An Asynchronous Neuromorphic Event-Driven Visual Part-Based Shape Tracking

David Reverter Valeiras, Xavier Lagorce, Xavier Clady, Chiara Bartolozzi, *Member, IEEE*, Sio-Hoi Ieng, and Ryad Benosman

***Abstract*— Object tracking is an important step in many artificial vision tasks. The current state-of-the-art implementations remain too computationally demanding for the problem to be solved in real time with high dynamics. This paper presents a novel real-time method for visual part-based tracking of complex objects from the output of an asynchronous event-based camera. This paper extends the pictorial structures model introduced by Fischler and Elschlager 40 years ago and introduces a new formulation of the problem, allowing the dynamic processing of visual input in real time at high temporal resolution using a conventional PC. It relies on the concept of representing an object as a set of basic elements linked by springs. These basic elements consist of simple trackers capable of successfully tracking a target with an ellipse-like shape at several kilohertz on a conventional computer. For each incoming event, the method updates the elastic connections established between the trackers and guarantees a desired geometric structure corresponding to the tracked object in real time. This introduces a high temporal elasticity to adapt to projective deformations of the tracked object in the focal plane. The elastic energy of this virtual mechanical system provides a quality criterion for tracking and can be used to determine whether the measured deformations are caused by the perspective projection of the perceived object or by occlusions. Experiments on real-world data show the robustness of the method in the context of dynamic face tracking.**

***Index Terms*— Neuromorphic sensing, part based, pictorial structures, time-encoded imaging, visual tracking.**

I. Introduction

OBJECT tracking is a fundamental problem in many computer vision applications, including video-based surveillance systems [1], [2], human–computer interaction [3], augmented reality [4], or traffic monitoring [5]. Despite this research topic being extensively studied over the last decade, fast object recognition and tracking remains a challenging and computationally expensive problem.

A major application of visual tracking is face detection and tracking. This area has been dominated in the recent years by appearance-based methods [6]. These techniques learn a global representation of the object from a set of training images and have been extensively studied, originating from the pioneering work of Viola and Jones [7], later improved in [8]–[14]. A complete review of the recent advances in the field of face detection can be found in [15].

Appearance-based methods implicitly assume a rigid spatial relation between the parts that constitute the object. Other methods that overcome this limitation rely on modeling an object as a set of simple parts with a flexible geometric relation between them. These techniques, known as part-based methods, have received a lot of attention as they allow the recognition of generic classes of objects and the estimation of their pose. An early example of this technique can be found in [16], in which a model known as pictorial structures is introduced. In the pictorial structures framework, the local elements of the object are assumed to be rigid and linked by springs. A probabilistic model is defined and looks for the best match that minimizes two cost functions: 1) for the matching of individual parts and 2) for the geometric relations between them. The main limitation of this technique, which has prevented its use, is the high computational cost needed to solve the resulting minimization problem. With the increase in available computational power, this method has received renewed attention [17]–[20]. Closely related is the work of Crandall *et al*. [21], which introduced the k-fans model. This is a statistical model in which k defines the degree of connection between the parts, allowing a tradeoff between representational power and computational cost.

More recently, improvements have been made to the pictorial structures technique. For example, Zuffi *et al.* [22] extended the model, introducing what they term deformable structures, allowing nonrigid deformations of the shape of local parts. In addition, Andriluka *et al.* [23] combined the pictorial structures approach with appearance-based methods, allowing them to leverage some of the advantages of both the approaches.

In spite of the recent improvements, these techniques are fundamentally limited by the low dynamics imposed by the frame rates of current cameras. Increasing the frame rate is possible, but current computation techniques do not allow frame rates above the conventional 60–90 Hz, preventing a true dynamic formulation of the problem capable of running at the temporal resolution of the observed scene. Increasing the frame rate requires a huge of amount of calculations [24], [25].

This paper introduces a new approach to the problem, which relies on time-encoded imaging to provide high temporal

Manuscript received April 23, 2014; revised October 20, 2014 and January 29, 2015; accepted February 4, 2015. Date of publication March 18, 2015; date of current version November 16, 2015.

D. Reverter Valeiras, X. Lagorce, X. Clady, S.-H. Ieng, and R. Benosman are with the Institut National de la Santé et de la Recherche Médicale, Paris 75654, France, the Institut de la Vision, University Pierre and Marie Curie, Sorbonne Universités, Paris 75252, France, and also with the Centre National de la Recherche Scientifique, Paris 75794, France (e-mail: david.reverter-valeiras@inserm.fr; xavier.lagorce@inserm.fr; xavier.clady@upmc.fr; sio-hoi.ieng@upmc.fr; ryad.benosman@upmc.fr).

C. Bartolozzi is with the iCub Facility, Istituto Italiano di Tecnologia, Genoa 16163, Italy (e-mail: chiara.bartolozzi@iit.it).

This paper has supplemental material available online at http://ieeexplore.ieee.org (File size: 26.1 Mb).

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.2015.2401834

resolution and sparse output, thus yielding a true dynamic framework to the problem [26], [27]. This allows for a drastic simplification of the method, for a true dynamic formulation and thus is closer to the initial ideas and ethos of the pioneering work on pictorial structures [16]. Time-encoded imaging, with its asynchronous acquisition, allows a data-driven part-based update of the model iteratively for every incoming detected event. Therefore, the resulting algorithm is much simpler and still robust. Instead of explicitly minimizing the energy function, we simply let the system evolve. As we apply the effects of the springs, their elastic energy tends to get smaller, in the same way as it would in a real mechanical system.

Recent work in computer vision has introduced neuromorphic imaging primarily in machine learning [28], [29] but also in the computation of stereovision [30], [31], optical flow [32], [33], and tracking [34]. However, the potential of these cameras allows the consideration of many more applications as they unlock new perspectives in reformulating vision problems.

Despite its promising characteristics, very few object tracking algorithms exploiting the possibilities of event-driven acquisition have been developed so far. An event clustering algorithm is introduced for traffic monitoring, where clusters can change in size but are restricted to a circular form [35], [36]. A fast sensory motor system has been built to demonstrate the sensor's high temporal resolution properties in [37]. In [38], a balancing robot is developed to stabilize a pencil using a fast event-based Hough transform. The sensor has been recently applied to track particles in microrobotics [34] and in fluid mechanics [39]. It was also used to track microgripper's jaws to provide real-time haptic feedback on the micrometer scale (lengths and sizes of objects around 1e-6 m) [40]. In [41], a blob tracker is described which is capable of adapting its shape and position to the distribution of incoming events, assuming that it follows a bivariate Gaussian distribution.

This paper is organized as follows. Section II describes the silicon retina. Section III describes the model and explains how it has been implemented. Section IV presents the experiments carried out to validate the method and discusses some of the results. Finally, Section V presents the conclusions and future perspectives.

II. Time Encoded Imaging

Biomimetic event-based cameras are a novel type of vision device that—like their biological counterparts—are driven by events happening within the scene. Unlike conventional vision sensors, these cameras are not driven by artificially created timing and control signals (e.g., frame clock) which have no relation to the source of the visual information [26]. Over the past few years, a variety of these event-based devices have been designed, including temporal contrast vision sensors that are sensitive to relative illuminance change [26], [27], [42], gradient-based sensors sensitive to static edges [43], edge-orientation sensitive devices, and optical-flow sensors [44], [45]. Most of these vision sensors output visual information about the scene in the form of asynchronous address events (AER) [46] and encode the visual information in the time dimension and not as voltage, charge, or current. The presented pattern tracking method is designed to work on data delivered by such time-encoding sensors and takes full advantage of the high temporal resolution and the sparse data representation. The asynchronous time image sensor (ATIS) used in this paper is a time-domain encoding vision sensor with a 304×240 pixel resolution [27]. The sensor contains an array of fully autonomous pixels that combine an illuminance change detector circuit and a conditional exposure measurement block.

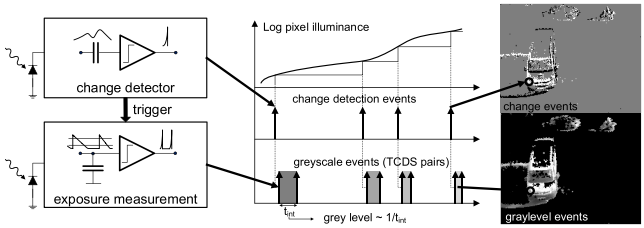


Fig. 1. Functional diagram of an ATIS pixel [42]. Two types of asynchronous events, encoding change and brightness information, are generated and transmitted individually by each pixel in the imaging array.

As shown in the functional diagram of the ATIS pixel in Fig. 1, the change detector individually and asynchronously initiates the measurement of an exposure/gray scale value only if—and immediately after—an illuminance change of a certain magnitude has been detected in the field-of-view of the respective pixel. The exposure measurement circuit in each pixel individually encodes the absolute instantaneous pixel illuminance into the timing of asynchronous event pulses, more precisely into interevent intervals.

Since the ATIS is not clocked like conventional cameras, the timing of events can be conveyed with a very accurate temporal resolution in the order of microseconds. The time-domain encoding of the intensity information automatically optimizes the exposure time separately for each pixel instead of imposing a fixed integration time for the entire array, resulting in an exceptionally high dynamic range and improved signal-to-noise ratio. The individual pixel change detector operation yields almost ideal temporal redundancy suppression, resulting in a sparse encoding of the image data. Frames are absent from this acquisition process. However, they can be reconstructed, when needed, at frequencies limited only by the temporal resolution of the pixel circuits (up to hundreds of kiloframes per second). This paper uses only the change detector events, as the timings of the events is the main information required to perform tracking. Reconstructed images from the sensor have been used for display purposes and during initialization stages.

III. Description of the Method

A. Gaussian Blob Trackers

A stream of visual events can be mathematically defined as follows: let $\text{ev}(\mathbf{u}, t) = [\mathbf{u}, t, \text{pol}]^T$ be a quadruplet giving the pixel positions $\mathbf{u} = [x, y]^T$, t be the time of the event, and pol be its polarity, which can be -1 or 1. When an

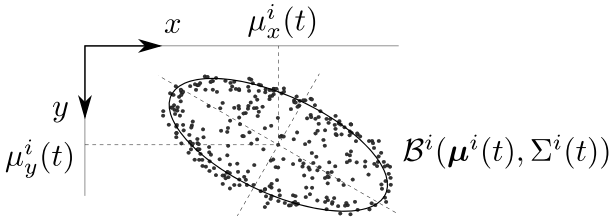


Fig. 2. Gaussian tracker $\mathcal{B}^i$ following a cloud of events is defined by its location $\mu^i(t) = [\mu_x^i(t), \mu_y^i(t)]^T$ and covariance matrix $\Sigma^i(t)$.

object moves, the pixels generate events, which geometrically form a point cloud that represents the spatial distribution of the observed shape.

The event-based Gaussian tracker developed in [41] assumes that these events are normally distributed. According to this assumption, the event cloud approximates a bivariate Gaussian distribution, whose parameters can be iteratively corrected with the incoming events. Thus, the Gaussian tracker is driven by the asynchronous events, causing it to move and to deform so as to approximate the event cloud's spatial distribution.

Let $\mathcal{B}^i(\boldsymbol{\mu^i}, \Sigma^i)$ be the Gaussian tracker shown in Fig. 2. The Gaussian tracker is then defined by its mean $\boldsymbol{\mu}^i(t) = [\mu_x^i(t), \mu_y^i(t)]^T$ that represents the object's position and its covariance matrix $\Sigma^i(t) \in \mathbb{R}^{2\times 2}$ that is used to compute its size and orientation and has the form

$$\Sigma(t) = \begin{bmatrix} \sigma_x^2(t) & \sigma_{xy}(t) \\ \sigma_{xy}(t) & \sigma_y^2(t) \end{bmatrix}. \tag{1}$$

In what follows, we will refer to the tracker as $\mathcal{B}^i$, for notational clarity. For the same reason, we will often drop the time dependence (t).

To illustrate the update procedure, let us assume that several Gaussian trackers have already been initialized. For each incoming event, the probability of it belonging to the ith Gaussian tracker $\mathcal{B}^i$ is given by

$$p^i(\boldsymbol{u}) = \frac{1}{2\pi}|\Sigma^i|^{-\frac{1}{2}} e^{-\frac{1}{2}(\boldsymbol{u}-\boldsymbol{\mu}^i)^T(\Sigma^i)^{-1}(\boldsymbol{u}-\boldsymbol{\mu}^i)} \tag{2}$$

where $\boldsymbol{u} = [x, y]^T$ is the pixel location of the event.

The event will then be assigned to the tracker with the highest Gaussian probability, provided that this probability is greater than a predefined threshold, $p^i(\boldsymbol{u}) > \delta p$ (usually set to 0.1). Once the most probable tracker has been identified, its parameters are updated by integrating the last distribution with the current event information, using a simple weighting strategy, as described in (3) and (4). Since only the chosen tracker is examined hereafter, the subscript i indicating the tracker number is omitted for clarity

$$\mu(t) = \alpha_1 \mu(t - \Delta t) + (1 - \alpha_1)\mathbf{u} \tag{3}$$

$$\Sigma(t) = \alpha_2 \Sigma(t - \Delta t) + (1 - \alpha_2)\Delta\Sigma \tag{4}$$

where Δt is the time difference between current and previous events, and α_1 and α_2 are update factors and should be tuned according to the event rate.

The covariance difference $\Delta\Sigma$ can be computed using the current tracker's location $\boldsymbol{\mu(t)} = [\mu_x(t), \mu_y(t)]^T$ and event's location $\boldsymbol{u}$

$$\Delta\Sigma = \begin{bmatrix} (x - \mu_x(t))^2 & (x - \mu_x(t))(y - \mu_y(t)) \\ (x - \mu_x(t))(y - \mu_y(t)) & (y - \mu_y(t))^2 \end{bmatrix}. \tag{5}$$

Finally, the activity $\mathcal{A}^i$ of each tracker $\mathcal{B}^i$ is updated at each incoming event $\text{ev}(\boldsymbol{u}, t)$, following an exponential decay function, which describes the temporal dimension of the Gaussian kernel:

$$\mathcal{A}^i(t) = \begin{cases} \mathcal{A}^i(t - \Delta t)e^{-\frac{\Delta t}{\tau}} + p^i(\boldsymbol{u}), \\ \qquad\qquad \text{if } ev(\boldsymbol{u}, t) \text{ belongs to tracker } i \\ \mathcal{A}^i(t - \Delta t)e^{-\frac{\Delta t}{\tau}}, \quad \text{otherwise} \end{cases} \tag{6}$$

where τ is a factor tuning the temporal activity decrease.

If the activity $\mathcal{A}^i$ of a tracker $\mathcal{B}^i$ is greater than a predefined threshold $\mathcal{A}_{\text{up}}$, then $\mathcal{B}^i$ is said to be active. Thus, a tracker is said to be active when it is correctly following a cloud of events.

B. Spring-Linked Feature Trackers

The Gaussian tracker introduced in Section III-A produces robust results when dealing with simple objects, especially in the case of ellipse-like shapes. Unlike other blob tracking algorithms, it is capable of estimating not only the main position of an object, but also its size and orientation. When attempting to track a more complex object, we can assume that it is composed of a set of simple shapes linked by geometric relations. These relations, however, cannot be fixed, as the movement of the object in the 3-D space will cause its projection onto the focal plane to be modified.

In order to build a tracker capable of following the structure of observed objects, we model our system as a set of simple trackers linked by springs. Thus, each tracker of the set will be driven both by the incoming events and by the elastic connections linking it to other elements.

1) Euclidean Configuration: According to the well known Hooke's law, the force F_k required to displace a linear ideal spring from its equilibrium position is given by

$$F_k = -k\,\Delta l \tag{7}$$

where Δl represents the elongation of the spring, which is the difference between its current length and the equilibrium length, and k is a characteristic of the spring, known as its stiffness.

Fig. 3(a) shows a linear spring to which an object of mass m has been attached. An energy dissipation mechanism is also added and is represented in this case by an ideal damper. The frictional force F_t applied by the damper is modeled as being proportional and opposed to the velocity v between its opposite sides

$$F_t = -cv = -c\frac{d\,\Delta l}{dt} \tag{8}$$

where c is known as the viscous damping coefficient of the damper.

Fig. 3(b) shows the system out of its equilibrium position and with a certain speed. In such a case, two forces F_k and F_t

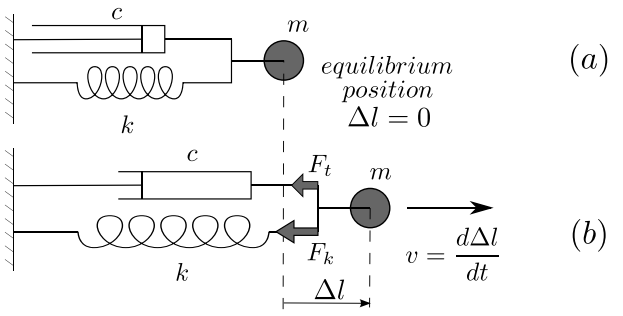


Fig. 3. Principle of a damped spring. (a) Mass m is attached to a mechanical system composed of a linear spring and a linear damper. (b) When the system is out of its equilibrium position, two forces F_k (elongation) and F_t (frictional) appear. Their directions are opposed, respectively, to those of the displacement and the velocity.

appear, their directions being opposed to those of the displacement and the velocity, respectively.

Applying Newton's second law, we obtain the differential equation of the system, needed to calculate the object's acceleration, velocity, and position

$$m\frac{d^2\Delta l}{dt} = -k\Delta l - c\frac{d\Delta l}{dt}. \tag{9}$$

This is a typical problem in classical mechanics, and its solutions are well known and studied [47], [48].

If a series of connections is set between the different trackers, assigning masses to these trackers allows us to actually model the behavior of this virtual dynamic system. Even if we are not modeling a real system, keeping the concept of masses for the trackers allows us to control their relative displacements, assigning bigger masses to the elements that we wish to be more stable.

Let $\mathcal{C}^{ij}$ be a connection bounding the Gaussian trackers $\mathcal{B}^i$ and $\mathcal{B}^j$. As shown in Fig. 2, the center of the trackers is represented by $\boldsymbol{\mu} = [\mu_x, \mu_y]^T$. Fig. 4(a) shows this connection in its equilibrium state, where l_0^{ij} represents the equilibrium distance and θ^{ij} the angle formed by the axis of the connection and the horizontal axis. In what follows, we will use a simplified representation of the connection, showing only the spring.

Let us assume that this connection behaves as a single linear spring that can freely rotate around its ends, where it is connected to the respective trackers. When modeling the system this way, we are only taking into account the Euclidean distance between the trackers, and not at all the direction of the connection. From now on, we will refer to this configuration as the Euclidean configuration.

Fig. 4(b) and (c) illustrates the evolution of the trackers from equilibrium, as they are driven by both the incoming events and the spring-like connections. As the events start arriving, they will cause the trackers to move away from their initial positions, eventually causing a certain elongation of the spring. Fig. 4(b) shows the state of the system after the trackers have been displaced by the events, where l^{ij} represents the current distance between the trackers and Δl^{ij} the corresponding elongation, given by

$$\Delta l^{ij} = \left(l^{ij} - l_0^{ij}\right). \tag{10}$$

As a consequence of this elongation, the trackers will then be driven by the spring-like connection, which tries to recover its initial equilibrium length. Fig. 4(c) shows the corresponding displacement of the trackers $\boldsymbol{\Delta\mu} = [\Delta\mu_x, \Delta\mu_y]^T$, which are always in the opposite direction to the elongation.

For computing these displacements, we will simply assume that they are proportional to the elongation. This approximation, causing an exponential decay toward equilibrium, is valid under the conditions described in the Appendix, and will result in the following values for the displacements:

$$\boldsymbol{\Delta\mu^i} = \begin{pmatrix}\Delta\mu_x^i \\ \Delta\mu_y^i\end{pmatrix} = \frac{\alpha^{ij}}{m^i}\Delta l^{ij}\begin{pmatrix}\cos(\theta^{ij}) \\ \sin(\theta^{ij})\end{pmatrix}$$

$$\boldsymbol{\Delta\mu^j} = \begin{pmatrix}\Delta\mu_x^j \\ \Delta\mu_y^j\end{pmatrix} = -\frac{\alpha^{ij}}{m^j}\Delta l^{ij}\begin{pmatrix}\cos(\theta^{ij}) \\ \sin(\theta^{ij})\end{pmatrix} \tag{11}$$

where α^{ij} is a scaling factor that controls the stiffness of the connection, and m^i and m^j represent the masses associated with the ith and jth trackers, respectively. θ_{ij} is the angle between the axis of the connection and the horizontal line.

Once we have defined our set of connections between the trackers, it is interesting to compute the elastic energy of the system. The elastic energy is the potential mechanical energy stored by a material or a physical system as a result of its deformation. Therefore, its value gives us a measure of the state of deformation of the system. In the next sections, we will discuss its possible use as a criterion to decide whether a cloud of events corresponds to the desired object or not.

For a single connection $\mathcal{C}^{ij}$, linked by a linear spring, its elastic energy E^{ij} is given by

$$E^{ij} = \frac{1}{2}\alpha^{ij}(\Delta l^{ij})^2. \tag{12}$$

In addition, for computing the elastic energy of the system E, we simply apply the superposition principle

$$E = \sum E^{ij}. \tag{13}$$

When computed this way, we will refer to it as the Euclidean elastic energy.

2) Torsional Configuration: If we want to keep the angle of each connection close to the equilibrium value, we can imagine the trackers as being linked by torsion springs. The force applied by a torsion spring is proportional to the difference between the current angle and the equilibrium angle. Adding an ideal prismatic joint between the trackers allows us to avoid taking into account the distance between them. The equivalent mechanical system can be seen in Fig. 5, where θ_0^{ij} represents the initial equilibrium angle of the connection and θ^{ij} represents its current value.

We will refer to this configuration as the torsional configuration. In this case, the torsional elongation is given by

$$\widetilde{\Delta\theta}^{ij} = \theta^{ij} - \theta_0^{ij} \tag{14}$$

which can be corrected by subtracting its mean value along the connections, in order to make the system insensitive to rotation

$$\Delta\theta^{ij} = \widetilde{\Delta\theta}^{ij} - \overline{\Delta\theta} \tag{15}$$

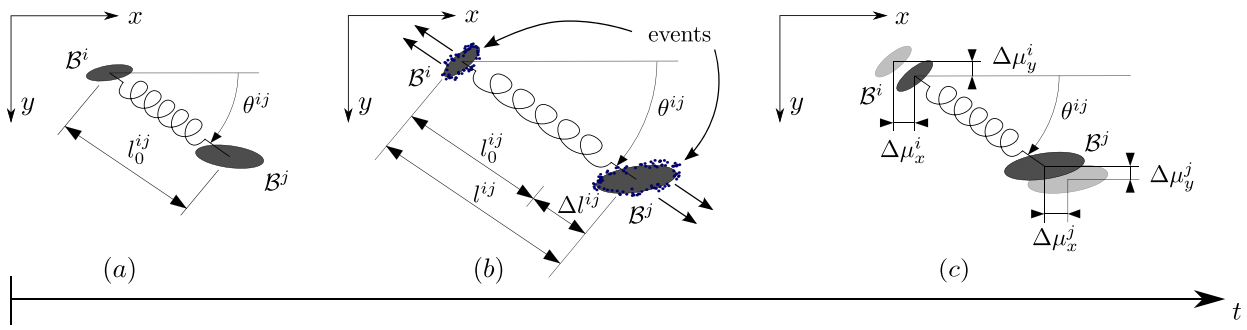


Fig. 4. (a) Connection $\mathcal{C}^{ij}$ (that links the Gaussian trackers $\mathcal{B}^i$ and $\mathcal{B}^j$) in its initial equilibrium state, where l_0^{ij} represents the initial length of the spring, and θ^{ij} represents the angle formed by the axis of the connection and the horizontal axis. (b) Trackers follow incoming events, moving the connection away from its equilibrium position. The difference between the current length of the connection l^{ij} and the initial length is known as the elongation of the spring Δl^{ij}. (c) Trackers are driven by the spring-like connection that tries to recover its initial length.

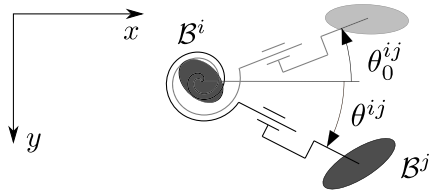


Fig. 5. Torsional configuration. Trackers are linked by torsion springs. The equilibrium angle is initially θ_0^{ij}, while θ^{ij} is its position after torsion.

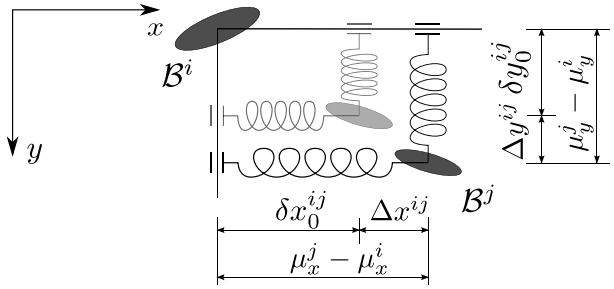


Fig. 6. Cartesian configuration: the initial distances between the trackers are equal to δx_0^{ij} and δy_0^{ij}, which correspond to the horizontal and vertical equilibrium distances of the connection. The tracker activity following events originates an elongation given by Δx^{ij} and Δy^{ij} causing a change of distance between trackers i and j.

where $\overline{\Delta\theta}$ is the mean torsional elongation. Thus, if all the connections rotate through the same angle in the same direction, the torsional elongation will be zero for all of them, making the system insensitive to rotation.

Next, from the value of the elongation, we compute the corresponding displacements to be applied to the trackers. The tracker's displacement is equivalent to a change in the force exerted by the spring. As in the case of the Cartesian configuration, we simplify the effect of the spring using a first-order approximation, thus the displacement is inducing a linear change in θ_{ij}. If the current relative position is $l^{ij}[\cos(\theta^{ij}), \sin(\theta^{ij})]^T$, the new angle will be $\theta^{ij} + \alpha^{ij}\Delta\theta^{ij}$ and the new relative position $l^{ij}[\cos(\theta^{ij} + \alpha^{ij}\Delta\theta^{ij}), \sin(\theta^{ij} + \alpha^{ij}\Delta\theta^{ij})]^T$. From here, the displacements to be applied to the trackers are equal to

$$\begin{aligned}\boldsymbol{\Delta\mu^i} &= \frac{l^{ij}}{m^i}\begin{pmatrix}\cos(\theta^{ij} + \alpha^{ij}\Delta\theta^{ij}) - \cos(\theta^{ij})\\ \sin(\theta^{ij} + \alpha^{ij}\Delta\theta^{ij}) - \sin(\theta^{ij})\end{pmatrix}\\ \boldsymbol{\Delta\mu^j} &= -\frac{l^{ij}}{m^j}\begin{pmatrix}\cos(\theta^{ij} + \alpha^{ij}\Delta\theta^{ij}) - \cos(\theta^{ij})\\ \sin(\theta^{ij} + \alpha^{ij}\Delta\theta^{ij}) - \sin(\theta^{ij})\end{pmatrix}.\end{aligned} \quad (16)$$

The elastic energy associated with this configuration will be given by

$$E^{ij} = \frac{1}{2}\alpha^{ij}(\Delta\theta^{ij})^2. \quad (17)$$

When computed this way, we will refer to it as the torsional elastic energy.

3) Cartesian Configuration: If we want to keep both the distance and the angle of each connection close to those of the equilibrium, we can set a horizontal and a vertical equilibrium distances. The equivalent mechanical system is represented in Fig. 6, where δx_{ij}^0 and δy_{ij}^0 are the horizontal and vertical equilibrium distances, respectively. Δx_{ij} and Δy_{ij} represent the horizontal and vertical elongations, given by

$$\begin{aligned}\Delta x^{ij} &= \left(\mu_x^j - \mu_x^i - \delta x_0^{ij}\right)\\ \Delta y^{ij} &= \left(\mu_y^j - \mu_y^i - \delta y_0^{ij}\right).\end{aligned} \quad (18)$$

When modeling the system this way, the computation becomes very simple. Keeping the same simplifications as in the previous cases, the displacements to be applied to the trackers are given by

$$\begin{aligned}\boldsymbol{\Delta\mu^i} &= \frac{\alpha^{ij}}{m^i}\begin{pmatrix}\Delta x^{ij}\\ \Delta y^{ij}\end{pmatrix}\\ \boldsymbol{\Delta\mu^j} &= -\frac{\alpha^{ij}}{m^j}\begin{pmatrix}\Delta x^{ij}\\ \Delta y^{ij}\end{pmatrix}.\end{aligned} \quad (19)$$

We will refer to this configuration as the Cartesian configuration. The elastic energy associated to this configuration will be given by

$$E^{ij} = \frac{1}{2}\alpha^{ij}((\Delta x^{ij})^2 + (\Delta y^{ij})^2). \quad (20)$$

When computed this way, we will refer to it as the Cartesian elastic energy.

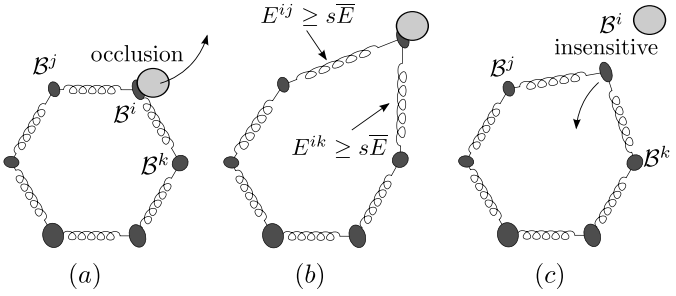


Fig. 7. (a) System is correctly tracking an object until an occlusion occurs. (b) If this occlusion attracts only one tracker $\mathcal{B}^i$, the energy of every connection linking this tracker will grow to be higher than the rest. (c) If the elastic energy of every connection linking the tracker $\mathcal{B}^i$ gets higher than a threshold (defined as proportional to the mean elastic energy), then the tracker becomes insensitive. This means that no event can be assigned to the tracker. Consequently, it will be driven exclusively by its connections and quickly recover its equilibrium position relative to its neighbors.

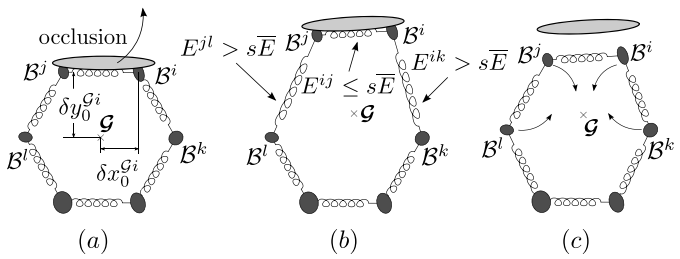


Fig. 8. (a) System is correctly tracking an object until an occlusion occurs. In this case, we suppose the occlusion to attract two trackers $\mathcal{B}_i$ and $\mathcal{B}^j$. (b) In this case, E^{ij} remains stable, and the previous mechanism will not be activated. However, the energy of the rest of connections linking these trackers will increase to be higher than the energy threshold. (c) If the elastic energy E^{ik} of any connection $\mathcal{C}^{ik}$ gets higher than a threshold (defined as proportional to the mean elastic energy), then both of the trackers $\mathcal{B}^i$ and $\mathcal{B}^k$ linked by the connection are attracted toward their equilibrium position, relative to the center of mass of the set of feature trackers.

C. Using the Energy as a Matching Criterion

As previously explained, the elastic energy is a measure of the deformation of a connection. We will assume that, when correctly tracking the desired object, the energy of all of the connections will remain relatively stable. On the contrary, if the energy of a connection becomes much higher than the rest, it is likely that one or both of the trackers linked by this connection have lost track of the desired feature, following a cloud of events that does not correspond to the tracked object. This will typically happen in the case of partial occlusions. Next, two energy-based mechanisms are presented that increase the robustness to partial occlusions.

1) Preventing a Single Tracker From Following the Wrong Cloud of Events: Let us imagine that a set of feature trackers is correctly tracking an object. Fig. 7(a) shows the state of the system in such a situation. As a certain degree of deformation is acceptable, the energy of its connections will typically be different from zero. However, we will assume them to be relatively stable and similar to each other as long as the system is correctly tracking the desired object. Next, let us imagine that a partial occlusion occurs, generating a cloud of events that does not correspond to the tracked object. In a first step, let us imagine that a single tracker $\mathcal{B}_i$ starts following this wrong cloud of events, while the rest of the trackers are unaffected. Fig. 7(b) shows the resulting situation; in this case, the energy of all the connections $\mathcal{C}_{ij}$ bounding $\mathcal{B}_i$ will grow to be much higher than the rest. Thus, when the energy of all the connections bounding a certain tracker is higher than a threshold, we will make this tracker stop following events. As we wish to allow stable growth of the elastic energy, this threshold will be defined as proportional to the mean elastic energy of all the connections. The criterion will therefore be expressed by

$$\text{if } E^{ij} \geq s\overline{E} \quad \forall j \text{ so that } \mathcal{C}^{ij} \text{ exists} \Rightarrow \mathcal{B}^i \text{ insensitive} \tag{21}$$

where $\overline{E}$ represents the mean elastic energy of all the connections, and s is a positive scaling factor. An insensitive tracker cannot have events assigned to it. As this condition is evaluated for every incoming event, the tracker will be insensitive until the energy of one of its connections drops below the threshold.

As a consequence of the tracker being insensitive to the incoming events, it will exclusively be driven by the spring-like connections. This will cause it to quickly recover its equilibrium position relative to its neighbors, typically finding the desired object again.

2) Preventing a Group of Trackers From Following the Wrong Cloud of Events: The second mechanism is designed to avoid a group of trackers following the wrong cloud of events. Fig. 8(a) shows the same system as in the previous case, correctly tracking the desired object. In this case, however, the cloud of events generated by the partial occlusion will attract two trackers $\mathcal{B}^i$ and $\mathcal{B}^j$. As we can see in Fig. 8(b), the previous mechanism will not be activated by this situation, as the energy of the connection bounding $\mathcal{B}^i$ and $\mathcal{B}^j$ remains stable.

Let $\boldsymbol{\mathcal{G}} = [\mathcal{G}_x,\ \mathcal{G}_y]^T$ be the coordinates of the center of mass of the set of feature trackers, and let $\delta x_0^{\mathcal{G}i}$ and $\delta y_0^{\mathcal{G}i}$ be the horizontal and vertical distances, respectively, from this point to a generic tracker $\mathcal{B}^i$ when the system is at equilibrium [Fig. 8(a)]. If the energy E^{ij} of a connection $\mathcal{C}^{ij}$ is higher than a certain multiple of the mean elastic energy, then we will displace both trackers $\mathcal{B}^i$ and $\mathcal{B}^j$ linked by the connection toward their equilibrium position, relative to the current position of the center of mass. In the same way as for the spring-like connections, the displacements applied to the trackers will be proportional to the distance to the equilibrium position (relative, in this case, to the center of mass)

$$\text{if } E^{ij} \geq s\overline{E} \Rightarrow \begin{cases} \boldsymbol{\Delta\mu^i} = \alpha_{\text{en}} \begin{pmatrix} \mathcal{G}_x + \delta x_0^{\mathcal{G}i} - \mu_x^i \\ \mathcal{G}_y + \delta y_0^{\mathcal{G}i} - \mu_y^i \end{pmatrix} \\ \boldsymbol{\Delta\mu^j} = \alpha_{\text{en}} \begin{pmatrix} \mathcal{G}_x + \delta x_0^{\mathcal{G}j} - \mu_x^j \\ \mathcal{G}_y + \delta y_0^{\mathcal{G}j} - \mu_y^j \end{pmatrix} \end{cases} \tag{22}$$

where α_{en} is the proportionality factor, equivalent to the stiffness of our spring-like connections. This mechanism is in fact quite similar to that of the spring-like connections. However, there is a fundamental difference: it is just applied when the connections surpass the energy threshold, and its equilibrium distance is defined relatively to the center of mass.

Algorithm 1 Global Algorithm

for every incoming event **do**
  Update the activity $\mathcal{A}^i$ of every tracker using (6).
  Update the best candidate sensitive tracker's position and size using (3) and (4).
  for every connection $\mathcal{C}^{ij}$ **do**
    Compute the displacements $\mathbf{\Delta\mu^i}$ and $\mathbf{\Delta\mu^j}$ of the trackers using (11), (16) or (26).
    Compute the Elastic Energy E^{ij} of the connection using (12), (17) or (20).
  end for
  Compute the mean Elastic Energy $\overline{E}$.
  for every connection $\mathcal{C}^{ij}$ **do**
    if $E^{ij} \geq s\overline{E}$ **then**
      displace $\mathcal{B}^i$ and $\mathcal{B}^j$ using (22).
    end if
  end for
  for every tracker $\mathcal{B}^i$ **do**
    if $E^{ij} \geq s\overline{E}$ for every $\mathcal{C}^{ij}$ bounding $\mathcal{B}^i$ **then**
      $\mathcal{B}^i$ is insensitive.
    end if
  end for
end for

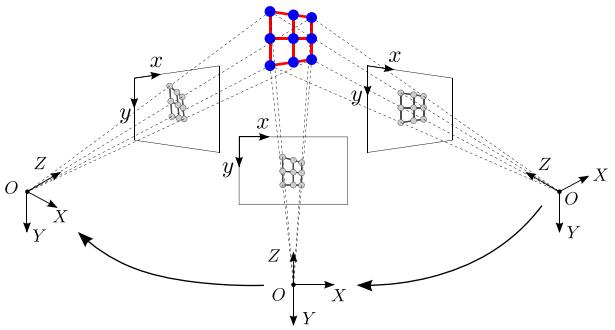


Fig. 9. Neuromorphic sensor observes a moving 3×3 grid of fixed sized blobs. Distortions are applied to the grid to simulate a free evolution in a 3-D space.

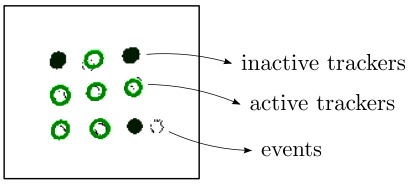


Fig. 10. Snapshot created from the output of the neuromorphic camera showing the state of the trackers for a configuration of the moving grid. The snapshot shows events happening over a time period of 10 ms. Green ellipses: active trackers. Solid black ellipses: inactive trackers.

The parameter s has a strong impact on the behavior of the system, and it should be carefully chosen. A detailed study of its impact will be discussed in the next section.

D. Remarks

When building the model of an object, we are not constrained to a unique configuration. Instead, we can imagine any combination of the three of them and assign them different stiffnesses, in order to obtain the desired behavior.

However, we need to be careful when applying the mechanisms described in Section III-C2. As explained in the Appendix, the stiffness α of our spring-like connections, as well as the masses of the trackers, are nothing but dimensionless scaling factors. This means that, when computing the elastic energy of a connection using (12), (17), or (20), the results we are obtaining do not have dimensions of energy. Instead, they are simply a weighted sum of the square of the elongations of each connection. The units of these elongations are pixels for the Cartesian and Euclidean configurations, and radians for the torsional configuration. As a result, making a comparison between these different types of energy does not have any physical interpretation. Consequently, we will be careful to only compare the same types of energy.

E. Global Algorithm

See Algorithm 1 for the global algorithm.

IV. Experiments

All experiments used an ATIS sensor, which sends events to a conventional PC via a USB link. The tracking method is implemented in C++ and runs in real time. The amount of incoming events outputted by the neuromorphic camera can be large, depending on the dynamics of the scene. In the case of the experiment described in Section IV-B, for example, it is equal to approximately 300 000 events/s. In order to perform the tracking in real time, the update of the state of the springs cannot be computed for every incoming event. Instead, we update the set of trackers for a fixed number of events. The choice of this number of events is a compromise between the accuracy of the system and the computational time and was set experimentally to 55 events. This corresponds to an update approximately every 180 μs, which is a very high updating rate—far higher than the usual 16–10 ms of conventional cameras.

A. Tracking a Planar Grid

The neuromorphic sensor observes a computer screen that displays a moving 3×3 grid of fixed sized blobs. Distortions are applied to the grid to simulate a free evolution in a 3-D space (Fig. 9). The position of the springs is represented by the grid's edges, connecting two neighboring circles together. Fig. 10 shows a sample of the recorded stimulus by the neuromorphic camera. The active trackers (green ellipses) are being deformed by incoming events generated by the moving stimulus. The stimulus is updated every 10 ms (100 frames/s) allowed by the technology of the used screen. The inactive trackers are represented as solid black ellipses. The update factors from (3) and (4) were set to $\alpha_1 = 0.02$ and $\alpha_2 = 5 \times 10^{-5}$.

In the first experiment, the grid is displayed while alternating a rotation around the vertical axis with a range of $-75°$ and $75°$ at a constant speed of $15°/s$. A vertical movement has been added in order to generate events in

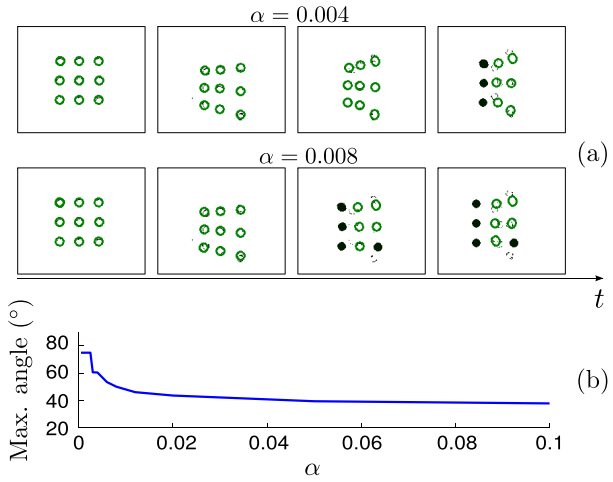


Fig. 11. (a) Results of the experiment for two different values of the stiffness $\alpha = 0.004$ and $\alpha = 0.008$. (b) Maximum tracking angle as a function of α.

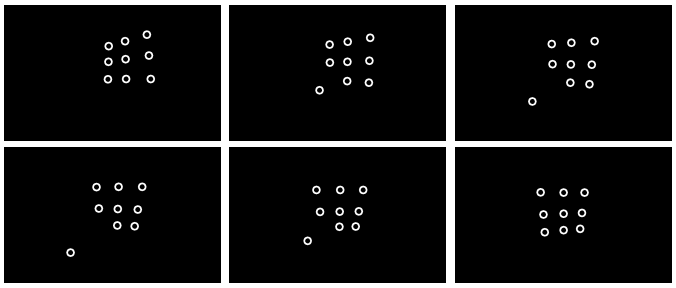


Fig. 12. Snapshots of the stimulus video. As the grid keeps following projective transformations, one of its blobs is taken apart.

all directions. Cartesian connections are set between the trackers, and the effect of their stiffness α is tested.

Fig. 11(a) shows the state of the system at four temporal locations, for two different values of the stiffness α. When $\alpha = 0.004$, the set of trackers is capable of following the grid up to a large angle until a subset of trackers (shown as black ellipses) are unable to follow the target.

Fig. 11(b) shows the maximum tracking angle as a function of the chosen stiffness. It is, as expected, a decreasing relation; when the stiffness of the connections is increased, the trackers cease the tracking at a much earlier stage. Choosing the right α requires a tradeoff between adaptability to the distortions of the scene and robustness to disturbances.

In the second experiment, the grid is shown rotating around the x-axis and y-axis simultaneously, at 15°/s and 10°/s, respectively. The boundary values of each angle are set to $-30°$ and 30° for the x-axis, and $-45°$ and 45° for the y-axis. We added vertical and horizontal motions as in the previous case. We also added random disturbances to the stimulus. An element of the grid is artificially moved away to test the reaction of the system to deformable objects. This also allows the simulation of occlusions that usually cause a subset of the trackers to follow the wrong cloud of events. Fig. 12 shows six snapshots of the stimulus, in which an element of the grid is separated and then returns to its position according to the applied motion.

The ground truth is computed from the positions of the points on the screen, and a homography is estimated between the screen and the focal plane of the neuromorphic camera. The error of the element of the grid simulating an occlusion is computed by comparing its current position to its desired one (the one it would have been located at if the artificial elongation was not applied).

Fig. 13(a) shows the tracking results of a single tracker, for $\alpha = 0.001$ and $s = 4.5$. Fig. 13(b) shows the moment at which one element of the grid starts being pulled from the rest. At the beginning, the tracker successfully follows the element [from Fig. 13(b) to (d)]. As the element keeps moving further away from the rest of the grid, the energy of its connections increases, causing the mechanisms defined in Section III-C to activate. As a result, the tracker will stop following this cloud of events, as shown in Fig. 13(e). Once it is insensitive, it quickly recovers its relative position to its neighbors. As shown in Fig. 13(g), the equilibrium position to which the point returns to will be close to the real position of the element in the scene. For each time t, we define the error as the mean error of all the trackers. We will characterize the system by the temporal mean of this error, expressed as a percentage of the length of the object.

Fig. 14 shows the evolution of the mean error ($\overline{e}$) for several values of s that represent the scaling factor of the mean elastic energy, for a fixed value of $\alpha = 0.001$. The best value obtained is $\overline{e} = 2.74\%$ for $s = 4.5$. The error is stable for $s \geq 12$, because above this value, the energy threshold gets so high that no connection overpasses it. As a result, the mechanisms defined in Section III-C are never activated. This figure shows the positive effect of the energy criteria on the performance of the system. This shows that one can use small values for the stiffness, allowing the system to be robust and thus efficiently tracking the moving target.

We repeated the same experiment for different values of the stiffness. Table I shows the minimum error for each tested value of α when the Cartesian configuration is selected and the corresponding value of s at which it is obtained. The optimal values are obtained for $\alpha = 0.001$ and $s = 4.5$.

To compare the tracking performance of the different spring configurations, we repeated the same experiment connecting neighboring trackers by both an Euclidean and a torsional spring, imposing α to be the same for both of them. Table I shows the tracking error for this configuration. In this case, the optimal values are experimentally obtained for $\alpha = 0.001$ and $s = 4.0$.

Fig. 15 shows the tracking error for both the types of connections. We can see that the Cartesian configuration slightly outperforms the combination of torsional and Euclidean springs for every value of the stiffness.

Another experiment is carried out to test the reaction of the system to rotations of the tracked object on the image plane. The grid rotates around the normal to the screen between 45° and $-45°$ and translates in the focal plane. The tracking errors are computed for both Cartesian connections and a combination of torsional and Euclidean connections. Fig. 16 shows the tracking error with respect to the spring parameters for the rotating grid. In such case, we can observe

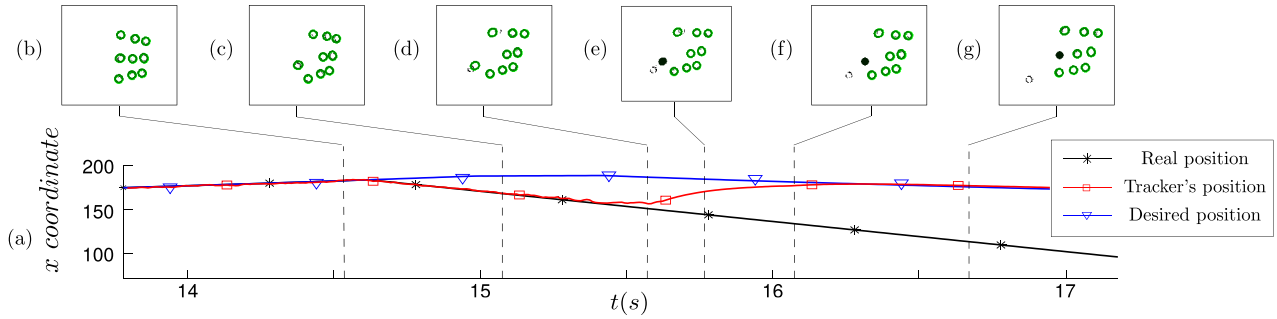


Fig. 13. (a) Position in x of a blob of the grid. The real position is known from the scene generation process. Second line: position of the tracker following the blob. Finally, the desired position represents the position where the tracker should be if the grid was not deformed (e.g., pulled away from the grid structure). As we can observe, the tracker initially follows the blob in its movement away from the grid. However, once the deformation goes beyond a predefined experimental threshold, the tracker becomes inactive, recovering its equilibrium position. (b) Nine trackers are correctly tracking the grid, before the deformation starts. (c) Blob of the grid is pulled away, attracting a tracker. (d) Tracker follows the blob, while the rest maintain their correct targeting blobs. (e) Energy mechanism starts acting, the tracker becomes insensitive to the incoming events. (f) Tracker is labeled as inactive and it quickly recovers its equilibrium position. (g) Tracker is inactive and is in its equilibrium position.

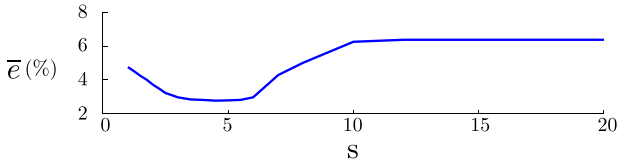


Fig. 14. Evolution of the temporal mean of the error with the value of s. This parameter is a proportionality factor that sets the value of the threshold for the energy-based criteria defined in Section III-C to be activated. When s is too large, these criteria are never activated leading to high tracking errors.

TABLE I
MINIMUM ERROR WITH RESPECT TO STIFFNESS

	Cartesian configuration		*Euclidean + Torsional*	
$\alpha \times 10^{-5}$	min($\overline{e}$)	s	min($\overline{e}$)	s
50	2.93	3.5	3.01	3.0
100	2.74	4.5	2.79	4.0
150	2.84	4.5	2.88	4.5
200	3.36	4.5	3.75	3.5
400	4.30	6.0	4.84	3.5
800	4.79	6.0	5.46	3.0

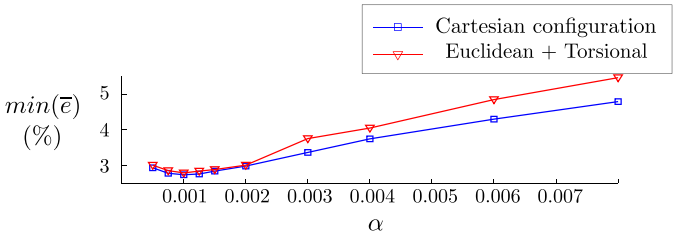


Fig. 15. Minimum tracking error obtained as a function of stiffness, when neighboring trackers are linked by a Cartesian connection or by a combination of a torsional and a Euclidean connection. In this case, the grid experiments general deformation and motion, and the Cartesian configuration slightly outperforms the combination of a torsional connection and a Euclidean connection.

how the Euclidean + torsional configuration is clearly more suited: the error is steady and much lower than in the case of the Cartesian configuration, for which it increases with α.

This allows us to conclude that the Cartesian configuration guarantees more robustness in keeping the desired shape.

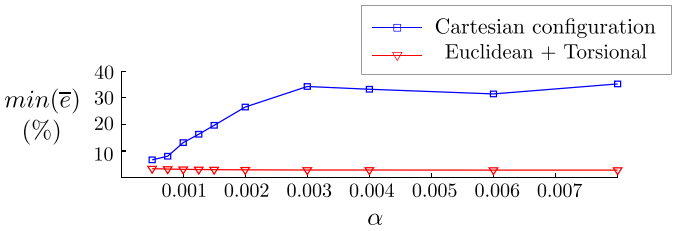


Fig. 16. Minimum tracking error obtained as a function of the stiffness, when neighboring trackers are linked by a Cartesian connection or by a combination of a torsional and a Euclidean connection. In this case, the grid experiments a rotation on the image plane and the combination of torsional and Euclidean connections clearly outperforms the Cartesian configuration.

However, the system becomes sensitive to rotation on the image plane.

B. Face Tracking

The second experiment tests the tracking technique on a real human face in an indoor environment. The target is a moving face, as shown in Fig. 17. The motion of the face is subject to complex dynamics, including phase of steep acceleration changes and scale variations as the target is waving and moving toward the camera.

To characterize this complex sequence, several measurements are defined and measured during the sequence: 1) d is the distance between the two eyes; 2) L is the distance of the eyes to the mouth; and 3) ϕ is the angle between the vertical axis and the central axis of the face (usually known as the roll head angle). Fig. 17 (left) shows the measured values of each parameter for the entire sequence. The number of events for a binning of 10 ms is also shown in Fig. 17(e). The labels A, B, C, and D shown in the Fig. 17 outline interesting time intervals of the experiment.

A moving face was recorded by the ATIS camera. A snapshot from the gray-level output is used to initialize a generic set of connected trackers corresponding to a mean face model. The initial size and orientation of the Gaussian trackers are chosen by modifying their covariance matrix, to fit with the elements of the face and are set for a frontal position located roughly at a distance of a human

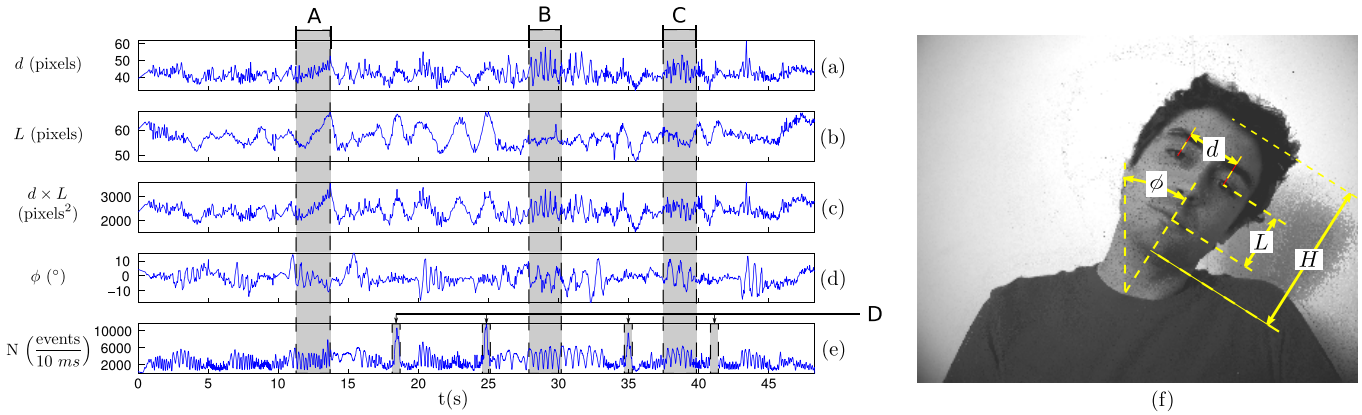


Fig. 17. Geometric parameters observed during the sequence. (a) and (b) d and L are the distances (in pixels) between the two eyes and between the eyes and the mouth. (c) $L \times d$ represents roughly the area of the face in the image. Its value increases when the face gets closer to the camera: the section tagged as A, where both d and L increase, shows such a typical case. On the other hand, changes in d for a constant value of L (or vice versa) correspond to rotations of the face around the x-axis or y-axis. This is represented by the section tagged as B. Rapid oscillations around the yaw axis of the head is producing such results. (d) ϕ is the angle (in degrees) between the vertical axis and the central axis of the face (or the roll angle of the head). (e) Number of incoming events is directly related to the dynamics of the scene. The number of events is shown on a time scale of 10 ms. The temporal locations of high number of events shown by D are the result of occlusions generated when moving hands cover partially the face (Fig. 19). (f) Snapshot extracted from the grayscale output of the ATIS camera, illustrating these parameters.

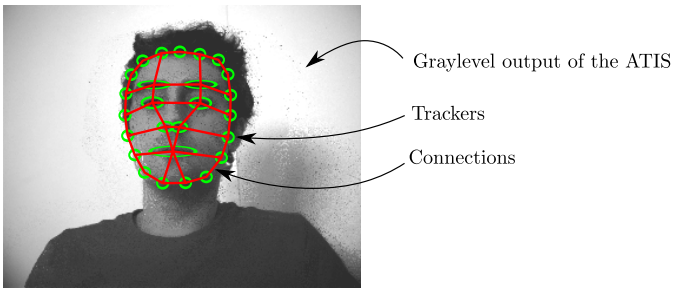


Fig. 18. Set of trackers and the structure of their connections used to follow a face from incoming events. Ellipses: position of the trackers. Lines: connections set between the trackers. Each connection is a combination of a Euclidean connection and a torsional connection, with $\alpha = 0.02$ for both the connections.

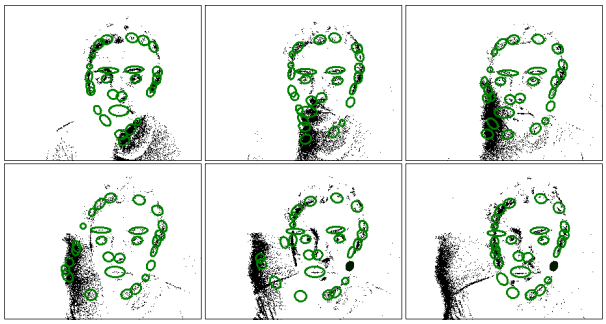


Fig. 19. Set of connected trackers is disturbed by a dynamic occlusion introduced by waving a hand in front of the face. As the hand passes in front of the face, it first attracts the trackers, displacing them from their right position. However, the system is sufficiently robust to compensate by attracting the trackers to the right position again, without losing track of the face.

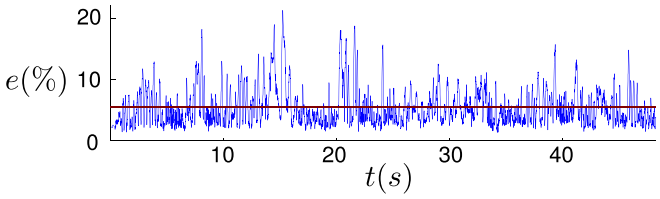


Fig. 20. Temporal evolution of the error. This error is equal to the mean error of the seven internal trackers (eyebrows, eyes, nostrils, and mouth), relative to the vertical length of the mask. Its temporal mean (indicated by the horizontal line) is equal to 5.42%.

arm from the camera. Fig. 18 shows the actual mask used in the experiment, composed of 26 blob trackers (two for the eyebrows, two for the eyes, two for the nostrils, one for the mouth, and nineteen to create the outline of the face), joined together by 40 connections. For every connection between a pair of trackers (shown in Fig. 18 as the thin lines between the ellipses), we chose to impose both a Euclidean and a torsional connection. Their equilibrium distance and angle were computed from their initial positions, and the stiffness α was experimentally set to 0.02 for both the mechanisms in every connection. The update factors from (3) and (4) were set to $\alpha_1 = 0.2$ and $\alpha_2 = 0.0002$. As a first step, we chose to use the Euclidean elastic energy for the energy-based criteria defined in Section III-C. The value of s was tested in the same way as in the previous experiment, and set to 2.5. During the recording, we asked the subject to wave his hand in front of his face to introduce occlusions. The total time of the recorded stimulus is 49 s.

Fig. 19 shows the state of the system while tracking a face. It shows how the system reacts to partial occlusions. As the hand passes in front of the face, it first attracts the trackers, displacing them from their correct position. The system is sufficiently robust to compensate by attracting the trackers to the correct position again, without losing track of the face.

The ground truth is obtained by manually selecting seven points of the face, i.e., eyebrows, eyes, nostrils, and mouth. The error is defined in the same way as in the previous experiment. First, we define the error of each individual tracker

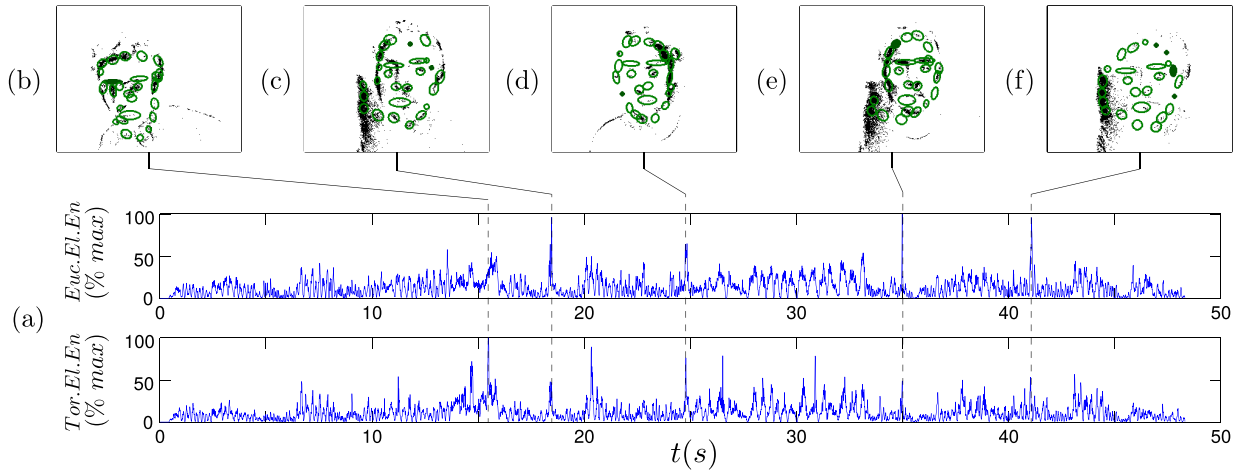


Fig. 21. (a) Evolution of the energy during the experiment. It is the sum of the energy of all the connections expressed as a percentage of the maximum value. As expected, the energy is higher when the mask is strongly deformed.

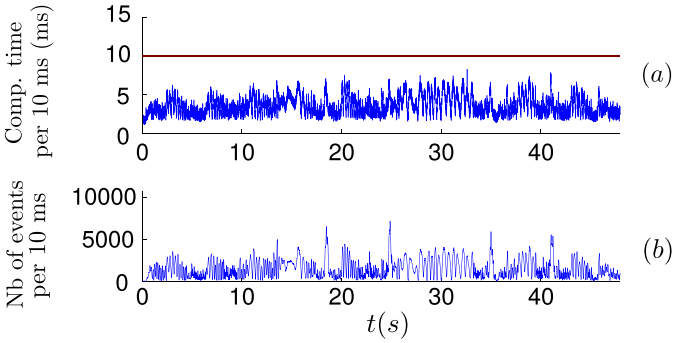


Fig. 22. (a) Computation time required for 10 ms of incoming events. As the computation time is <10 ms (the computation time is below the horizontal line), the system runs in real time. (b) Number of events processed in 10 ms. Both the results have a similar shape.

as the distance (in pixels) between the position of the tracker and the position of the corresponding feature in the image plane. Next, we define the error for each instant as the mean of the errors of the ground truth locations. It is expressed as a percentage of the vertical length of the face. Fig. 20 shows the temporal evolution of the error. We characterize the system by the temporal mean of this error, indicated in Fig. 20 by a horizontal line. In the case of this experiment, it is equal to 5.42%.

Finally, Fig. 21(a) shows the evolution of the two types of elastic energy during the experiment. For each type, we compute the cumulative sum of the energy of all the connections and express it as a percentage of its maximum during the experiment. The results show the Euclidean elastic energy and the torsional elastic energy. There is coherence between these two types of energy, since large deformations of the shape will usually imply changes both in angle and distances. Local maxima correspond to the moments when a hand was waved in front of the camera, since these partial occlusions generate large deformations.

As pointed out in Section III-D, a comparison between the different types of energy does not make sense, since they do not have the same units. The energy-based criteria defined in Section III-C using both the Euclidean elastic energy and the torsional elastic energy provide similar results. We will therefore consider in what follows only the Euclidean elastic energy.

C. Computation Time

The presented experiments were carried out using a conventional laptop, equipped with an Intel Core i7 processor and running Debian Linux. The algorithm is implemented in C++. The computation time increases with the complexity of the model of the object. For the first experiment, the model is composed of 9 blob trackers, joined together by 12 connections. This tracking can be easily achieved in real time in the system. When tracking a face, the complexity of the model is increased. As previously detailed in Section IV-B, the model consists of 26 blob trackers linked by 40 connections. The computation time is higher than in the case of the simple grid. Fig. 22(a) shows the computational time required to process 10 ms of incoming events, when the effect of the springs is computed every 55 events. If the computation time is below 10 ms (indicated in the figure by a horizontal line), the system is capable of tracking the given object in real time. As one can see, this is the case for the whole experiment. The mean time required for computing 10 ms is 3.36 ms. Fig. 22(b) represents the number of events processed by the system for 10 ms. The comparison of Fig. 22(a) and Fig. 22(b) shows that both the results have a similar shape. The ratio between these two values provides the computational time required per event.

Fig. 23 shows the ratio between the computation time required and the number of events processed every 10 ms. The mean of this result represents the mean time required per event. It is equal to 2.26×10^{-3} ms $\equiv 2.26$ μs. This is equivalent to a rate of 451 kHz.

The computation time does not depend on the tuning parameters (α, s) of the system, since the number of operations per event remains the same. The only parameter impacting the processing time is the number of events for which we update

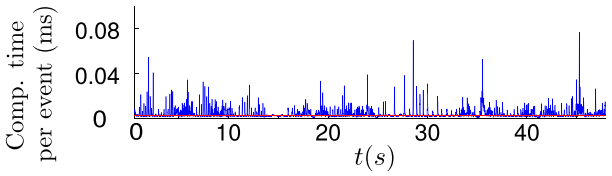


Fig. 23. Ratio between the computational time required and the number of events treated for each 10 ms. The mean of this curve represents the mean time required per event, that is equal to 2.26×10^{-3} ms $\equiv 2.26$ μs.

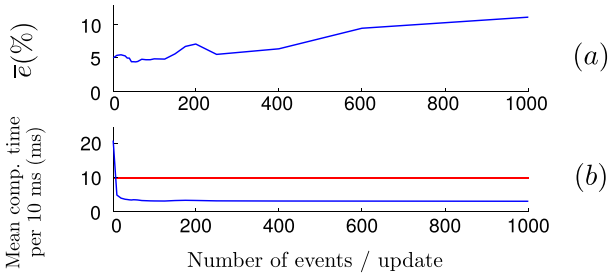


Fig. 24. Evolution of the tracking error and computation time with the number of events for which we update the state of the springs. (a) Mean tracking error. (b) Mean computation time for a time bin of 10 ms.

the springs' state. Fig. 24 shows the evolution of both the mean error and the mean computation for a time bin of 10 ms with this parameter.

V. Conclusion

A new method for visual tracking of complex objects from the output of an asynchronous event-based silicon retina has been proposed. So far, very few algorithms have been developed for tracking complex objects using this innovative technology and the time of arrival of events. The method is truly event driven as every incoming event updates the dynamics of the tracker. Compared with conventional frame-based acquisition, we have shown that neuromorphic event-driven high temporal resolution cameras allow the problem to be tackled by introducing true dynamics into the system. This paper introduces for the first time a true real-time part-based model running at the native resolution of scenes' dynamics and updating its energy at several hundred kilohertz. When correctly tuned, the energy-based criterion increases the performance of the method, and allows it to be robust to partial occlusions. An improvement of the system would be to use more specialized trackers, allowing them to track specific space-time patterns rather than just following incoming blobs of events. This should allow the method to be more robust to occlusions as it is very improbable that occlusions have the same shape as the tracked stimulus. The algorithm has been designed to flexibly create the model of the object allowing the user to add as many Gaussian trackers as needed linked by any number of connections of several types. The performance of the method is strongly dependent on the tuning of the parameters of the model. In the presented algorithm, parameters are defined experimentally according to the object to track. However, they remain relatively stable showing that there is a plateau where parameters allow the tracking of a wide variety of objects. An improvement of the approach would be to introduce learning techniques to determine the model of the object. The pool of trackers would then connect based on their temporal coactivations, thus making use of the high temporal resolution of the sensor. Current developments are tackling the implementation of this Hebbian learning strategy built from the relative displacement of an unconstrained pool of free Gaussian trackers.

Appendix

Equation (9) is a very typical equation in classical mechanics, known as the differential equation of a damped harmonic oscillator. It is a second-order differential equation, and it is usually rewritten as

$$\frac{d^2\Delta l}{dt^2} + 2\zeta\omega_0\frac{d\Delta l}{dt} + \omega_0^2 = 0 \tag{23}$$

where ω_0 is the undamped natural frequency of the oscillator and ζ is a constant parameter known as the damping ratio of the system, given by

$$\zeta = \frac{c}{2\sqrt{mk}}. \tag{24}$$

The damping ratio ζ is a dimensionless quantity that will determine the behavior of the system. According to its value, the system can be as follows.

1) *Overdamped ($\zeta > 1$):* The system returns to the equilibrium state without oscillating.
2) *Critically Damped ($\zeta = 1$):* The system returns to the equilibrium state as quickly as possible without oscillating.
3) *Underdamped ($\zeta < 1$):* The system oscillates.

As we do not wish our system to oscillate, we can impose it to be always overdamped. In that case, the solution to (23) is given by

$$\Delta l(t) = A_1 e^{\lambda_1 t} + A_2 e^{\lambda_2 t} \tag{25}$$

where A_1 and A_2 depend on the initial state of the system, and λ_1 and λ_2 are real negative coefficients, given by

$$\lambda_1 = \frac{-c - \sqrt{c^2 - 4}\ \text{km}}{2m}$$
$$\lambda_2 = \frac{-c + \sqrt{c^2 - 4}\ \text{km}}{2m}. \tag{26}$$

The elongation of the system is then given by the addition of two declining exponential functions with different time constants $1/\lambda_1$ and $1/\lambda_2$. The first of them is much smaller and corresponds to the rapid cancellation of the effect of the initial speed. The second one is bigger and describes the slow decay toward the equilibrium position.

Fig. 25 shows the different dynamics of the two declining exponential functions for an harmonic oscillator with $k = 10$ N/cm, $m = 20$ kg, $c = 42$ Ns/cm, and initial conditions $\Delta l(t = 0) = 10$ cm and $v(t = 0) = 2$ cm/s. As we can see, one of them is much faster and smaller than the other one. This allows us to approximate the position $\Delta l(t)$ of the system just by the second exponential

$$\Delta l(t) \approx A_2 e^{\lambda_2 t} \tag{27}$$

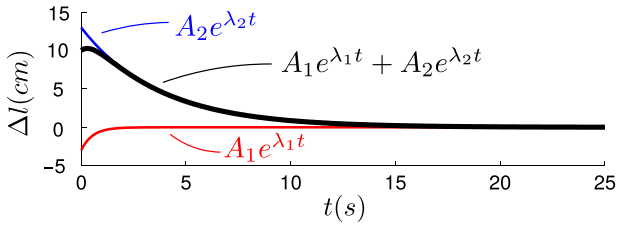


Fig. 25. Different dynamics of the two declining exponential functions for a harmonic oscillator with $k = 10$ N/cm, $m = 20$ kg, $c = 42$ Ns/cm, and initial conditions $\Delta l(t = 0) = 10$ cm and $v(t = 0) = 2$ cm/s.

from where we can isolate t that takes the value

$$t = \frac{1}{\lambda_2} \log\left(\frac{\Delta l}{A_2}\right). \tag{28}$$

Deriving (27), we obtain an approximation for the speed

$$\frac{d(\Delta l)}{dt} \approx \lambda_2 A_2 e^{\lambda_2 t}. \tag{29}$$

In addition, combining (28) and (29), we obtain the speed as a function of the position

$$\frac{d(\Delta l)}{dt} \approx \lambda_2 A_2 e^{\lambda_2 \frac{1}{\lambda_2} \log\left(\frac{\Delta l}{A_2}\right)} = \lambda_2 \Delta l. \tag{30}$$

Next, if we assume the speed to be constant during a certain period of time Δt, we can approximate it by

$$\frac{d(\Delta l)}{dt} \approx \frac{\Delta(\Delta l)}{\Delta t} \approx \lambda_2 \Delta l \tag{31}$$

where $\Delta(\Delta l)$ is the displacement experienced by the system during the time interval Δt. Isolating the displacement $\Delta(\Delta l)$ in (32), we obtain its approximate value, given by

$$\Delta(\Delta l) \approx \lambda_2 \Delta t \Delta l = \frac{-c + \sqrt{c^2 - 4 \text{ km}}}{2m} \Delta t \Delta l = \frac{\alpha}{m} \Delta l \tag{32}$$

where

$$\alpha = \frac{-c + \sqrt{c^2 - 4 \text{ km}}}{2} \Delta t \tag{33}$$

would be the stiffness parameter of our spring-like connections.

As we can see in (33), α actually depends on the mass of the object for which we are solving the differential equation, and not only on the characteristics of the harmonic oscillator. According to this, we could not assign an α to each connection, independently of the value of the masses. However, for simplicity, this is what we are going to do [see (11), (16), and (26)]. Thus, the stiffness α and the masses of the trackers become nothing but a dimensionless scaling factor.

ACKNOWLEDGMENT

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

REFERENCES

[1] G. L. Foresti, "Object recognition and tracking for remote video surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 7, pp. 1045–1062, Oct. 1999.
[2] I. Cohen and G. Medioni, "Detecting and tracking moving objects for video surveillance," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Fort Collins, CO, USA, Jun. 1999, p. 325.
[3] R. J. K. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," *Mind*, vol. 2, no. 3, p. 4, 2003.
[4] U. Neumann and S. You, "Natural feature tracking for augmented reality," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 53–64, Mar. 1999.
[5] B. Coifman, D. Beymer, P. McLauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transp. Res. C, Emerg. Technol.*, vol. 6, no. 4, pp. 271–288, 1998.
[6] M.-H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 34–58, Jan. 2002.
[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Kauai, HI, USA, Dec. 2001, pp. I-511–I-518.
[8] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proc. IEEE Int. Conf. Image Process.*, Rochester, NY, USA, Sep. 2002, pp. I-900–I-903.
[9] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.
[10] B. Wu, H. Ai, C. Huang, and S. Lao, "Fast rotation invariant multi-view face detection based on real AdaBoost," in *Proc. 6th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Seoul, Korea, May 2004, pp. 79–84.
[11] T. Mita, T. Kaneko, and O. Hori, "Joint Haar-like features for face detection," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, Beijing, China, Oct. 2005, pp. 1619–1626.
[12] M. Jones and P. Viola, "Fast multi-view face detection," Mitsubishi Electr. Res. Lab., Cambridge, MA, USA, Tech. Rep. TR-20003-96, 2003.
[13] B. Froba and A. Ernst, "Face detection with the modified census transform," in *Proc. 6th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Seoul, Korea, May 2004, pp. 91–96.
[14] P. Menezes, J. C. Barreto, and J. Dias, "Face tracking based on Haar-like features and eigenfaces," in *Proc. IFAC/EURON Symp. Intell. Auto. Vehicles*, Lisbon, Portugal, Jul. 2004.
[15] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2010-66, 2010.
[16] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comput.*, vol. C-100, no. 1, pp. 67–92, Jan. 1973.
[17] M. C. Burl, M. Weber, and P. Perona, "A probabilistic approach to object recognition using local photometry and global geometry," in *Proc. 5th Eur. Conf. Comput. Vis.*, Freiburg im Breisgau, Germany, 1998, pp. 628–641.
[18] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Madison, WI, USA, Jun. 2003, pp. II-264–II-271.
[19] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 55–79, 2005.
[20] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 1014–1021.
[21] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 10–17.
[22] S. Zuffi, O. Freifeld, and M. J. Black, "From pictorial structures to deformable structures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3546–3553.
[23] M. Andriluka, S. Roth, and B. Schiele, "Discriminative appearance models for pictorial structures," *Int. J. Comput. Vis.*, vol. 99, no. 3, pp. 259–280, 2012.
[24] Y. Watanabe, T. Komuro, and M. Ishikawa, "955-fps real-time shape measurement of a moving/deforming object using high-speed vision for numerous-point analysis," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 3192–3197.

[25] T. Senoo, Y. Yamakawa, Y. Watanabe, H. Oku, and M. Ishikawa, "High-speed vision and its application systems," *J. Robot. Mechatron.*, vol. 26, no. 3, pp. 287–301, 2014.

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

[27] C. Posch, D. Matolin, and R. Wohlgenannt, "An asynchronous time-based image sensor," in *Proc. IEEE Int. Symp. Circuits Syst.*, Seattle, WA, USA, May 2008, pp. 2130–2133.

[28] S. Chen, P. Akselrod, B. Zhao, J. A. Perez-Carrasco, B. Linares-Barranco, and E. Culurciello, "Efficient feedforward categorization of objects and human postures with address-event image sensors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 302–314, Feb. 2012.

[29] J. A. Perez-Carrasco *et al.*, "Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward ConvNets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2706–2719, Nov. 2013.

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

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

[32] R. Benosman, S.-H. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan, "Asynchronous frameless event-based optical flow," *Neural Netw.*, vol. 27, pp. 32–37, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2011.11.001

[33] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 407–417, Feb. 2014.

[34] Z. Ni, C. Pacoret, R. Benosman, S. Ieng, and S. Regnier, "Asynchronous event-based high speed vision for microparticle tracking," *J. Microsc.*, vol. 245, no. 3, pp. 236–244, 2012. [Online]. Available: http://dx.doi.org/10.1111/j.1365-2818.2011.03565.x

[35] M. Litzenberger *et al.*, "Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Toronto, ON, Canada, Sep. 2006, pp. 653–658.

[36] M. Litzenberger *et al.*, "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *Proc. 12th-Signal Process. Edu. Workshop, 4th Digit. Signal Process. Workshop,* Grand Teton National Park, WY, USA, Sep. 2006, pp. 173–178.

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

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

[39] D. Drazen, P. Lichtsteiner, P. Häfliger, T. Delbrück, and A. Jensen, "Toward real-time particle tracking using an event-based dynamic vision sensor," *Experim. Fluids*, vol. 51, no. 5, pp. 1465–1469, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00348-011-1207-y

[40] Z. Ni, A. Bolopion, J. Agnus, R. Benosman, and S. Regnier, "Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1081–1089, Oct. 2012.

[41] X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, "Asynchronous event-based multikernel algorithm for high-speed visual features tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[42] 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.

[43] T. Delbrück, "Silicon retina with correlation-based, velocity-tuned pixels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 4, no. 3, pp. 529–541, May 1993.

[44] R. Etienne-Cummings, J. Van der Spiegel, and P. Mueller, "A focal plane visual motion measurement sensor," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 44, no. 1, pp. 55–66, Jan. 1997.

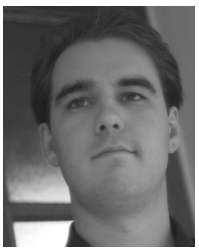
[45] J. Krammer and C. Koch, "Pulse-based analog VLSI velocity sensors," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 44, no. 2, pp. 86–101, Feb. 1997.

[46] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 5, pp. 416–434, May 2000.

[47] J. R. Taylor, *Classical Mechanics*. Sausalito, CA, USA: Univ. Science, 2005.

[48] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, vol. 60. Berlin, Germany: Springer-Verlag, 1989.

David Reverter Valeiras received the B.Sc. degree in mechanical engineering from the Escola Técnica Superior de Enxeñeiros Industriais de Vigo, Vigo, Spain, in 2012, and the M.Sc. degree in advanced systems and robotics from Université Pierre et Marie Curie, Paris, France, in 2013. He is currently pursuing the Ph.D. degree in event-based vision and its application to robotics with the Institut de la Vision, Sorbonne Universités, Paris.



Xavier Lagorce received the B.Sc. degree, the Agrégation degree in electrical engineering, and the M.Sc. degree in advanced systems and robotics from the École Normale Supérieure de Cachan, Cachan, France, in 2008, 2010, and 2011, respectively. He is currently pursuing the Ph.D. degree in neurally inspired hardware systems and sensors with the Institut de la Vision, Sorbonne Universités, Paris, France.

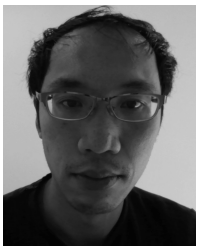
Xavier Clady received the Engineering degree from Telecom Physique, Strasbourg, France, in 1998, with a specialization in image processing and acquisition, and the Ph.D. degree in computer vision for robotics from Blaise Pascal University, Clermont-Ferrand, France, in 2003.

He is currently an Associate Professor with Université Pierre et Marie Curie—Paris 6, Paris, France. He is involved in research activities with the Institut de la Vision (UMR S-968, UMR 7210), Sorbonne Universités, Paris. His current research interests include computer vision, image processing, and pattern recognition applied in objects, human or gesture recognition, and neuromorphic vision.

Chiara Bartolozzi (M'11) received the Laurea (Hons.) degree in biomedical engineering from the University of Genova, Genoa, Italy, in 2001, the Ph.D. degree in natural sciences from the Department of Physics, Federal Institute of Technology Zurich, Zurich, Switzerland, and the Ph.D. degree in neuroscience from the Neuroscience Center Zurich, Zurich, in 2007.

She joined the Istituto Italiano di Tecnologia, Genoa, first as a Post-Doctoral Researcher with the Department of Robotics, Brain and Cognitive Sciences and then as a Researcher with the iCub Facility, where she is currently the Head of the Neuromorphic Systems and Interfaces Group. She coordinated the eMorph (ICT-FET 231467) project that delivered the unique neuromorphic iCub humanoid platform, developing both the hardware integration and the computational framework for event-driven robotics. Her current research interests include design of event-driven technology and their exploitation for the development of novel robotic platforms.

Dr. Bartolozzi is a member of the IEEE Circuits and Systems Society, the Sensory Systems Committee, and the Neural Systems and Applications Committee.



Sio-Hoi Ieng received the Ph.D. degree in computer vision from Université Pierre et Marie Curie, Paris, France, in 2005.

He was involved in the geometric modeling of noncentral catadioptric vision sensors and their link to the caustic surface. He is currently an Associate Professor with Université Pierre et Marie Curie, and a member of the Institut de la Vision, Sorbonne Universités, Paris. His current research interests include computer vision, with a special reference to the understanding of general vision sensors, cameras networks, neuromorphic event-driven vision, and event-based signal processing.

Ryad Benosman is currently a Full Professor with Universit Pierre et Marie Curie, Paris, France. He leads the Neuromorphic Vision and Natural Computation Group. He focuses mainly on neuromorphic engineering, visual computation, and sensing. He is a Pioneer of omnidirectional vision systems, complex perception systems, variant scale sensors, and noncentral sensors. He is invested in applying neuromorphic computation to retina prosthetics, and is the Co-Founder of the startup company Pixium Vision. His current research interests include the understanding of the computation operated by the visual system with the goal to establish the link between computational and biological vision.