathbf{p}, \\ |\alpha_{ij} - \theta_i| < \Delta\alpha, \\ |t_j - t_{ij}^{\text{pred}}| < \dfrac{1}{2}\Delta t_i^{\text{pred}}, \\ \dfrac{\|\mathbf{v}_j - \mathbf{v}_i\|}{\|\mathbf{v}_i\|} < \Delta V, \end{cases} \tag{14}$$

where $\Delta\mathbf{p}$, $\Delta\alpha$ and ΔV are experimentally set (see Section 3.2), $\alpha_{ij} = \text{angle}(\mathbf{p}_j - \mathbf{p}_i)$ is the angle between both spatial locations of

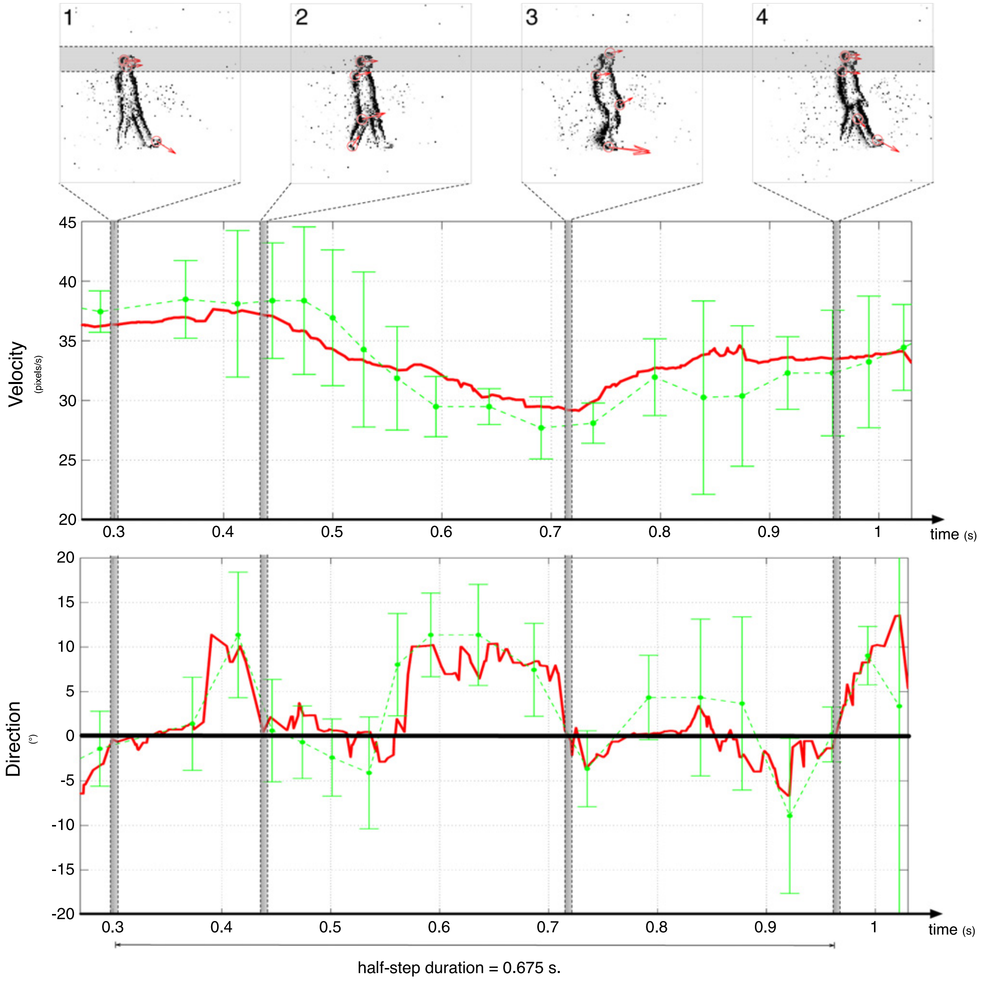


Fig. 15. Results of the Experiment 3: evolution of the head velocity obtained by collecting the velocity amplitudes and directions of the corner events included in a region of interest (the gray zone in the top snapshots). The green and dashed curves correspond to the ground-truth. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 2 Computation of the intersection of the constraints

```
for all event $\mathbf{e} = (\mathbf{p}, t, pol)^T$ do
    Compute the plane $\mathbf{\Pi}$ and $\mathbf{v}^n$ (Algorithm 1).
    Compute the set $\{\mathbf{n}_j\}$ (Eq.13) and store them into the stream $\mathcal{S}^n$
    Determine the set of normal events $\{\mathbf{n}_l\}$ in stream $\mathcal{S}^n$, such as $\begin{cases} \mathbf{p}_l = \mathbf{p} \\ \mid t_l - t \mid < \Delta T \end{cases}$
    for all event $\mathbf{n}_l = (\mathbf{p}, \mathbf{v}_l^n, t_l)^T$ do
        Compute weight $w(\mathbf{\Pi}_l, \mathbf{e})$ (Eq. 7)
        Compute likelihood measures $\mathcal{L}(\mathbf{\Pi}_l, \mathbf{\Pi}_k)$ (Eq. 8)
    end for
    Compute $\mathcal{L}(\mathbf{\Pi}_l, \mathbf{e})$ for all constraints (Eq. 9)
    Compute matrix $M = (WA)^T WA$.
    Compute the selection criterion $\mathcal{C}(\mathbf{e})$ (Eq. 10)
    if $\mathcal{C}(\mathbf{e}) > S$ then
        Mark $\mathbf{e}$ as a corner event $\mathbf{c}$
        Solve the equation system $WA\mathbf{v} = -W\mathbf{C}$ (Eq.6)
        Set $\mathbf{v}$ to the corner event $\mathbf{c}$
        Store $\mathbf{c}$ into the stream $\mathcal{S}^c$
    end if
end for
```

the corner events and $t_{ij}^{pred} = \frac{\|\mathbf{p}_j - \mathbf{p}_i\|}{\|\mathbf{v}_i\|}$ is the traveling time. Δt_i^{pred} is set as an estimated time during which the corner "travels" through a pixel (and can probably generated another corner event), such as $\Delta t_i^{\text{pred}} = \gamma \frac{1}{\|\mathbf{v}_i\|} + 2\nu$, where γ is a multiplicative factor and ν is an added constant value in order to take into account of, respectively, error in velocities estimation and noise inherent to the sensor's electronic board.

The first two conditions define an oriented spatial neighborhood with the location of corner event $\mathbf{c}_i$ as its origin point. The third one defines a temporal relation similar to an inter-spike synchrony measurement. This inspired us an implementation for matching, similar to the one described in the previous section. For each detected corner event, temporary events, called "predicted corner events" and noted $\mathbf{c}^{\text{pred}}$, are generated for each pixel locations in its oriented neighborhood and matched with future detected corner events as illustrated in Fig. 7. The last condition introduces an assumption on the velocity of the corner: it does not change brutally during the time interval between both corner event detections.

Remark 4. In order to take into account the inherent noise on visual event timing (noted ν and essentially due to sensor's electronic board), a fifth constraint is added: $t_j - t_i > \Delta t^{\text{noise}}$, with $\Delta t^{\text{noise}} > 2\nu$. This allows us to prevent the case in which two events (a corner event and a predicted corner event) occur in the same time interval because of the noise.

3. Experimental results

Several sets of experiments have been carried out on natural scenes. Two of them are achieved in indoor environments.

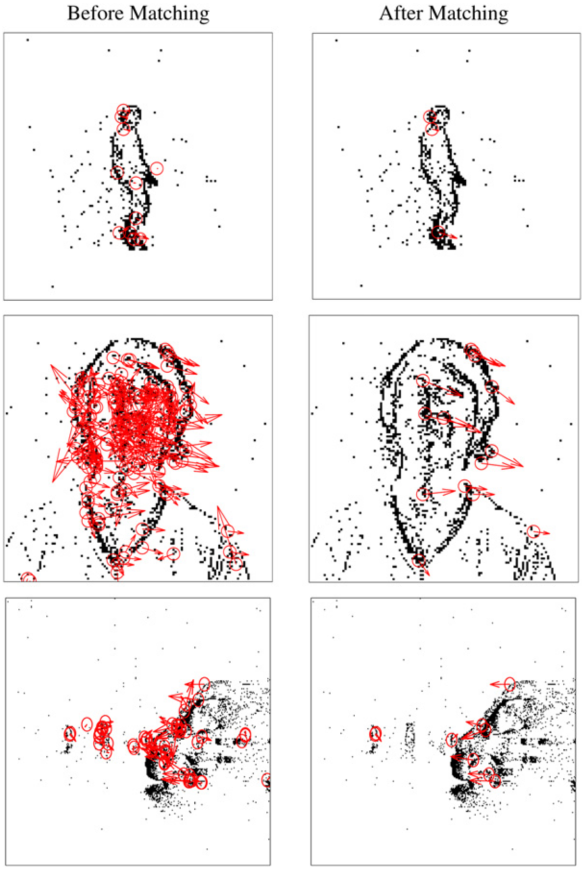


Fig. 16. Examples of corner events computed in the same time window, but before (left) and after (right) matching.

The indoor experiments involved single and contrasted objects, moving in front of the sensor. Outdoor experiments are carried out under uncontrived conditions to assess the algorithms robustness.

3.1. Corner event detection

For the plane fitting step, the spatiotemporal window $\Omega(\mathbf{e})$ is selected experimentally. In what follows $R = \sqrt{2}$ pixels and $N = 15$. In Algorithm 1, the threshold in step 5 can be set to $th_1 = 1e{-}5$; it is usually the magnitude of accuracy we get from this iterative estimation. The second threshold in step 6 is set to $th_2 = 1e{-}6$, which is also the typical accuracy we get from experiments when fitting planes to surfaces swept by moving edges. To collect the constraints, the spatiotemporal neighborhood of $\mathbf{e}$ from which $\mathcal{P}(\mathbf{e})$ is built, is defined experimentally with $\max(t_l - t) \leq \Delta T = 0.01$ s. and $|\mathbf{p} - \mathbf{p}_l| \leq 7$pixels. Θ has been experimentally set to 20°. Finally the threshold S on the corner events' selection criterion is experimentally set to 0.01.

The two indoor experiments are carried out with objects placed in front of a white background to ensure high contrast stimulus (Figs. 8 and 12). Frames are generated by summing events over a time period of 2 ms. from which we built a ground-truth. Corner pixels are labeled manually in each frame and their positions and velocities between two frames are linearly interpolated at timings of all visual events.

A swinging 3D cube is shown to the silicon retina. Fig. 8 shows the detected events for the entire sequence. On the top row, the raw signal is shown while the second row is showing only the detected events. Eleven corners (or junctions) have been detected by the method. Results are shown by the histogram shown in Fig. 9. They provide a measurement in terms of mean frequency of correct and false detections. The method is robust and reliable in localizing corners in space and time. A corner event is considered as a good detection if its spatial distance to a ground truth trajectory is less than or equal to 2 pixels. Examples of corner events detected for time windows of 1 ms are shown in Fig. 10. The method is able to estimate the location of 1200 detected events per second with less than 0.5% of false detections. The false detections are mainly due to the string holding the cube, which introduce sometimes a corner, not referenced in the ground-truth as illustrated in Fig. 10.1. Fig. 11 shows the mean accuracy (in pixels) obtained with different values of the threshold S applied to the selection criterion $\mathcal{C}(\mathbf{e})$ (Eq. (10)). It represents the spatial distance between corner events and location of the closest ground-truth's corner (linearly interpolated at each visual event's timing from the ground-truth). Globally (Fig. 11.a), the mean accuracy is around 1 pixel for a large range of S. This is consistent with the temporal precision of the visual events provided by the DVS sensor estimated under 4 ms (Akolkar et al., 2015). Indeed, it means a spatial accuracy for visual events of around 1 pixel, regarding the speed of the object meanly estimated around 400 pixels per second.

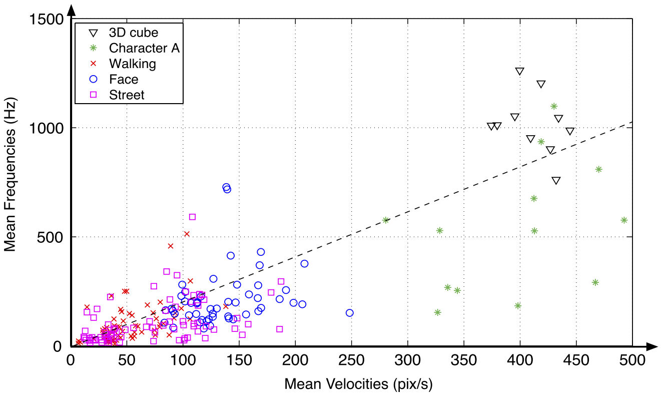


Fig. 17. Mean frequencies of the matched corner events (linked into tracks) related to their mean velocities. A linear regression of these data is represented as a dashed curve in order to highlight the high correlation between corner event frequencies and estimated object dynamics (the correlation coefficient is equal to 0.76), due to the asynchronous acquisition provided by neuromorphic sensor.

Table 2
Ratio of matched corner events.

Stream name	Matched corner events
3D Cube (Fig. 18)	99.5%
Character A (Fig. 19)	79.2%
Walking (Fig. 20)	46.0%
Face (Fig. 21)	8.2%
Street (Fig. 22)	13.1%

A more complete accuracy evaluation is given in Table 1. The corner events' parameters (spatial location and velocity) and the 11 corners' ones (obtained with the ground-truth) are compared using different measures of errors. Each corner event is associated to the spatially closest ground-truth corner. The obtained results show that most corners are accurate in terms of localizations and velocity estimations. In addition, they show that the introduction of the likelihood in the weighting matrix and the $\mathcal{L}$-filter improve the detector performances.

The second experiment is designed to evaluate the accuracy of the corner events' velocities provided by solving Eq. (6). They are compared with a ground truth motion built by labeling manually corners in the generated frames. A 2D printed pattern is mounted on a treadmill and is placed parallel to the camera (see Fig. 12). The pattern is translated at various speeds.

Fig. 13 shows the pattern's estimated velocity. The top row represents the amplitude and the bottom one, its orientation.

Table 3
Accuracy evaluation of matched corner events.

Type of error	% of matched corner events	3D cube (Fig. 18)	Character A (Fig. 19)	Walking (head) (Fig. 20)	Face (head) (Fig. 21)	Street (van) (Fig. 22)
Velocity	20%	<2.8°	<1.5°	<3.1°	<3.5°	<2.4°
Direction	50%	<6.3°	<4.4°	<7.9°	<9.9°	<8.6°
$\lvert\theta - \theta_{gt}\rvert$	80%	<9.5°	<4.6°	<10.5°	<15.5°	<12.5°
Velocity	20%	<3.6%	<2.8%	<4.0%	<5.0%	<4.6%
Amplitude	50%	<7.0%	<7.1%	<9.0%	<13.3%	<8.1%
$\left\lvert\frac{\lVert\mathbf{v}\rVert - \lVert\mathbf{v}_{gt}\rVert}{\lVert\mathbf{v}_{gt}\rVert}\right\rvert$	80%	<11.4%	<12.5%	<14.0%	<20.3%	<15.6%
Abs. Diff	20%	<9.8%	<6.0%	<11.3%	<12.1%	<11.7%
of Vectors	50%	<15.5%	<10.7%	<17.8%	<18.8%	<16.2%
$\frac{\lVert\mathbf{v}-\mathbf{v}_{gt}\rVert}{\lVert\mathbf{v}_{gt}\rVert}$	80%	<19.8%	<17.6%	<21.5%	<24.1%	<21.8%

Table 4
Survival time of tracks (linking matched corner events).

Stream name	Stream duration (s)	Survival time (in % of stream duration)				
		Mean	Standard deviation	Median	Min	Max
3D Cube (Fig. 18)	0.16	95.3%	3.5%	96.0%	86.3%	98.6%
Character A (Fig. 19)	0.45	21.0%	15.1%	29.1%	2.7%	40.0%
Walking (Fig. 20)	1.90	13.3%	21.2%	6.6%	0.7%	97.6%
Face (Fig. 21)	0.15	40.7%	16.1%	34.9%	14.5%	74.4%
Street (Fig. 22)	0.53	15.4%	13.7%	10.2%	2.0%	75.6%

Table 5
Frequencies of matched corner events linked into tracks.

Stream name	Matched corner events frequencies				
	Mean (Hz)	Standard deviation (Hz)	Median (Hz)	Min (Hz)	Max (Hz)
3D Cube (Fig. 18)	934	126	1009	762	1543
Character A (Fig. 19)	753	764	569	153	2970
Walking (Fig. 20)	84	84	61	9	513
Face (Fig. 21)	215	131	191	80	728
Street (Fig. 22)	112	88	82	12	539

Estimated data are shown as sparse red dots, the ground truth is given by the dashed curves. The detected events follow closely the ground truth. At low speed, the detection is poor because the stimulus is not producing a sufficient amount of events to accurately estimate the velocity. This is an expected result that is due to the silicon retina transient properties and the used settings. Another interesting result is provided by the comparison with the accuracy of the manually labeled ground-truth. To build it, all 11 corners of the character 'A' have been carefully labeled in frames (generated by collecting events over a time period as explained above). For each labeled point, its positions are derived in order to obtain velocity estimations and the ground-truth values (dashed green curve) correspond to the mean values computed for each frame considering the fact they should be equal for all labeled points in the same frame. Vertical error bars represent the standard deviation errors obtained for these manual estimations. Fig. 13 shows that the accuracy of the velocity estimations provided by the corner events detector is closely similar to the manually obtained ones. In the next section, quantitative evaluations (see Table 3) considering only corner events matched with the method explained in Section 2.4 will show that the accuracy for this experiment is globally similar to the ones obtained with the 3D cube (Table 1).

3.2. Time-interval based corner event matching

To complete these first analyses, three event streams acquired from more complicated and outdoor scenes are added to the evaluation set.

An experiment uses an unconstrained sequence of a person walking in front of the silicon retina. Fig. 14 shows the spatiotemporal stream of corner events, detected by the method. Frames are built only for clarity purpose by showing where corner events are generated. Most of them are localized around the head, the feet and the hands. A sub-sequence is shown in Fig. 15 to emphasize the motion consistency from the estimated velocities (red, plain curve). We focused on the cluster of events detected around the head and plot their mean velocity amplitude and direction. As it can be seen, the amplitude is oscillating as the head is subject to acceleration changes when weight transfers from one leg to another. Particular key points from these curves (zeros-crossing of the velocity direction) can be extracted; events characterizing the gait cycle (Whittle, 1996) such as the foot-strikes (key points 1 and 4), the heel-off (key-point 2) or the midswing (key-point 3) are identified very precisely at these key-points. The comparison with the ground truth (green and dashed curves) validates the obtained head motion estimations. To build it, the same method explained for Experiment 2 (Fig. 13) is used, but only three points have been selected, such that they correspond to the limits between the neck and the head and to the top of the head.

Another experiment uses a sequence of events acquired observing a human tilting his head, in front of the sensor (see Fig. 16-middle line- and Fig. 21). This stream is very difficult because of the side lighting conditions, causing diffuse shadows on the face (especially in the middle), and the texture of the person's hair. As shown in Fig. 16 (left column), several corners are detected, but many of them do not correspond to the head motion.

A complex street scene is processed. Several persons are walking (background scene, see Figs. 16, 22 and 23) while a van passes in front. Processing the acquired stream is very challenging. People are far and are represented by small sets of pixels. The van is moving horizontally and most of its edges are parallel to the motion direction; these edges generate few events. Conversely, some 3D structures (bumper, van's passenger compartment visible

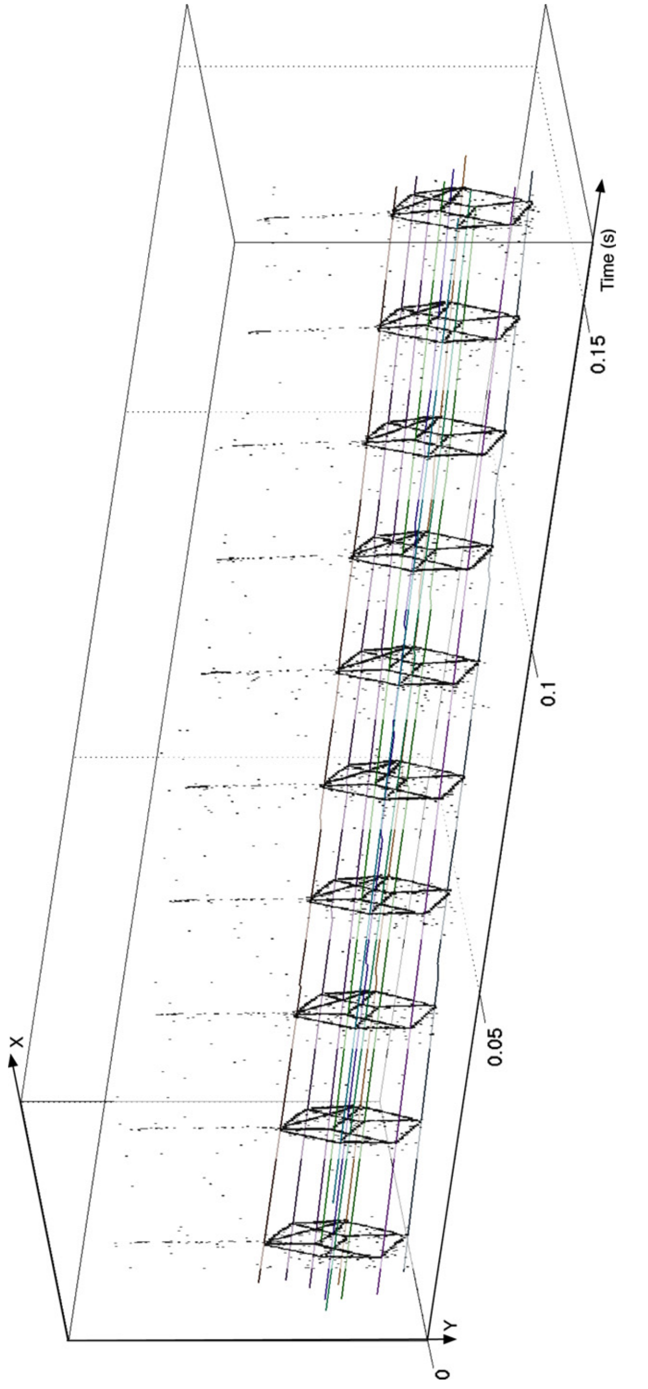


Fig. 18. Tracks obtained with the stream of corner events provided from the moving 3D cube.

through the windows, wheel compartments, rear-view mirror, etc.) cause shadows and subpixellic textures, and then generate very dense and large local event streams not necessarily related to the vehicle's motion. In addition, moving objects are present in the background of the scene but are too far to be identified or generate regularly events. In these conditions, many corner events are incorrectly detected and correspond to false alarms (see Fig. 16, left column).

Accuracy evaluations (despite these extreme conditions) are shown in Section 2.4. The parameters for matching have been set according to the evaluation results obtained for the 3D cube in Table 1 (for 80% of the detected corners): $\Delta\alpha$, ΔV and γ are respectively fixed to 10°, 20% and 1.1. In addition, ν, Δt^{noise} and $\Delta\mathbf{p}$ have been experimentally set to 4 ms, 10 ms and 4 pixels. Only corners matched at least 1 time will be preserved and evaluated. As shown in Fig. 16 (right column), they concern globally the principal motions of objects in the scenes.

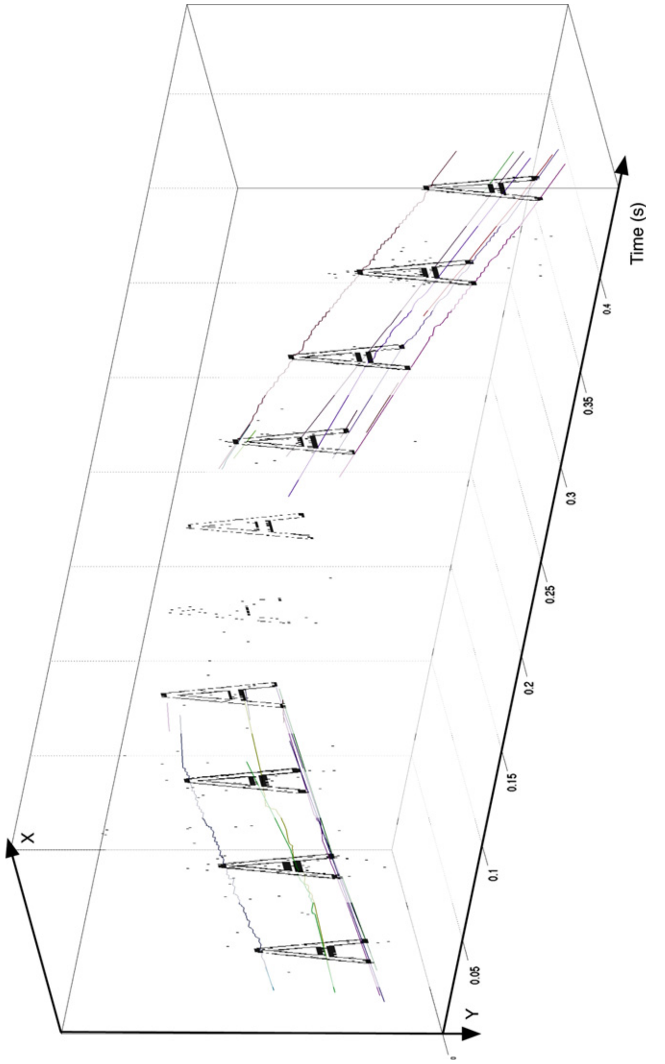


Fig. 19. Tracks obtained with the stream of corner events provided from the moving character A.

Table 2 reviews the ratio of preserved corner events, called "matched corner events", for the five experiments. Unsurprisingly, these ratios are high for the two first experiments. Concerning the 3D cube stream, it matches well to the results obtained in Fig. 9: 0.5% of the corner events are rejected by the matching process. This corresponds to the same proportion of false alarms shown in Fig. 9. The ratio is lower for the Experiment 2 (Character A). Most of corner events during the direction change period are rejected because the motion is unstable (rattling motions of the motorized mechanical structure supporting the pattern). For the outdoor experiments, ratios go even lower. Most rejected corner events concern deformable structures (hair, shadows, clothes in the "face" and "walking" streams), the further objects (in the "street" stream) and the previously mentioned 3D structures of the van.

Considering only the matched corner events, accuracy evaluations, similar to the ones provided in Table 1, are shown in Table 3. For outdoor experiments, evaluation is restricted to non-deformable objects, such as heads and faces or the van of the street

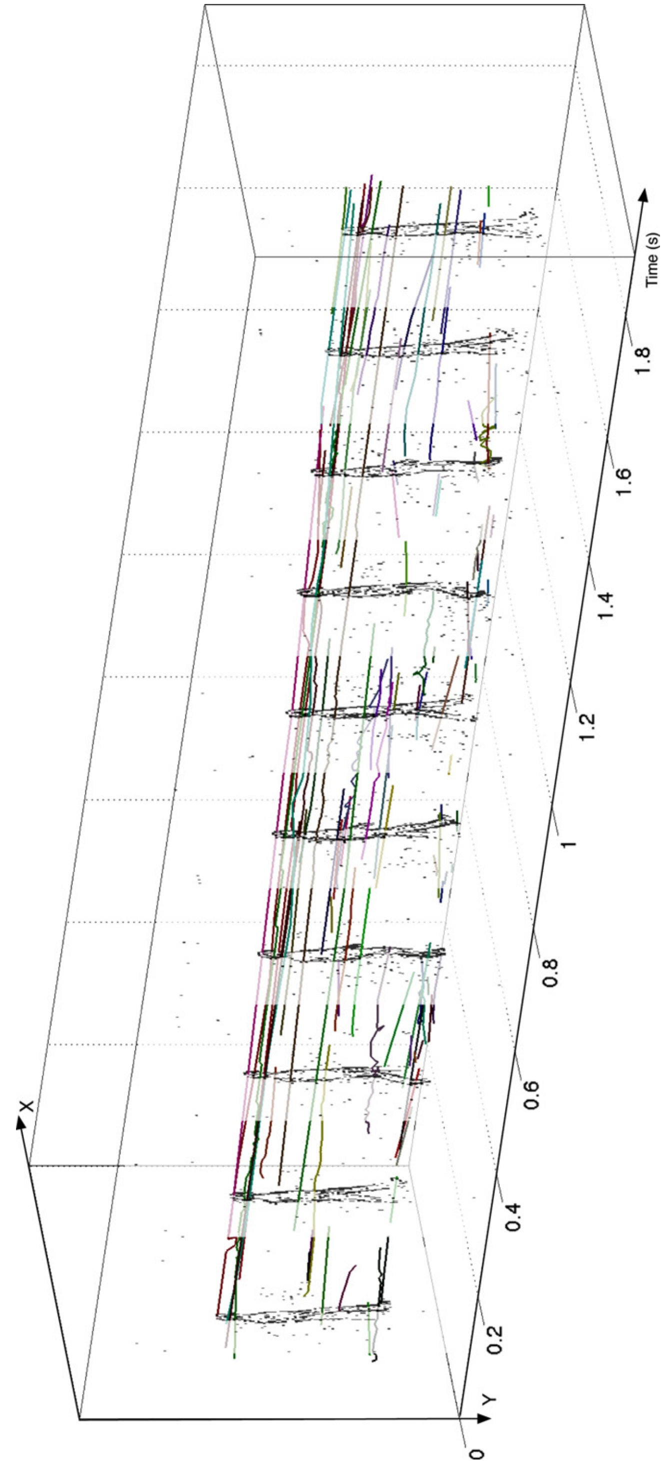


Fig. 20. Tracks obtained with the stream of corner events provided from a walking person.

sequence. In this case, the ground-truth corners and regions of interest can be clearly identified and labeled every 10 ms. Minor variations across accuracy results are observed even when scenes contain different objects moving with different velocities. The best results have been obtained with the character ‘A’ stream; it is likely because corners in the pattern are clearly defined and motion is well controlled (excepting during the direction change time period). The lowest performances are achieved for the face stream. In addition to the extreme conditions described above, the 3D shape of the face (and haircut) might result in differences between ground-truth and velocity estimations.

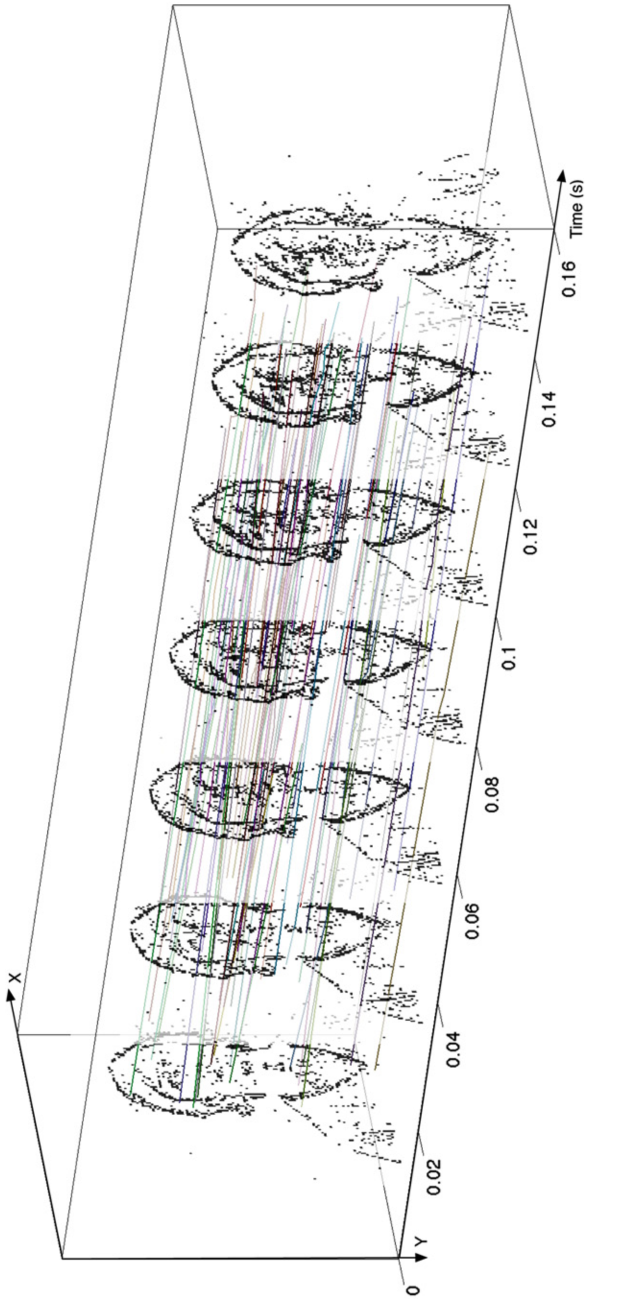


Fig. 21. Tracks obtained with the stream of corner events provided from a moving human face.

Using the matching process, tracks (Figs. 18–23) can be obtained linking matched corner events. This allows estimating the survival times for these corner events (Table 4). For streams acquired from deformable objects (walking persons in walking and street streams), the appearance times are obviously shorter because the corners can appear and disappear according to the motion.

Linking tracks built before and after direction changes requires to define descriptors extracted from the neighborhood of the corner events and distances to match them. In the matching process defined in Section 2.4, such feature distances could be used as a substitute for the assumption on corner velocity (fourth

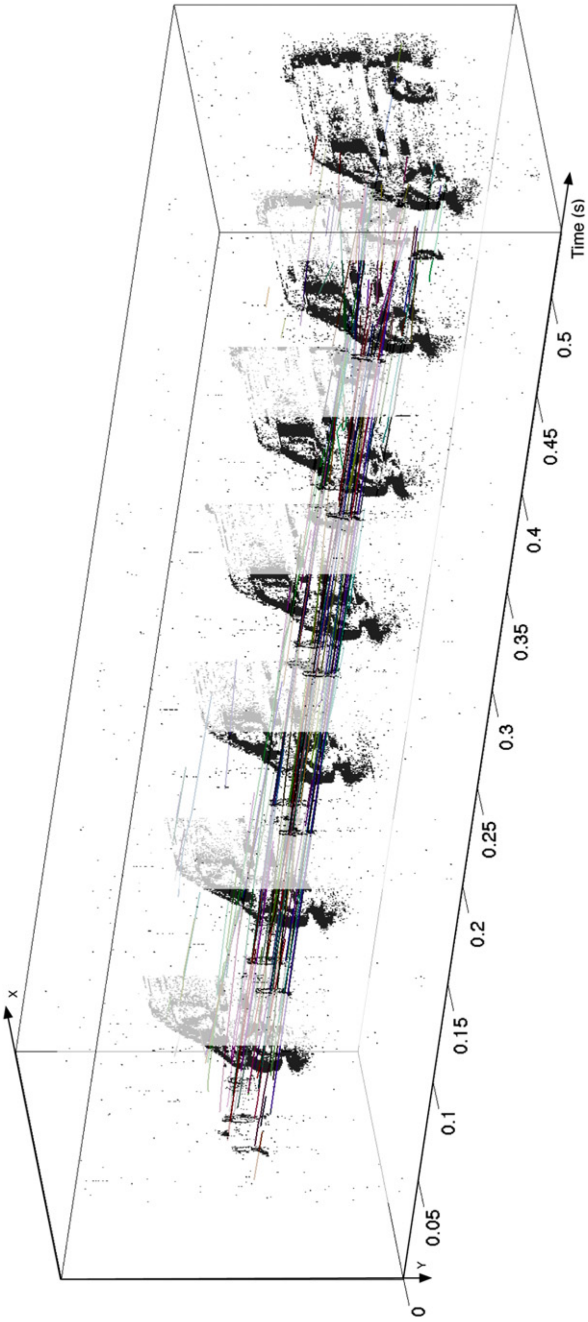


Fig. 22. Tracks obtained with the stream of corner events provided from moving persons and vehicles in a street.

condition in Eq. (14)). In addition, such approaches would allow disambiguating tracks. Indeed, as observed in Fig. 19 (character A stream), individual tracks have merged because corners (in particular the 3 corners at the top of the character A) are spatially close with the same motion. Consequently wrong matches can be computed. This is also reflected in the frequencies of corner events grouped in tracks (Table 5): some of them are particularly high and are not necessarily related to the object dynamics, as it is the case with other streams (see Fig. 17).

Descriptors for sparse features, taking full advantages of neuromorphic sensors, still remain an open issue; determining the optimal spatiotemporal scale (of the neighborhood) and information coding (in relation to targeted applications) are challenges to be addressed for asynchronous event-based stream, as it has been done for synchronous frame-based data (Dickscheid et al., 2011; Gauglitz et al., 2011; Gil et al., 2010; Laptev, 2005; Mikolajczyk & Schmid, 2005; Mokhtarian & Mohanna, 2006; Moreels & Perona, 2007). Recent works about event-based convolutional networks (Perez-Carrasco et al., 2013) and Gabor-like orientation-selective cells (Akolkar et al., 2015) for asynchronous visual feature extraction are among the solutions to be explored.

4. Discussion and conclusions

This paper shows that time is an essential component in corner detection when dealing with event-based visual acquisition. The precise timing conveyed by the neuromorphic asynchronous event-based retina ensures accurate corner detection because of the ability to estimate accurately the events' velocity. Indeed a remarkable result is the accuracy of the detected features in spite of the poor spatial resolution of the retina. All computations are strictly local and incremental. In conventional frame-based processing, the low temporal resolution implies additional computational costs as large neighborhoods and multi-scale approaches are often mandatory to increase the robustness and accuracy of corner points detectors. Neuromorphic retinas open new perspectives for artificial vision as they are clearly more efficient energetically and also computationally.

Another observation can be made on the nature of the information required for the corner detection: a fitted plane that models the motion (i.e. position and velocity at each timing) of the visual stimuli. The information about the "cornerness" of the local feature is implicitly encoded within the information of its local motion. We underline also that the knowledge of motion leads to the measurement of local connexity between visual events through a temporal coding mechanism. It is interesting to notice that these information are encoded in the primary area (V1) of the visual cortex via the selectivity of V1 neurons. This implicit inclusion of the geometric information might explain why motion information driven by the dorsal pathway in the visual cortex can be sufficient to perform classification of biological movements, such as human body motion (Giese & Poggio, 2003), as it is operated in recent approaches for action recognition (Kaaniche & Bremond, 2010; Wang, Klaser, Schmid, & Liu, 2013); features are extracted in each frame and matching across time in order to extract local trajectories. Because it performs event-driven corner detection, asynchronous event-based paradigm might then provide a natural and efficient solution to this problem.

Acknowledgments

This work performed in the frame of the LABEX LIFESENSES [ANR-10-LABX-65] was supported by French state funds managed by the ANR within the Investissements d'Avenir programme [ANR-11-IDEX-0004-02]. The authors are also grateful to both the CapoCaccia Cognitive Neuromorphic Engineering Workshop and the NSF Telluride Neuromorphic Cognition workshops.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.neunet.2015.02.013.

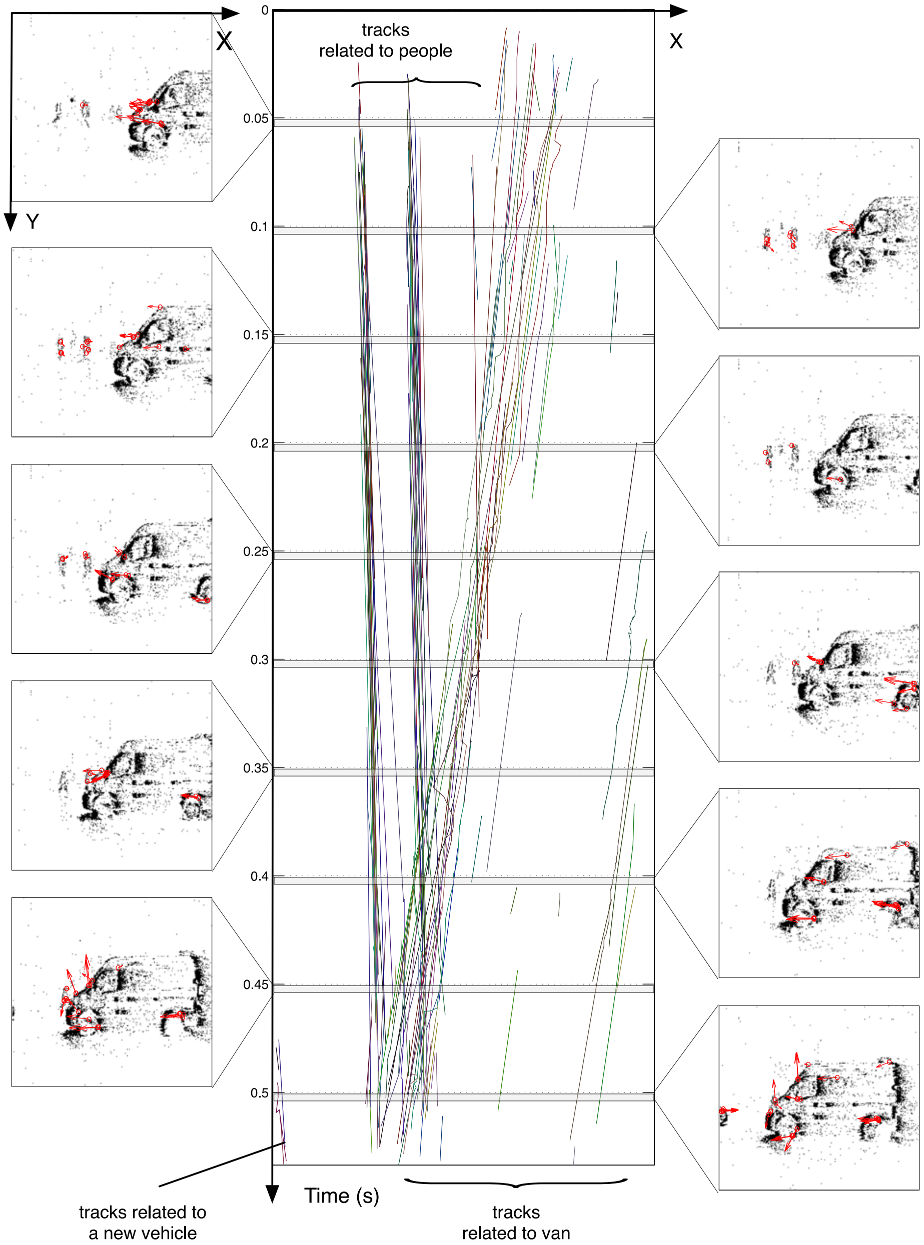


Fig. 23. Tracks and snapshots of the matched corner events obtained from the street stream.

References

Adelson, E., & Movshon, J. (1982). Phenomenal coherence of moving visual patterns. *Nature*, *200*(5892), 523–525.

Akolkar, H., Meyer, C., Clady, X., Marre, O., Bartolozzi, C., Panzeri, S., & Benosman, R. (2015). What can neuromorphic event-driven precise timing add to spike-based pattern recognition? *Neural Computation*, *27*(3), 561–593. http://dx.doi.org/10.1162/NECO_A_00703.

Benosman, R., Clercq, C., Lagorce, X., Ieng, S., & Bartolozzi, C. (2014). Event-based visual flow. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(2), 407–417. http://dx.doi.org/10.1109/TNNLS.2013.2273537.

Brüderle, D., Petrovici, M. A., Vogginger, B., Ehrlich, M., Pfeil, T., Millner, S., et al. (2011). A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems. *Biological Cybernetics*, *104*(4–5), 263–296.

Delbruck, T., Linares-Barranco, B., Culurciello, E., & Posch, C. (2010). Activity-driven, event-based vision sensors, ISCAS (pp. 2426–2429).

Deselaers, T., Keysers, D., & Ney, H. (2008). Features for image retrieval: an experimental comparison. *Information Retrieval*, *11*(2), 77–107.

Dickscheid, T., Schindler, F., & Förstner, W. (2011). Coding images with local features. *International Journal of Computer Vision*, *94*(2), 154–174.

Furber, S., Lester, D., Plana, L., Garside, J., Painkras, E., Temple, S., et al. (2013). Overview of the spinnaker system architecture. *IEEE Transactions on Computers*, *62*(12), 2454–2467.

Gauglitz, S., Höllerer, T., & Turk, M. (2011). Evaluation of interest point detectors and feature descriptors for visual tracking. *International Journal of Computer Vision*, *94*, 335–360.

Giese, M. A., & Poggio, T. (2003). Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*, *4*(3), 179–192.

Gil, A., Mozos, O. M., Ballesta, M., & Reinoso, O. (2010). A comparative evaluation of interest point detectors and local descriptors for visual slam. *Machine Vision and Applications*, *21*(6), 905–920.

Harris, C., & Stephens, M. (1988). A combined corner and edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 147–151.

Kaaniche, M. B., & Bremond, F. (2010). Gesture recognition by learning local motion signatures. In *2010 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2745–2752). IEEE.

Laptev, I. (2005). On space–time interest points. *International Journal of Computer Vision*, *64*(2/3), 107–123.

Lategahn, H., Gross, S., Steh, T., & Aach, T. (2010). Texture classification by modeling joint distributions of local patterns with Gaussian mixtures. *IEEE Transactions on Image Processing*, *19*(6), 1548–1557.

Li, J., & Allinson, N. M. (2008). A comprehensive review of current local features for computer vision. *Neurocomputing*, *71*, 1771–1787.

Lichtsteiner, P., Posch, C., & Delbruck, T. (2008). A 128 ∗ 128 120 dB 15us latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid State Circuits*, *43*(2), 566–576.

Mahowald, M. (1992). *VLSI analogs of neuronal visual processing: A synthesis of form and function* (Ph.D. thesis). Pasadena, California.

Mikolajczyk, K., Leibe, B., & Schiele, B. (2005). Local features for object class recognition. In *Tenth IEEE international conference on computer vision, 2005. vol. 2*, ICCV 2005. (pp. 1792–1799). IEEE.

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(10), 1615–1630.

Mokhtarian, F., & Mohanna, F. (2006). Performance evaluation of corner detectors using consistency and accuracy measureness. *Computer Vision and Image Understanding*, *102*(1), 81–94.

Mokhtarian, F., & Suomela, R. (1998). Robust image corner detection through curvature scale space. *IEEE Trans on Pattern Analysis and Machine Intelligence*, *20*(12), 1376–1381.

Moravec, H. (1980). *Obstacle avoidance and navigation in the real world by a seeing robot rover, Tech. rep., CMU-RI-TR-80-03*. Robotics Institute, Carnegie Mellon University and doctoral dissertation, Stanford University.

Moreels, P., & Perona, P. (2007). Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, *73*(3), 263–284.

Noble, J. (1988). Finding corners. *Image Vision Computing*, *6*(2), 121–128.

Park, S., Ahmad, M., Seung-Hak, R., Han, S., & Park, J. (2004). Image corner detection using radon transform. In A. Lagano, M. Gavrilova, V. Kumar, Y. Mun, C. Tan, & O. Gervasi (Eds.), *Lecture Notes in Computer Science*: *Vol. 3046*. *Computational science and its applications*, ICCSA 2004, (pp. 948–955). Berlin, Heidelberg: Springer.

Perez-Carrasco, J. A., Zhao, B., Serrano, C., Acha, B., Serrano-Gotarredona, T., Chen, S., & Linares-Barranco, B. (2013). Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate-coding. Application to feed forward convnets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(11), 2706–2719.

Roska, B., & Werblin, F. (2003). Rapid global shifts in natural scenes block spiking in specific ganglion cell types. *Nature Neuroscience*, *6*, 600–608.

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In *European conference on computer vision*, vol. 1 (pp. 430–443).

Theunissen, F., & Miller, J. P. (1995). Temporal encoding in nervous systems: a rigorous definition. *Journal of Computational Neuroscience*, *2*(2), 149–162.

Tuytelaars, T., & Mikolajczyk, K. (2007). Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, *3*(3), 177–280.

van Rossum, M. (2001). A novel spike distance. *Neural Computation*, *13*, 751–763.

Wallach, H. (1935). Uber visuell wahrgenommene bewegungsrichtung. *Psychologische Forschung*, *20*, 325–380.

Wang, H., Klaser, A., Schmid, C., & Liu, C. L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, *103*(1), 60–79.

Weinland, D., Ronfard, R., & Boyer, E. (2011). A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, *115*(2), 224–241.

Whittle, M. (1996). *Gait analysis: an introduction* (2nd Ed.). Oxford: Butterworth–Heinemann.

Wohlgemuth, A. (1935). On the after-effect of seen movement. *British Journal of Psychology*, *1*(Suppl.), 1–117.