Bioinspired Visual Motion Estimation

Understanding how biological vision operates in the spatio–temporal domain informs new motion-sensing technology.

By Garrick Orchard and Ralph Etienne-Cummings, *Fellow IEEE*

ABSTRACT | Visual motion estimation is a computationally intensive, but important task for sighted animals. Replicating the robustness and efficiency of biological visual motion estimation in artificial systems would significantly enhance the capabilities of future robotic agents. Twenty five years ago, in this very journal, Carver Mead outlined his argument for replicating biological processing in silicon circuits. His vision served as the foundation for the field of neuromorphic engineering, which has experienced a rapid growth in interest over recent years as the ideas and technologies mature. Replicating biological visual sensing was one of the first tasks attempted in the neuromorphic field. In this paper, we focus specifically on the task of visual motion estimation. We describe the task itself, present the progression of works from the early first attempts through to the modern day state-of-the-art, and provide an outlook for future directions in the field.

KEYWORDS | Analog-to-digital integrated circuits; image motion analysis; motion estimation; sensor systems; velocity measurement; very large scale integration

I. INTRODUCTION

Visual sensing is a computationally intensive, but crucial task for sighted animals [1]. Although motion estimation is just one aspect of visual sensing, its importance is easily understood when observing its wide range of uses in biology [2], including depth perception, egomotion estimation, collision avoidance and triggering escape reflexes, time-to-contact estimation (landing control), prey detection and identification, segmentation by motion, and visual odometry.

The ability to reliably estimate visual motion in artificial systems would find applications ranging from surveillance and tracking, to visual flight control [2], to video compression, image stabilization, and even the computer mouse [3]. However, the most relevant application of bioinspired visual motion estimation is for embedded sensing onboard robotic agents capable of moving through and interacting with their environment, since this is precisely the function for which biological visual motion systems have evolved. Key characteristics which distinguish this application from others are: the sensor must operate in real time, the sensor must operate under egomotion, the sensor must be physically carried by the agent, and the sensor should provide information which is relevant for enabling the agent to interact meaningfully with its environment. In fact, Wolpert argues in a recent Technology, Entertainment, Design (TED) talk that control of motion is the primary purpose of the brain [4].

There are significant differences between biological and artificial systems regarding how visual information is acquired and processed. State-of-the-art modern visual motion estimation methods still rely on capturing sequences of images (frames) in rapid succession, even though the majority of data in these images is redundant [5]. The problem of storing and transmitting this redundant information is partially overcome by using dedicated video compression application-specific integrated circuits (ASICs), or in the case of standalone visual motion sensors, by computing on chip [6]–[9]. Nevertheless, these artificial approaches capture frames at predetermined discrete time points regardless of the visual scene. On the other hand, biological retinae continuously capture data and perform a combination of compression and preprocessing in analog (using graded potentials) at the focal plane itself, with the visual scene largely driving when and where data are transmitted as spikes (voltage pulses) down the optic tract. Spikes are similar to digital pulses in artificial systems in that their signal amplitude can be restored and they are therefore particularly useful for

Manuscript received May 21, 2014; revised July 31, 2014; accepted August 2, 2014. Date of publication September 4, 2014; date of current version September 16, 2014.

G. Orchard is with the Singapore Institute for Neurotechnology (SINAPSE), National University of Singapore, Singapore 117456 (e-mail: garrickorchard@nus.edu.sg).
R. Etienne-Cummings is with the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: retienne@jhu.edu).

Digital Object Identifier: 10.1109/JPROC.2014.2346763

communicating over longer distances, such as along the optical tract.

Processing also differs significantly between biological and artificial systems. Similarly to how artificial systems typically capture data at a constant rate, they must also compute at a constant rate to ensure all the captured data are processed, thus processing of visual information continues even if the scene is static. On the other hand, computation in biological systems is driven by the sparse captured data (spikes), in turn ensuring that neuron activation is sparse [10] (since neuron activation is driven by the sparse incoming data). This sparsity combined with the low power consumption of neurons which are not computing [11] results in significant energy savings. Modern biologically inspired sensors generate sparse data (events) in response to activity in the scene [12], [13] and these data can be used to drive sparse computation on modern neural simulator platforms [14]. Together, these sensors and neural simulators allow both data capture and computation to scale with scene activity.

The architectures used for processing also differ. Typical artificial systems compute on a small number of parallel processors, each of which performs sequential operations in a precise repeatable manner, and operates at a timescale on the order of nanoseconds. On the other hand, biology relies on massively parallel processing using a very large number of imprecise computing elements (neurons), each of which operates on a timescale on the order of milliseconds [15]. However, parallelism in artificial systems is increasing, particularly for visual processing [graphical processing units (GPUs)], and emerging custom neural hardware platforms [14], [16]. State-of-the-art ASICs dedicated to visual motion estimation are also optimized to perform processing in as parallel a manner as possible.

Despite the imprecise nature of individual neurons, biological systems perform robustly and continue to do so even after the death of individual neurons. The same is not true of artificial systems, where a single fault can cause catastrophic failure of the entire system. Similar fault tolerance is highly desirable in artificial systems, especially as the number of transistors per device continues to increase, and as the size limits of silicon technology continue to be pushed. Even looking past silicon to nanotechnology, device yield continues to be a major challenge [17].

Beyond handling minor faults, biological systems are also able to learn and adapt to changes in the visual system itself [18], as well as to different environments through visual experience [19], allowing them to operate effectively under a wide range visual of conditions. Such self-contained online learning and adaptation would also prove valuable for artificial systems, removing any need for manual tuning of parameters for operation in different environments.

Biology's robust processing, low power consumption, and ability to learn and adapt all present desirable characteristics for artificial systems and drive the field of bioinspired sensing and computation.

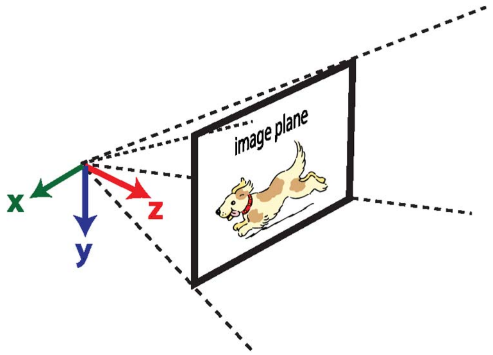

Fig. 1. ***Definition of axes used in (1) for a pinhole camera approximation with unit focal length. The z-axis points out toward the scene perpendicular to the image plane, while the x- and y-axes are parallel to the image plane.***

In this paper, we provide a brief introduction to the visual motion estimation problem and provide background on the methods used in traditional computer vision versus biological systems, before reviewing advances in bioinspired visual motion estimation for artificial systems, presenting our own approach to the problem, and discussing future directions.

II. THE VISUAL MOTION ESTIMATION PROBLEM

When relative motion is present between an observer (eye or camera) and objects in a scene, the projections of these objects onto the image plane (retina or pixel sensor) will move. Visual motion estimation is the task of estimating how the projections of these objects move on the image plane.

Assuming a pinhole camera approximation, with the image plane at unit focal length, visual motion can be described as a function of the image plane coordinates (x, y), the relative rotation $(\omega_x, \omega_y, \omega_z)$ and translation (T_x, T_y, T_z) between the camera and the object being viewed, and the depth of the object (z) [20], as shown in Fig. 1 and in

$$\begin{aligned}\frac{\delta x}{\delta t} &= \frac{T_z x - T_x}{z} - \omega_y + \omega_z y + \omega_x xy - \omega_y x^2\\ \frac{\delta y}{\delta t} &= \frac{T_z y - T_y}{z} + \omega_x - \omega_z x - \omega_y xy + \omega_x y^2.\end{aligned} \qquad (1)$$

Visual motion is constrained to lie in the image plane and therefore has no z-direction component. The first term in each equation describes the visual motion due to translation, which is depth dependent, while the remaining terms describe visual motion due to rotation, which is independent of depth. In other words, visual motion due to rotation does not depend on the structure of the scene, while visual motion due to translation does depend on scene structure. The rotations and translations in the equation above are for motion of the camera relative to the origin as depicted in Fig. 1.

The relationship described in (1) also shows that multiple different combinations of scene structure and relative motion can result in identical visual motion. Thus, visual motion alone is not enough to infer relative motion or scene structure and additional information is required. For example, if the scene is static and the rotational motion of the sensor is known, then the translational motion direction can be determined, and a relationship between scene depth and camera translation speed can be obtained. Thus, measuring or even eliminating rotational motion allows additional valuable information to be derived from visual motion.

Visual motion can only be estimated in the presence of an intensity gradient. A shape of uniform color moving against a background of identical color will have no intensity gradient and will, therefore, not elicit a visual motion stimulus. More specifically, for motion to be detected, the intensity gradient must be nonzero in the direction of motion. This is an example of the aperture problem [21], to which all visual systems are prone, and is illustrated in Fig. 2.

When considering only a small image region which has no intensity gradient in a particular direction, the magnitude of image velocity in that direction cannot be determined unless additional information is available. The component of motion in the direction of the maximum image gradient can be determined, and is known as the "normal flow," since it is perpendicular (normal) to the edge orientation. The larger the image region under consideration, the more likely it will contain gradients in different directions, helping to alleviate the aperture problem. A common approach is to simultaneously consider multiple neighboring image regions and assume their motion to be either consistent [22] or smoothly varying [23], thus providing the additional constraint required to uniquely determine motion.

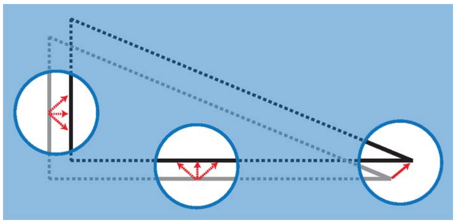

Fig. 2. ***Aperture problem illustrated with a triangle moving between an initial position (gray) and a final position (black), while viewed only through three apertures (blue circles). For the leftmost aperture, the image only varies in the horizontal direction, and, therefore, only the horizontal component of motion can be estimated. Similarly, for the middle aperture, only the vertical component of motion can be estimated. For the rightmost aperture, the viewed image varies along both horizontal and vertical directions and, therefore, motion can be uniquely determined.***

III. APPROACHES TO VISUAL MOTION ESTIMATION

For the purpose of providing background for later sections, we introduce here the basic theory underlying each of the three main classes of visual motion estimation approaches: correlation methods, gradient methods, and frequency methods.

Underlying all three of these methods is the assumption of brightness constancy, known as the brightness constancy constraint, which states that the brightness of a point remains constant after moving a small distance on the image plane $[\Delta x, \Delta y]$, within a small period of time Δt. Formally, this can be written as

$$I(x, y, t) \approx I(x + \Delta x, y + \Delta y, t + \Delta t) \qquad (2)$$

where $I(x, y, t)$ is the intensity of the point located at (x, y) on the image plane at time t.

A. Correlation Methods

Correlation methods for motion estimation rely on detecting the same visual feature at different points in time as it moves across the image plane. Correlation is used to determine whether two feature signals detected at different points in time relate to the same or different features. The feature signals on which correlation is computed can take the form of continuous-time analog signals, discrete-time analog signals, discrete digital signals, or even single bit binary tokens indicating only the presence or absence of a feature ("token methods"). The change in feature location can be combined with the change in time between detections to determine the feature's motion. The simplest features to use are brightness patterns, or derivatives thereof, the appearance of which remains constant over small time periods as described in (2).

Most state-of-the-art commercially available motion estimation ASICs rely on correlation methods [6]–[8]. These devices capture frames at high frame rates and detect correlations in local pixel intensity patterns between images to determine motion. One very common approach is the sum of absolute differences (SAD) block-matching algorithm [9], which matches "blocks" of pixels between frames by computing the SAD for pixel intensities. This process is repeated for many different blocks and the estimates from all of these blocks are combined to determine motion.

B. Gradient Methods

Gradient methods rely on the Taylor series expansion of (2), which for a first-order expansion can be rearranged into the form

$$\frac{\delta I(x,y,t)}{\delta x}\frac{\delta x}{\delta t}+\frac{\delta I(x,y,t)}{\delta y}\frac{\delta y}{\delta t}+\frac{\delta I(x,y,t)}{\delta t}=0 \tag{3}$$

where $\delta x/\delta t$ and $\delta y/\delta t$ are the visual motion values that must be estimated, while $\delta I(x,y,t)/\delta x$, $\delta I(x,y,t)/\delta y$, and $\delta I(x,y,t)/\delta t$ are intensity derivatives which can be obtained from captured frames.

Note that if the intensity derivative in either spatial direction is zero, then motion in that direction is removed from the equation and cannot be estimated. This is the aperture problem discussed in Section II. When these spatial derivatives are nonzero, but small, they are sensitive to noise and can still result in erroneous motion measurements. Even if accurate nonzero intensity derivatives are available, (3) is a single equation with two unknowns and thus does not provide a unique solution.

To arrive at a unique solution, additional constraints must be imposed, such as that the motion of all points in an image patch will be equal (as is used in the Lucas–Kanade algorithm [22]), or that motion varies smoothly across image locations (as is used in the Horn–Schunck algorithm [23]).

C. Frequency Methods

Frequency-based methods rely on the observation that there is a relationship between temporal frequency, spatial frequency, and velocity [24]. For simplicity, consider a point (Dirac delta function [25]) moving in the x-direction. This point will trace out a line in the space-time plot on the left of Fig. 3 with slope equal to the velocity. Taking the Fourier transform results in a line in frequency space with slope equal to the inverse of velocity, as shown on the right of Fig. 3

$$\begin{aligned} I(x,t) &= \delta(x-v_x t)\\ v_x &= \frac{\Delta x}{\Delta t}\\ I_\omega(\omega_x,\omega_t) &= \delta(\omega_t+v_x\omega_x)\\ \frac{1}{v_x} &= -\frac{\Delta\omega_x}{\Delta\omega_t} \end{aligned} \tag{4}$$

where $I(x,t)$ is the intensity at location x at time t, v_x is the velocity in the x-direction, δ is the Dirac delta function, and $I_\omega(\omega_x,\omega_t)$ is the Fourier transform [26] of $I(x,t)$.

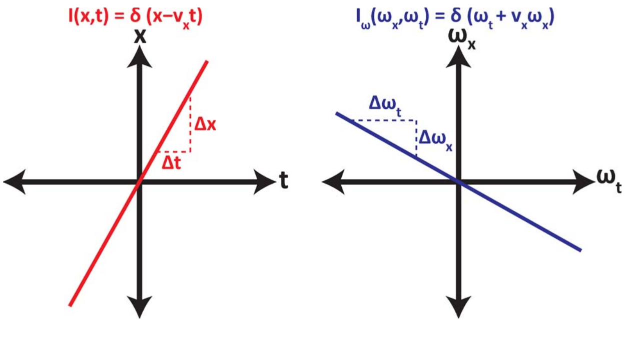

Fig. 3. ***Frequency representation of the motion described in (4). The left plot (red) shows the relationship between time and location for a point moving with constant velocity. The right plot (blue) shows the velocity-dependent relationship between spatial and temporal frequency for the same moving point.***

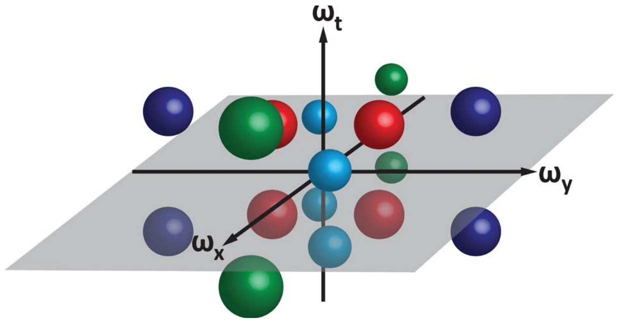

Fig. 4. ***Example placement of four different quadrature pairs of spatio–temporal filters in frequency space. Each quadrature pair is indicated by a different color and is symmetric about the ω_t-axis and the plane $\omega_t = 0$ (shaded gray). The red and dark blue quadrature pairs are sensitive to different velocities in the y-direction. The green quadrature pair is sensitive to motion of a specific speed in the x-direction. The light blue quadrature pair is most sensitive to a particular velocity which has both x and y components.***

The case described above is an ideal case where the stimulus is a point (Dirac delta function) and, therefore, has equal energy at all spatial frequencies. In the more general case of a stimulus with an arbitrary distribution of spatial frequency content, the energy will still be constrained to lie along the line shown on the right of Fig. 3.

Visual motion can be estimated by finding the slope of the line in Fig. 3, which can be achieved by tiling frequency space with spatio–temporal filters, as shown in Fig. 4, and combining their responses to find the location of the energy peak.

D. Implementation

Typical artificial approaches to motion estimation rely on a frame-based camera to capture snapshots (frames) of the scene at fixed intervals. A processor is then used to apply one of the three methods described above to estimate visual motion. In computer vision, typically correlation or gradient-based methods are used, with frequency methods regarded as bioinspired approach as will be discussed in Section IV.

In all three of the methods outlined above, increasing the frame rate improves the accuracy of the algorithm because the brightness constancy constraint relies on the assumption of a small time period between observations (frames).

For correlation methods, increased frame rate also decreases the distance a feature can move between frames, thereby helping to restrict the search region for that feature in subsequent frames. For example, typical optical mouse algorithms operate at thousands of frames per second (albeit small frames), and the search can be in increments of fractions of a pixel. For gradient and frequency methods, increased frame rate is equivalent to a higher temporal sampling rate, reducing aliasing and allowing for higher order digital filters to be used when estimating the temporal derivative or frequency.

However, increasing frame rate also increases the computing power required to sustain real-time operation, since more frames must be processed within the same time period. The additional computing needs are typically met by using more powerful hardware, such as GPUs, field-programmable gate arrays (FPGAs), and custom ASICs. Tight coupling of a frame-based sensor and ASIC is often used to reduce communication costs for embedded applications, such as for a standalone motion estimation unit relying on a high frame rate, or in-camera video compression relying on block matching.

Some artificial approaches to motion estimation do not rely on frames, but instead process on continuous-time analog signals derived from complementary metal–oxide–semiconductor (CMOS) photodiodes. Notable examples include [3] and [27].

For all three of the approaches described, motion estimates within a local image region can be computed independently of motion estimates for other image regions. This opens up the possibility of simultaneously computing motion for different image regions in parallel. GPUs, FPGAs, and ASICs all take advantage of this.

The computational complexity of each approach varies. A major disadvantage of gradient methods is that they typically require the expensive computation of a matrix inverse (or pseudoinverse) in order to find the solution which best satisfies (with the least square error) the brightness constancy (3) and secondary constraints. However, gradient methods have the advantage of computing on instantaneous gradient values and, therefore, require very little memory (only enough to estimate temporal gradients).

Computing correlations for correlation methods is computationally simpler, but requires the system to have memory of previously observed features, whether by delaying feature signals or by explicitly storing them.

Digital implementations of frequency methods require even more memory because many time points are required to detect temporal frequencies, particularly if very low frequencies are present. The spatial and temporal frequency content of the scene is typically not known in advance, so frequency methods require a large number of spatio–temporal filters in order to accurately detect the frequency content of different motion stimuli. Implementing these filters digitally is costly both in terms of computation and memory.

The robustness of the approaches also varies. When computing the correlation between two signals, it is not always clear whether a large output has resulted from a strong correlation or from large input signals. Signals can be normalized before correlation to overcome this ambiguity. However, the typical signals on which correlation is computed arise from a combination of image motion and image spatial contrast, and it is not always possible to disambiguate the effects of motion versus contrast in the final output.

Gradient methods rely on the ratio of the temporal gradient to the spatial gradient (3) and are therefore very sensitive to noise, particularly when spatial gradient signals are weak.

Spatio–temporal frequency models are far more robust to noise, but as discussed above, the computational and memory requirements for estimating frequencies is far higher.

IV. VISUAL MOTION ESTIMATION IN BIOLOGY

The animal kingdom is incredibly diverse, and the visual systems of many creatures have evolved independently [29] (although they may share a common origin), resulting in variations in size, number, shape, location, wavelength sensitivity, and acuity of eyes across families [30]. Similarly to how eyes vary by family, so does the process of visual motion estimation. For the sake of discussion we will focus specifically on *Drosophila* (fruit flies) and macaque (monkeys), which are both well-studied genera, but possess very different visual systems. Properties of the vision systems of *Drosophila* and macaque generalize to many other insects and mammals, respectively. This section is intended only as a brief introduction. For more details, the reader is directed to neuroscience reviews covering *Drosophila* [31] and primate vision [32].

Despite the differences between macaque and *Drosophila* vision systems, there are also many similarities. In both systems, initial computation is performed at the focal plane by neurons which communicate using a combination of spiking and graded responses. These neurons respond to intensity changes, with responses to intensity increases (ON) and decreases (OFF) processed independently by parallel pathways [33]. In *Drosophila*, motion estimation can still be performed in the L1 (ON) pathway if the L2 (OFF) pathway is blocked, and *vice versa*. In macaque, direction-selective starburst amacrine cells (SACs) [34] can be found in the retina itself, and separate SACs are used to process ON and OFF responses in parallel. Although direction selective, these cells show very limited speed sensitivity, with true velocity-sensitive neurons only found in higher visual areas.

In *Drosophila*, as with many other animals, a fast escape reflex triggered by visual motion aids in evading approaching predators. Low latency motion detection is critical for this task and is achieved by keeping the motion processing circuitry relatively simple and located close to the photoreceptors [35]. These motion processing circuits rely on correlation methods, which are fast, and allow for compact implementation which can be realized in the limited space available near the photoreceptors [36]. Furthermore, these circuits are tuned to detect stimuli characteristics indicative of an approaching predator, rather than to accurately measure a wide range of complex motion stimuli.

On the other hand, the macaque visual system is more concerned with accurately estimating motion for a wide range speeds and visual stimuli than detecting approaching predators. Accurate motion estimates help to achieve a deeper scene understanding, which can then be used for action planning. This link between motion estimation and scene understanding requires interaction between motion detectors and cortex, and motion-sensitive neurons are, therefore, found in various cortical areas [37].

Low latency detection is also desirable for macaque, but is not as important as when triggering escape reflexes, allowing the luxury of using more complex motion estimation methods and taking advantage of the computational resources available in cortex. The macaque visual system is, therefore, not restricted to using simple correlation methods. The presence of spatio–temporal frequency-sensitive neurons in the middle temporal (MT) visual area, which plays an important role in motion estimation [38], suggests that motion estimation in primates relies on frequency methods [39].

In both macaque and *Drosophila*, the visual motion estimation system is tightly coupled with the motor system. In macaque, the vestibular ocular reflex is important for visual perception [40]. Visual inputs can trigger saccades, and saccades can suppress visual responses [41]. In *Drosophila*, the visual system can trigger motor responses through the optomotor reflex, and flight control is heavily reliant on visual motion estimation [2].

It is, therefore, important when considering biological vision systems to note that they do not exist in isolation. The visual system is part of an embodied system capable of moving through, and interacting with the environment. The visual and motor systems are tightly coupled and deficits in either system can affect the other, as documented in both primates [42] and *Drosophila* [43]. The optomotor response is so strong in many insects that motor outputs in response to visual stimuli can provide insight into the visual system.

It was through an investigation of the optomotor response of the beetle Clorophanus in the 1950s that Hassenstein and Reichardt arrived at their seminal model of the elementary motion detector (EMD) [36], shown in Fig. 5(a). The Hassenstein–Reichardt EMD computes the correlation between the signal of one photoreceptor and the time-delayed signal of a neighboring photoreceptor. The delay is typically modeled as a low-pass filter and correlation performed as multiplication. Strong correlation indicates the presence of motion in the preferred direction. A mirror-symmetric circuit detects motion in the opposite direction, and the difference between the circuit outputs indicates motion direction.

The Hassenstein–Reichardt EMD does not provide a direct measure of the stimulus velocity. Instead, it provides an indication of how well the motion stimulus matches the EMD's preferred combination of speed and spatial frequency. Multiple combinations of speed and spatial frequency can result in the same EMD output magnitude, so speed cannot be uniquely determined. Even if the spatial frequency is known, there are speeds greater than and less than the EMD's preferred speed for which the response magnitude would be equal, so speed would still not be uniquely determined.

These observations led to many interesting predictions which were later verified. Evidence of the existence of the Hassenstein–Reichardt model has since been found in many other visual systems, including that of *Drosophila*.

By the 1960s, Hubel and Weisel had isolated directionally selective units in the cat cortex [44]. Later, similar responses were observed in the tectum of pigeons and frogs. Barlow and Levick found such directional responses even earlier in the visual pathway of the rabbit, in the retina itself. Based on their recordings, they proposed what is now known as the Barlow–Levick model [45] shown in Fig. 5(b). The Barlow–Levick model relies on inhibition instead of excitation as the underlying mechanism, with null direction motion inhibiting a motion unit's response. Although the presence of Hassenstein–Reichardt and Barlow–Levick motion models has been ruled out in macaque, a similar mechanism relying on unbalanced inhibition is thought to underlie directional selectivity of starburst amacrine cells in primate retina [34].

As mentioned earlier, the presence of cells in MT sensitive to specific spatio–temporal frequencies indicates that frequency methods likely underlie motion perception in macaque. In 1985, Adelson and Bergen [24] and Watson and Ahumada [46] proposed similar architectures for motion estimation based on spatio–temporal frequency filtering. The Adelson–Bergen motion energy model is shown in Fig. 5(c). Each unit computes the energy at a specific spatio–temporal frequency, with the relationship between spatial and temporal frequency being indicative of speed, as outlined in Section III. Adelson and Bergen also showed how the opponent energy output by their model was equivalent to the output of the Hassenstein–Reichardt model. The Adelson–Bergen model has since been used to explain many visual illusions [47], [48] and Simoncelli and Heeger [49] have proposed a model describing how the

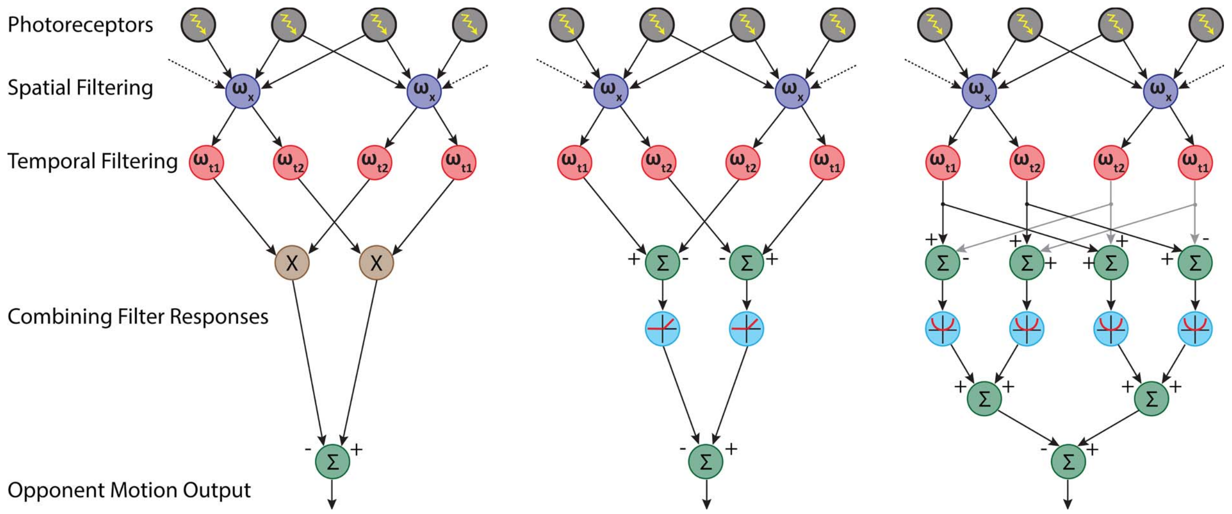

Fig. 5. *Comparison of the Hassenstein–Reichardt (left), Barlow–Levick (center), and Adelson–Bergen (right) models, similar to that in [28]. Spatial filters (dark blue) can either differ in location (as shown) or in phase. Two temporal filters (red) are used, with the second (ω_{t2}) having a longer delay than the first (ω_{t1}). The Reichardt model detects correlation between the signal at one location and the delayed signal from a neighboring location. The correlation can be modeled as a multiplication (brown), or as a logical AND if using single bit signals. The Barlow–Levick model instead uses the signal-to-block response to motion in the null direction. This can be modeled as a summation (green) followed by half-wave rectification (light blue) to prevent negative intermediate responses. If using 1-b signals, the inhibition can be modeled as an AND gate with the inhibiting (delayed) input negated. The Adelson–Bergen model combines separable spatio–temporal filter responses incorporating a squaring (light blue) nonlinearity to compute motion energy. The output of the Adelson–Bergen model is formally equivalent to that of the Hassenstein–Reichardt model [24].*

spatio–temporal responses in MT may be computed in biology.

Simoncelli [50] also took the brightness constancy constraint (3) relied upon by gradient methods and used it to develop a probabilistic Bayesian framework capable of explaining responses of MT neurons, although the framework does not describe how such responses are computed physiologically.

In the aforementioned models, individual motion units do not encode velocity directly, but rather are selective to specific spatio–temporal stimuli. To infer velocity, the responses of motion units must be combined as described in [51].

V. REVIEW OF BIOINSPIRED WORKS

In this section, we discuss integrated real-time bioinspired visual motion estimation works, which are compared in Table 1. The discussion is divided into three subsections, one for each of the three methods described in Section III, and within each subsection, works are described in chronological order.

A. Gradient Methods

In his seminal 1986 thesis, Tanner [3] presented two very large scale integration (VLSI) implementations of visual motion estimation. The first made use of the correlation method and will be discussed in Section V-B. Tanner's second implementation used a 2-D analog VLSI gradient-based approach relying a feedback loop to arrive at a minimum-error solution which simultaneously satisfies both the brightness constancy constraint (3) and a local smoothness constraint. This gradient-based approach was later adopted and extended in other works by Stocker [90] and Mehta [86].

B. Correlation Methods

1) Block Matching: Tanner's second implementation uses a correlation method. It comprises a linear array of 16 pixels which is sampled and binarized in a single step to produce a binary image. These binary images are captured and their correlation to shifted versions of an initial image is computed (the shifts are 1 pixel left, no shift, and 1 pixel right). Digital pulses indicate when leftward or rightward displacement by a single pixel has occurred, at which point a new initial image is captured and the process is repeated. Velocity is encoded by the digital pulse rate at the output of the sensor. Gottardi and Yang [60] would later present a similar approach using charge-couple device (CCD) pixels coupled with CMOS circuity, but capable of detecting motion of up to 5 pixels per image.

Yakovleff and Moini [63] presented a local matching approach which uses the sign of temporal gradients as the feature to be matched by digital circuits.

Arreguit *et al.* [64] presented the first 2-D array for block matching, which used spatial edges as features and was designed for use as a pointing device (computer

Table 1 Summary of Bioinspired VLSI Visual Motion Estimation Works

Author	Year	Process	Array	Motion	Method	Feature
Tanner [3]	1986	2μ	1D (1x16)	Global	Block-Matching	Intensity variation
Tanner [3]	1986	1.5μ	2D (8x8)	Global	Gradient	-
Franceschini [52]	1989	-	1D (1x100)	Local	Token	Temporal edge
Andreou [53]	1991	2μ	1D (1x25)	Global	Reichardt	ON-center OFF-surround
Etienne-Cummings [54]	1992	2μ	2D (5x5)	Global	Token (TI)	Temporal edge of center surround
Horiuchi [55], [56]	1992	2μ	1D (1x17)	Local	Token (TS)	Temporal edge
Delbruck [57]	1993	2μ	2D (25x25)	Local	Reichardt	Temporal contrast
Sarpeshkar [58], [59]	1993	2μ	NA NA	NA	Token	Temporal edge of spatial contrast
Gottardi [60]	1995	2μ	1D (1x115)	Global	Block-Matching	Intensity values
Kramer [61], [62]	1995	2μ	1D (1x8)	Local	Token (FS)	Temporal edge
Yakovleff [63]	1996	2μ	1D (1x61)	Local	Block-Matching	Sign of spatiotemporal gradients
Arreguit [64]	1996	2μ	2D (9̃x9)	Local	Block-Matching	Spatial edge
Etienne-Cummings [65], [66]	1997	2μ	2D (9x9)	Global	Token	Temporal edge of center surround
Moini [67], [68]	1997	1.2μ	2x1D (2x64)	Local	Block-Matching	Spatiotemporal templates
Harrison [69]	1998	2μ	2D (1x2)	Local	Reichardt	Temporal contrast
Higgins [70]	1999	1.2μ	2D (14x13)	Local	Token (ITI, FS)	Temporal edge
Indiveri [71]	1999	1.2μ	2D (8x8)	Global	Token (FS)	Temporal edge
Jiang [72]	1999	0.6μ	2D (32x32)	Global	Token (ISI)	Temporal edge of spatial contrast
Etienne-Cummings [51]	1999	2μ	2x1D (2x18)	Global	Adelson-Bergen	Spatiotemporal energy of edge map
Barrows [73]	2000	1.2μ	2x1D (2x4)	Global	Token (FS)	Spatial features
Liu [74]	2000	1.2μ	1D (1x37)	Global	Reichardt	Temporal contrast
Pant [75]	2000	1.6μ	2D (13x6)	Local	Reichardt	Temporal contrast
Higgins [76]	2000	1.2μ	2D (13x15)	Local	Token (FS)	Temporal edge
Harrison [77]	2000	1.2μ	1D (1x22)	Global	Reichardt	Temporal contrast
Higgins [78]	2002	1.2μ	2D (27x29)	Local	Token (ITI)	Temporal edge
Yamada [79]	2003	1.5μ	2D (2x10)	Local	Token (FS)	Spatial edge
Ozalevi [80]	2003	1.5μ	2D (6x6)	Global	Adelson-Bergen	Spatiotemporal energy
Massie [81]	2003	0.5μ	12x1D (12x90)	Yaw/Pitch/Roll	Token (FS)	Temporal edge of spatial contrast
Stocker [27]	2004	0.8μ	2D (30x30)	Local	Gradient	-
Ozalevi [82], [83]	2005	1.6μ	2D (6x7)	Global	Reichardt	Temporal contrast
Harrison [84]	2005	0.5μ	2D (16x16)	Global	Reichardt	Temporal contrast
Shoemaker [85]	2005	0.35μ	1D (1x7)	Global	Reichardt	Temporal contrast
Mehta [86]	2006	0.5μ	2D (95x52)	Local	Gradient	-
Moeckel [87]	2007	1.5μ	1D (1x24)	Local	Token (FS)	Temporal contrast
Bartolozzi [88]	2011	0.6μ	1D (1x64)	Local	Token (FS)	Temporal edge
Roubieu [89]	2013	-	1D (1x5)	Global	Token (ISI)	Temporal contrast

mouse). Pixels computed motion locally, and local estimates were combined to estimate global motion.

Moini *et al.* [67], [68] later presented a block matching chip which consisted of two orthogonal 1-D arrays and used spatio–temporal templates as the features to be matched.

2) Hassenstein–Reichardt and Barlow–Levick Models: At a high level, Tanner's implementation can be seen as sequentially implementing Reichardt detectors with increasing time delays until sufficient correlation is detected. In 1991, Andreou *et al.* [53] reported an analog implementation of the Reichardt detector which instead outputs the correlation value itself. The sensor computes ON-center OFF-surround features in analog and uses an all-pass filter to implement the delay. Outputs from all the Reichardt detectors along a linear array are summed and output as a differential current.

Delbruck [57] later presented a 2-D analog variation of the Reichardt detector. Temporal contrast was used as the input signal and was delayed using a multitap analog delay line capable of propagating a signal across multiple pixels. At each pixel, the correlation is computed between the propagating signal and the local signal before being propagated to the next pixel. The output, therefore, incorporates signals from many pixels and increases if motion is sustained across multiple pixels. However, the magnitude of the output is highly dependent on contrast.

Harrison and Koch [69] presented another analog Reichardt implementation which also computes correlation based on temporal contrast, but exhibits decreased dependence on contrast magnitude. The detectors in the chip are either accessed individually, or combined using a linear summation to obtain a global response. Harrison and Koch also used the approach to generate an artificial optomotor response (torque signal) and compared it to measurement of *Drosophila* in experiments [77].

Pant and Higgins [75] presented an analog Reichardt implementation in which the responses of multiple Reichardt detectors are combined ON-chip in a nonlinear fashion. Pant and Higgins showed how the output can be used to generate a torque signal to control the gaze direction of a robot during visual tracking, even though adjusting the gaze of the robot induces optical flow through egomotion (1).

Liu [74] also presented an analog Reichardt implementation with nonlinear ON-chip integration of detector outputs, and showed that the frequency response of the chip is similar to the frequency response of horizontal system (HS) neurons in *Drosophila* [43].

Shoemaker and O'Carroll [85] presented an analog Reichardt implementation based on visual motion processing in the fly. The sensor makes use of nonlinearities in the processing chain to reduce the dependence of the sensor output on scene contrast and spatial frequencies.

Harrison [84] later presented a time-to-contact sensor for collision avoidance based on detecting 2-D looming motion fields with Reichardt detectors.

Although partially addressed in the latter of the works mentioned thus far, one of the shortcomings of the Reichardt detector is the dependence of the output on stimulus contrast. This can be overcome by using so-called "token" methods, where thresholding provides a 1-b token to signal the presence or absence of a stimulus.

3) Token Methods: Horiuchi *et al.* [55], [56] developed a linear array which used detection of a sufficient temporal intensity derivative (edge) as a digital token. Tokens from neighboring pixels propagate down a digital delay line in opposite directions until they cross. The location in the delay lines at which they cross indicates both speed and direction. The delay line between each pixel pair contributes a "vote" into a winner-take-all circuit which outputs the location with the most votes.

Etienne-Cummings *et al.* [54] developed a chip for 2-D motion detection which uses the temporal derivative of a center-surround feature as the token. The chip encoded speed using the width of a pulse initiated by departure of a token from one pixel and terminated by arrival of the token at either neighbor. A voting scheme was used to determine direction.

Sarpeshkar *et al.* [58], [59] also took a token approach in which tokens trigger digital pulses. The pulse from one pixel would be delayed and correlated against its neighbor. The output provided a measure of how well the observed stimulus matched the circuit's optimal stimulus, but the response to non-optimal is ambiguous because both increasing and decreasing the stimulus speed cause the response to decrease. In the same paper, Sarpeshkar *et al.* proposed a facilitate-and-trigger approach to overcome the speed ambiguity.

Similar approaches to that proposed by Sarpeshkar *et al.* soon became popular. Fig. 6 outlines some of these token methods. The trigger-and-inhibit (TI) mechanism [Fig. 6(b)] triggers a pulse when a token is detected at a pixel, and ends (inhibits) the pulse when the token is detected at the next pixel, thereby providing a pulse with width inversely proportional to speed. However, for motion in the null direction, the inhibition occurs before the trigger, and the pulse can continue indefinitely. The shortest of two pulses from directionally opposing circuits is typically assumed to be the correct one.

The facilitate-and-trigger (FT) method [Fig. 6(c)] only triggers a pulse if a facilitation signal generated by the previous pixel is present. The pulse ends when the facilitation signal ends, thereby limiting the maximum pulse width to the width of the facilitation pulse. For motion in the null direction, the trigger occurs before the facilitation signal and no pulse is generated. Unlike the TI case, the pulse from the FT approach is directly proportional to speed.

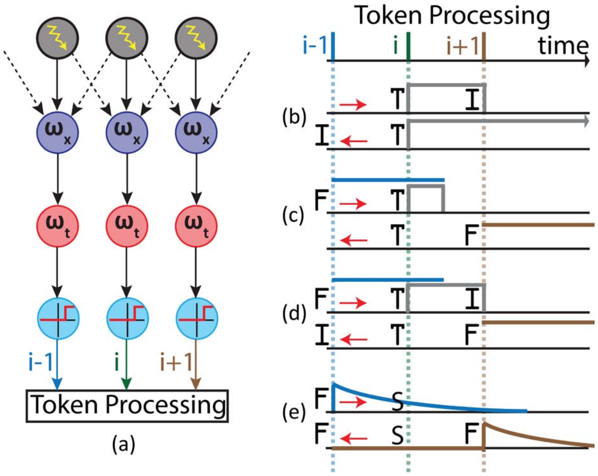

Fig. 6. ***Popular token methods. The left image (a) shows how tokens are typically generated by thresholding (light blue) the output of a temporal bandpass filter (red) to detect changes in pixel intensities. Spatial filtering (blue) is optionally performed as a first step. The trigger-and-inhibit (TI), facilitate-and-trigger (FT), facilitate–trigger–inhibit (FTI), and facilitate-and-sample (FS) methods are shown in (b)–(e), respectively, for a rightward moving stimulus, which will trigger responses from left (first) to right (last) when moving across the array. Red arrows indicate preferred direction of motion in each case.***

The facilitate–trigger–inhibit (FTI) method [Fig. 6(d)] combines the outputs of three pixels, using a facilitation signal for directional selectivity, followed by a TI mechanism to generate a pulse inversely proportional to the stimulus speed.

In the TI, FT, and FTI methods [Fig. 6(b)–(d)], the duration of a pulse must be measured in order to infer speed. The facilitate-and-sample (FS) method [Fig. 6(e)] overcomes this by using a shaped facilitation pulse. Instead of triggering an output pulse, the facilitation pulse is sampled providing a value proportional to speed. For motion in the null direction, the facilitation signal will be zero when it is sampled.

Other methods have been proposed which take a similar approach, but rely on inhibition to suppress response in the null direction rather than facilitation to enable response in the preferred direction.

Kramer [61] and Krammer and Koch [62] presented the first FTI and the first FS token implementations, each using an 8-pixel linear array with temporal contrast edges as the token.

Etienne-Cummings *et al.* [65], [66] developed a foveated sensor for tracking and stabilization, consisting of a 19 × 17-pixel array for detecting onset and offset of spatial edges, with the middle 5 × 5 pixels replaced by a 9 × 9 array of smaller motion estimating pixels which output motion direction only, thereby realizing a "bang–bang" output.

Higgins *et al.* [70] later presented the first inhibit–trigger–inhibit (ITI) implementation as well as the first 2-D FS implementation, both using temporal contrast edges for the token. His FS implementation also subtracted the samples of opposite direction circuits ON-chip to provide a signed velocity. They later further developed the concept and extended it to larger array [78]. Jiang and Wu [72] meanwhile presented the first 2-D FTI implementation, using temporal edges of spatial contrast as the token, while Yamada and Soga [79] demonstrated an FS token implementation using 1-D arrays and reported on its possible application to traffic flow measurement and monitoring blind corners while driving.

As neuromorphic front–end sensors for temporal contrast detection improved, Higgins and Koch [76] and Indiveri *et al.* [71] adopted multichip approaches to motion estimation, relying on a standalone specialized front–end sensor for temporal contrast detection, and a separate chip for the FS token algorithm implementation. The multichip approach carries the disadvantage of requiring additional power for OFF-chip communication between the front–end sensor and the motion computation chip. However, moving the in-pixel motion computation circuits to a separate chip reduces pixel size on the front–end chip, allowing a denser pixel array at the front–end.

Barrows and Neely [73] implemented a multichip token method which combined a microcontroller for post-processing with a front–end sensor for extracting programmable spatial features. Different spatial features were found to work well for different stimuli, and the combined use of multiple spatial features was used to improve performance. The chips were used to control the rudder of a microaerial vehicle (MAV) to help it avoid obstacles.

The multichip approach also allows signals to be remapped between the front–end sensor and back–end motion computation chip, allowing mapping from Cartesian to polar coordinates, which can be useful for measuring looming motion fields as Indiveri *et al.* [71] and Higgins and Koch [76] both demonstrated. Furthermore, signals can easily be copied and routed to multiple motion processing chips. Higgins and Koch demonstrated such an approach for simultaneously computing motion in Cartesian and polar coordinates using two motion processing chips in parallel. Higgins and Koch also demonstrated how two front–end sensors can feed into a single motion processing chip to compute motion only at a specific disparity (depth in the *z*-direction) [76].

Massie *et al.* [81] presented a combination imager and motion estimation chip for roll, pitch, and yaw estimation. The chip consists of 12 linear 90 pixel arrays (2 for yaw, 2 for pitch, and 8 for roll) relying on the token-based FS method. Integrated into the same chip was a 128 × 128 pixel variable acuity imager capable of providing maximum resolution on objects of interest, while conserving bandwidth by combining pixel responses from "uninteresting" regions.

Ozalevi *et al.* [82], [83] presented a multichip approach which used a separate front–end sensor to generate temporal edge tokens, but a low-pass filter was used to convert these tokens back into analog signals which were processed by separate chips implementing Hassenstein–Reichardt and Barlow–Levick models. The low-pass filter also serves to create the delays required by these models (see Fig. 5). Thus, an analog implementation of the Hassenstein–Reichardt and Barlow–Levick models is realized, but the intermediate "token" stage serves to normalize signal amplitude, largely removing the dependence on stimulus contrast.

Moeckel and Liu [87] presented a linear array relying on the FS token method with improved robustness to noise allowing the chip to extract motion over two decades of speeds.

Bartolozzi *et al.* [88] recently presented a prototype linear array motion tracking chip which relies on the FS token method using temporal contrast edges as the token. Temporal derivatives are computed as part of the token generation process, but the chip also computes spatial derivatives in parallel. Both these derivatives as well as the motion estimates themselves are fed into a winner-take-all (WTA) circuit with programmable input weights, allowing the user to track the most salient feature in the array. The programmability of the WTA allows the most salient feature to be defined as a weighted summation of the spatial contrast, temporal contrast, and motion features.

Roubieu *et al.* [89] presented a 23.3-mm × 12.3-mm sensor weighing under 1 g (including optics). The sensor consists of five pairs of 1-D motion sensors which use tokens to measure the time for a feature to travel between neighboring locations, similar to the TI token method.

Orchard *et al.* [91] proposed an algorithm in which simple spiking neurons with preprogrammed synaptic delays can be combined with a silicon retina [13] to implement motion-sensitive receptive fields. Similarly to Fig. 3(a), where a point with motion in one spatial dimension traces out a line in a 2-D space-time plot, and an edge with motion in two spatial dimensions will trace out plane in a 3-D space-time plot, with the slope of the plane encoding the local motion velocity. This can be seen as a Reichardt detector, where the interpixel delays uniquely describe a plane, although the algorithm still computes on temporal contrast tokens provided by the front–end sensor. This approach is elaborated on in Section VI.

In the authors' implementation, each neuron is designed to detect the presence of a particular local space-time plane (i.e., a specific interpixel delay) but combines responses from 5 × 5 pixels per motion unit to simultaneously determine both the direction and speed of the normal flow. The responses of multiple motion units are then combined in a second layer of the neural network to attenuate errors due to the aperture problem. However, this algorithm has not yet been implemented on embedded hardware or in real time.

At a similar time, Benosman *et al.* [92] proposed an approach which also relies on detection of local planes in data provided by a silicon retina. Instead of using multiple receptive fields tuned to detect different motions (planes), the best fit for a single local plane is mathematically computed, with the normal of the computed fit indicating the normal flow locally. The algorithm runs in JAVA on a host computer.

C. Frequency Methods

Others have focused on Adelson–Bergen-type models [24] relying on spatio–temporal filtering for motion detection. The first such implementation was reported by Etienne-Cummings *et al.* [51], using a front–end silicon retina to compute a binary map of spatial edges, thereby providing an input signal of normalized amplitude. Subsequent processing using a multichip reconfigurable neural processor implements pairs of spatial and temporal filters to extract the oriented energy at a particular spatio–temporal frequency, thereby implementing an Adelson–Bergen motion unit. The oriented energies from multiple motion units of different frequencies are computed in parallel and their outputs are combined (as described in [51]) to obtain an estimate of image motion.

Ozalevi and Higgins [80], in an approach similar to his implementation of the Hassenstein–Reichardt and Barlow–Levick models, described a multichip Adelson–Bergen model. Although successful, this model only implemented a single motion energy unit per pixel, therefore indicating the presence of a preferred motion stimulus and direction without indicating speed.

Modern computing technologies allow for processing on a larger scale than ever before. Orchard *et al.* [93] demonstrated how an FPGA can be used to implement and combine 720 Adelson–Bergen motion energy units per pixel in real time for a 128 × 128 pixel array running at 30 frames per seconds (fps).

VI. A SPIKING NEURAL NETWORK FOR VISUAL MOTION ESTIMATION

As mentioned in Section V, the authors have developed a spiking neural network architecture for visual motion estimation [91] which relies on synaptic delays to create motion-sensitive receptive fields.

Discrete temporal contrast tokens from a separate front–end sensor [13] are used as input spikes to the architecture, and LIF neurons with a linear decay are used for computation. Such neurons are good at detecting temporal coincidence of their inputs, but motion signals are inherently spread over time, as modeled in (5).

As stated in [91] and repeated here for convenience, a motion-sensitive unit in the architecture relies on the assumption that if we consider a small enough spatial region, a moving edge can be approximated as being a straight edge moving with constant velocity. The following equation shows how a motion stimulus is modeled:

$$\begin{aligned} I(x,y,t) &= H(x - v_x t)\\ \frac{dI(x,y,t)}{dt} &= \delta\left(t - \frac{x}{v_x}\right)\\ E(x,y,t) &= \delta\left(t - \frac{x}{v_x}\right) III_1(x) III_1(y) \end{aligned} \tag{5}$$

where x and y describe a location on the image plane. I is intensity, t is time (milliseconds), H is the Heaviside step function, v_x is the x-component of velocity in pixels per millisecond, δ is the Dirac delta function, $dI(x,y,t)/dt$ is the temporal derivative of image intensity, $E(x,y,t)$ is the sensor output, and III_1 is a sampling comb with period 1 pixel. Multiplying by the sampling combs converts the continuous space signal into a discrete space signal which only has values at integer pixel locations.

Fig. 7 shows how a receptive field sensitive to a specific motion stimulus (in this case a speed of 1/5 pixels per millisecond in the x-direction) can be constructed. The underlying concept relies on using synaptic delays (green arrows) to convert a temporal sequence of spikes (red crosses) into a group of spikes coincident in time (green plane). The delayed spikes serve as input to a LIF neuron, which is good at detecting temporal coincidence of its inputs. In practice, there will not be perfect temporal coincidence because the actual spikes received from the

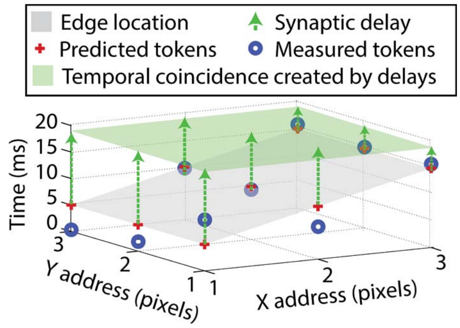

Fig. 7. ***Construction of a 3 × 3 pixel receptive field sensitive to motion of an edge parallel to the y-axis traveling in the positive x-direction with speed $v_x = 1/5$ pixels per millisecond. The location of the edge can be described by $x = v_x t$, where x is measured in pixels and t is measured in milliseconds. This equation describes a spatio–temporal plane (shown in gray). Red crosses are located where the plane crosses pixel locations and indicate which pixels are expected to respond when (blue circles indicate actual recorded data). The length of green arrows above each pixel location indicates the synaptic delay for the synapse connecting from that pixel, and the green plane indicates the time at which the neuron would respond to this stimulus.***

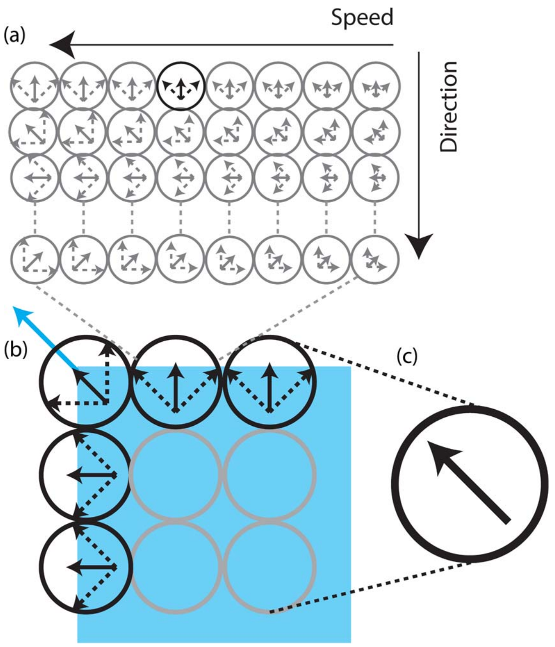

Fig. 8. ***Multilayer architecture detecting the motion of a blue box. (a) A set of neurons tuned to detect different speeds and directions of motion. A full set of such neurons is present at every image location. (b) Multiple image locations as they detect the motion of a stimulus (blue box moving upwards to the left). Black circles show locations of activated neurons, with arrows indicating which neuron (speed and direction) was activated. (c) Layer 2 neuron which determines the correct motion by combining layer 1 outputs to alleviate the aperture problem.***

front–end sensor (blue circles) will not perfectly match the spike times predicted by our model (red crosses).

Lowering the threshold voltage of the LIF neuron will cause it to still respond when its inputs are slightly spread in time, and the threshold value can be used to control how much time spreading can be tolerated before the neuron stops responding.

Our approach can be seen as a Reichardt detector covering multiple pixels, since it is effectively delaying the signal from neighboring pixels while the LIF neuron detects multipixel correlations in the delayed spikes.

As with the Reichardt detector, multiple detectors (in our case neurons) are required in order to detect different speeds and directions of motion. Our architecture uses 8 × 8 neurons per pixel location to detect all possible combinations of eight different directions and eight different speeds, as shown in Fig. 8(a). Directions vary from 0° to 315° in steps of 45°, while speeds vary from $\sqrt{2}/50$ to $\sqrt{2}^8/50$ pixels/ms by factors of $\sqrt{2}$.

Equations (5) are independent of motion parallel to the edge direction, presenting a form of the aperture problem where only motion perpendicular to an edge (the normal flow) can be detected. An edge moving in a direction perpendicular to its orientation at speed s would look identical to the same edge moving in a direction 45° to its orientation with speed $s\sqrt{2}$ (see dotted arrows in Figs. 2 and 8) since, in both cases, the perpendicular component of motion is just s. This relationship gives rise to the $\sqrt{2}$ factor used between different speeds.

A key feature which sets this work apart from previous token and Reichardt works is the use of a second stage of processing to overcome the aperture problem. The second stage of processing is implemented by a second layer of neurons, with each neuron receiving inputs over a wider spatial region than first layer neurons [Fig. 8(c)]. A layer 2 neuron sensitive to speed s and direction d would incorporate inputs from layer 1 neurons sensitive to the same speed and direction, but also from layer 1 neurons sensitive to speed $s/\sqrt{2}$ and directions $d \pm 45°$. This multilayer approach bears resemblance to gradient-based methods such as Lucas–Kanade [22] and Horn–Schunck [23], which compute normal flow locally in a first step before incorporating the normal flow from other nearby locations to more accurately approximate the true optical flow.

Fig. 8 shows the system architecture. At each pixel location, there are 8 × 8 neurons sensitive to different speeds and directions of motion [Fig. 8(a)], but mutual inhibition ensures that only one neuron (shown in black) can respond to a stimulus (produce an output spike) at any time.

Fig. 8(b) shows a stimulus covering 3 × 3 pixel locations. Dark circles indicate locations where neurons are responding, with solid arrows indicating the speed and direction selectivity of the neuron responding at that location. Dotted arrows indicate other speeds and directions consistent with the aperture problem as discussed above. Fig. 8(c) shows a layer 2 neuron determining the correct motion from the inputs it receives from the layer 1 neurons of Fig. 8(b).

Fig. 9 shows actual outputs from each layer for a real-world scene of a bus crossing a bridge. Layer 1 outputs tend

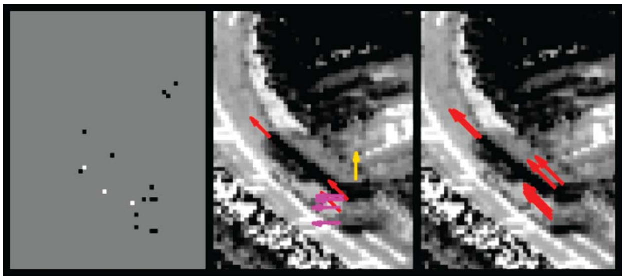

Fig. 9. ***Motion of a moving bus as detected by the network. The left pane shows temporal change tokens (black for decrease, white for increase, gray for no change). The middle pane shows outputs of layer 1, which tend to be perpendicular to edges. The right pane shows the output of layer 2, which uses the data from layer 1 to detect the actual flow. In each pane, only 3.3 ms worth of data is shown. Grayscale values are obtained using the exposure measurement function of the asynchronous time-based image sensor (ATIS) [13].***

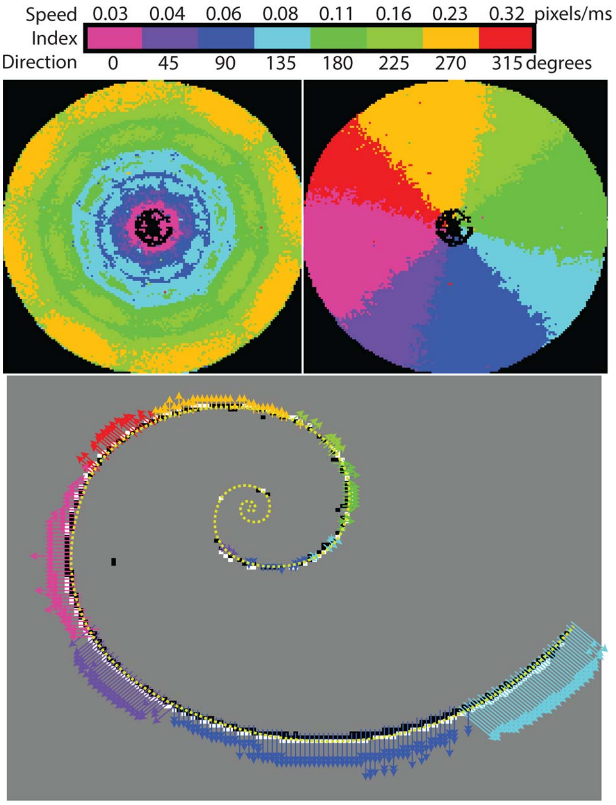

Fig. 10. ***Output of the architecture when viewing a rotating spiral. Top images show speed (left) and direction (right) outputs accumulated over multiple rotations. The lower image shows 10 ms of output data while the spiral is spinning. The motion vectors all point outward from the center, creating a looming field.***

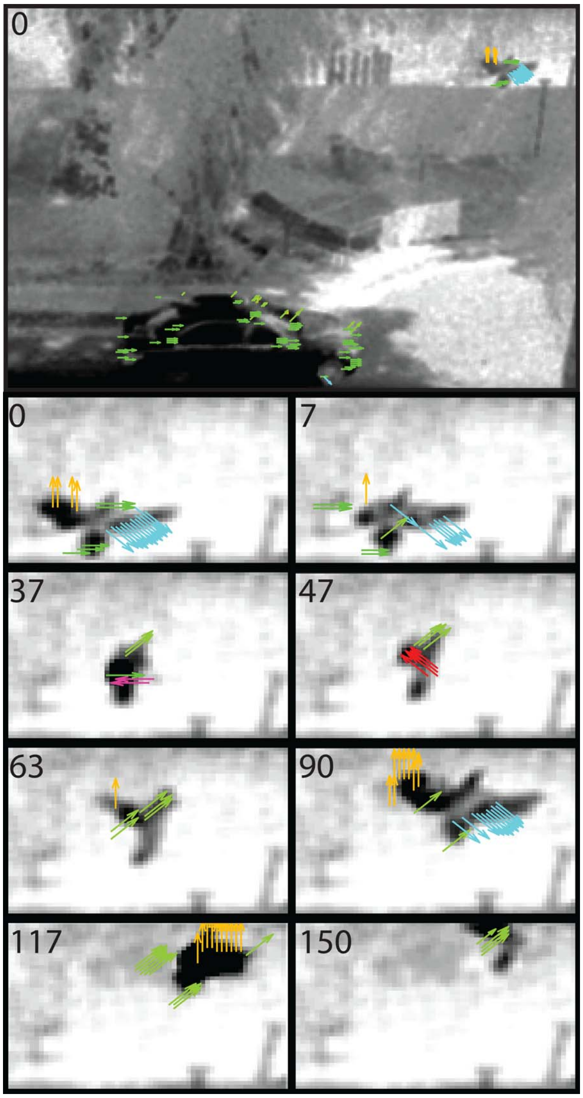

Fig. 11. ***Example real-world scene (top) consisting of a moving car and a startled bird. The lower subimages show a cropped region around the bird during flight at different points in time. The time in milliseconds is indicated at the top left of each image. Notably, the motion of each wing is detected as well as the motion of the body.***

to be perpendicular to edges, while layer 2 outputs more accurately describe the actual motion of the bus by incorporating data over a larger spatial region.

Fig. 10 presents the output of the architecture for a controlled stimulus consisting of a spinning spiral. The figure shows how it can reliably detect different speeds and directions of motion. The top images show data accumulated over multiple rotations of the spiral. Color is used to encode speed (left) and direction (right) according to the legend provided above the images. Speed varies as a function of distance to the axis of rotation, while direction varies with angle. Near the axis of rotation, the motion is slower than the slowest receptive field and therefore elicits no responses. The lower image shows the motion architecture's output accumulated over a period of 10 ms. The color of arrows helps to encode the motion direction, while their length encodes speed. The spiral stimulus used has been superimposed on the image as a yellow dashed line.

Fig. 11 shows another example of the architecture operating on real-world data. The top part of the image shows a full scene captured with the ATIS. There are two moving objects in the scene: a rightward moving car in the lower part and a bird in the top right. The rest of the images in Fig. 11 show motion responses elicited by the flying bird. Inset numbers indicate the time (in milliseconds) at which each image was captured. Each image shows 3.3 ms of motion output data. In the first frame ($t = 0$), rightward motion of the bird's body is detected, while separate motion is detected for each wing. In the second frame ($t = 7$ ms) retraction of the left wing is detected, followed by retraction of the right wing at 37 ms, as shown by the opposite motion estimates for each wing while the body continues motion upward to the right. Ten milliseconds later extension of the left wing is detected (red arrows at $t = 47$ ms). By 90 ms, the bird has returned to a similar pose to that seen at 0 ms. At 117 ms, both wings have been pulled in front of the body, causing motion upwards. After 150 ms, the bird exits the scene.

The bird example is particularly tricky for conventional motion estimation techniques due to the bird significantly and rapidly changing its appearance while moving.

The model as presented here has not yet been integrated into a real-time implementation. However, the model is computationally efficient. Neurons are only updated when an input spike arrives, and each neuron update only requires four additions, two greater-than comparisons, and one multiplication. Sparsity of the incoming spikes from the ATIS front–end sensor keeps the required number of operations per second low and easily achievable with commercially available hardware.

The challenge in achieving real-time implementation does not lie in the computation rate, but rather in the implementation of synaptic delays. Incoming spikes must be delayed and stored, which drives up memory requirements. These memory requirements can be reduced by observing that only a small percentage of synapses are active at any one time, so memory can be shared between synapses. However, different synapses have different delays, so a simple first-in–first-out (FIFO) buffer will not work because the order in which spikes are placed in the buffer will not be the same as the order in which they must be read out.

VII. FUTURE DIRECTIONS

In the previous sections, we have summarized past and present works. In this section, we outline the directions for future works.

Although great progress has been made, the organization of computational circuits in artificial approaches still differs significantly from biology. In the retina, neurons are arranged in interconnected layers stacked on top of each other and lying above the photoreceptor layer. Similarly, in visual cortex, neurons are arranged in 3-D layers, with both short local connections, and longer range axonal connections between more distant regions of cortex.

In silicon, photoreceptors and computational circuits mimicking different layers of biological processing are restricted to lying side by side within the same plane, which limits both the photoreceptor size and spacing, which in turn affects the signal strength and spatial resolution, respectively. When mapping a 3-D biological structure onto 2-D silicon, short vertical connections are often mapped to long lateral connections, increasing line capacitance, energy consumption, and occupying valuable space. This is overcome in many artificial implementations by only considering motion in one lateral direction, then stacking circuits in the other lateral direction instead of vertically [3], [53], [55], [58], [62], [63], [74], [77], [88], [94].

As 3-D stacked silicon technology matures, it can be used to alleviate the wiring problem, allow for larger photodiode fill factors, and achieve a more biologically realistic organization of neural circuits. This compact 3-D organization is typical of insect vision and primate retina, however, the early stages of primate retina and visual cortex are located far from each other, and thus compact integration of cortical circuits and photoreceptors is not necessarily accurate to biology.

Some of the described works have relied on an approach in which a spiking "silicon retina" and neural processing are implemented in separate chips [12], [13] (although 3-D integration is useful for both chips). Implementing retinal and cortical processing as two different components provides advantages during system development. First, an improvement in either component can be achieved without refabricating the other, and second, data can be recorded as it is transmitted between the two, allowing for in-depth offline analysis which can provide insights into how neural algorithms for processing can be further improved.

The last decade has seen silicon retinae mature to the point where they are now commercially available and are used by many labs around the world, and there are a number of works emerging (see [95] and [96]) which argue for the superiority of these sensors for high-speed visual tasks which must be executed in real time on a limited power budget. Reconfigurable neural processing platforms are also rapidly maturing, spurred by the dramatic increase in interest and funding the neuromorphic field has experienced in the last few years.

It is an exciting time for the neuromorphic area, with major companies including Qualcomm, Samsung, Intel, and IBM coming onboard and launching their own research projects in the field. There are also major projects in the United States (the $200 million BRAIN initiative [97]) and Europe (the €1 billion Human Brain Project [98]) which are incorporating neuromorphic aspects. The U.S. Defense Advanced Research Projects Agency (DARPA) has also been taking notice, funding projects such as the Unconventional Processing Of Signals For Intelligent Data Exploitation (UPSIDE) and Systems of Neuromorphic Adaptive Plastic Scalable Electronics (SYNAPSE) projects.

Modern CMOS technology is quite different from biological "wetware." CMOS typically operates at frequencies ranging from megahertz to gigahertz, while a general rule of thumb is that biological neurons do not operate at frequencies above 1 kHz. This massive speed difference is not necessarily an advantage for silicon. In fact, slowing silicon neural circuits down to biologically realistic time scales can prove quite challenging, and often requires extra design effort and cost to implement.

Biology leverages parallel processing, and the speed of CMOS can be useful when one wants to approximate multiple parallel units from biology using a single high-speed sequential unit in silicon. However, this approach comes at a disproportionate power cost. Higher operating speeds require higher operating voltages, and power scales proportionally to voltage squared. It is, therefore, preferable to have many low-speed, low-voltage processors (like biology) than a few high-speed, high-voltage processors (like modern CMOS). Hence, biology provides a roadmap for the future, where the scaling of CMOS will allow the realization of ultralow-voltage (hence low-power) circuits performing massively parallel computation in very small and 3-D stacked dies. As technology moves in this direction, CMOS can learn about 3-D connectivity, massively parallel computation, density of computational elements, and stochastic circuits from biology.

Despite technological improvements, wiring remains an issue in the connectivity which can currently be achieved between artificial neurons. Even though 3-D integration can help, interneuron connectivity with present technologies is constrained to remain far sparser than in biology. The combined use of silicon circuits and carbon nanotube crossbar arrays has been proposed to improve physical connectivity, with memristor devices capable of learning proposed for use as synaptic connections between nanotubes [99].

The development of new online learning algorithms and architectures, whether relying on memristor devices or conventional silicon, are likely to play an increasingly important role. Using learning through visual experience to help configure and organize a neural architecture can improve fault tolerance (and, therefore, device yield) and save man hours spent on manual configuration. This learning is especially important if copies of the same device are to adapt to operation under very different visual conditions, such as in urban versus forested environments, or onboard flying versus ground vehicles.

As mentioned in the Introduction, biological sensors are embodied, and have evolved in conjunction with motor systems. The interplay between motor and sensory systems can be useful for sensing. Examples of this include the peering behavior used by many animals to induce motion parallax for depth perception [100], the optomotor response in insects [101], and the vestibular ocular reflex in humans [40]. Recent studies also suggest that microsaccades during fixation play an important role in perception, particularly for object recognition in humans [102]. In *Drosophila*, motion is found to also play a role in motion perception [52].

An embodied biological sensor also serves a particular purpose, to provide information relevant to the agent for self-preservation and meaningful interaction with the environment. The value metric of biological motion estimates is, therefore, not directly assessed by how accurately motion is perceived, but rather by how motion estimates improve the effectiveness of the agent's behavior (although to a degree, more accurate motion estimates will be more effective in affecting behavior).

It is, therefore, important to keep in mind the intended use of the system being constructed. Inspiration from biology is useful, but at some stage, the design must deviate from precise biomimicry. A MAV may benefit from an artificial version of the *Drosophila* vision system, but for the vehicle to be of value to the operator, it will be expected to execute a goal-oriented task rather than simply behave like *Drosophila*. Also, at some point, making a system more biologically accurate will come at a performance cost rather than benefit due to the inherent differences between silicon circuits and biological neurons. It was Carver Mead who first developed the concept of imitating neural processing in silicon circuits by noting the similarities between the two [103], but it was also Carver Mead who said "Listen to the technology; find out what it's telling you."

Nevertheless, we are still a long way from matching the power efficient performance of biology in artificial systems, so for the foreseeable future, continued research into bioinspired visual motion estimation techniques will reap rewards for artificial systems. ■

REFERENCES

[1] D. N. Lee and H. Kalmus, "The optic flow field: The foundation of vision," *Philosoph. Trans. Roy. Soc. Lond. B, Biol. Sci.*, vol. 290, no. 1038, pp. 169–179, 1980.

[2] M. V. Srinivasan and S. Zhang, "Visual motor computations in insects," *Annu. Rev. Neurosci.*, vol. 27, pp. 679–696, 2004.

[3] J. E. Tanner, "Integrated optical motion detection," Ph.D. dissertation, Eng. Appl. Sci., California Inst. Technol., Pasadena, CA, USA, 1986.

[4] D. Wolpert, "The real reason for brains," *TED Talks*, Jul. 2011.

[5] A. Hyvrinen, J. Hurri, and P. O. Hoyer, *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*, 1st ed. New York, NY, USA: Springer-Verlag, 2009.

[6] M. E. Adan, T. Aoyagi, T. E. Holmdahl, T. M. Lipscomb, and T. Miura, "Operator input device," U.S. Patent 6 172 354, Jan. 2001.

[7] G. B. Gordon, D. L. Knee, R. Badyal, and J. T. Hartlove, "Seeing eye mouse for a computer system," U.S. Patent 6 433 780, Aug. 2002.

[8] C. A. Mead and J. E. Tanner, "Correlating optical motion detector," U.S. Patent 4 631 400, Dec. 1986.

[9] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, "An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 1736–1741.

[10] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Curr. Opinion Neurobiol.*, vol. 14, no. 4, pp. 481–487, 2004.

[11] N. K. Logothetis and B. A. Wandell, "Interpreting the BOLD signal," *Annu. Rev. Physiol.*, vol. 66, pp. 735–769, 2004.

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

[13] 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.

[14] B. V. Benjamin *et al.*, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.

[15] S. Panzeri, N. Brunel, N. K. Logothetis, and C. Kayser, "Sensory neural codes using multiplexed temporal scales," *Trends Neurosci.*, vol. 33, no. 3, pp. 111–120, 2010.

[16] E. Painkras *et al.*, "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, Aug. 2013.

[17] G. A. Silva, "Neuroscience nanotechnology: Progress, opportunities and challenges," *Nature Rev. Neurosci.*, vol. 7, no. 1, pp. 65–74, 2006.

[18] T. N. Wiesel and D. H. Hubel, "Single-cell responses in striate cortex of kittens deprived of vision in one eye," *J. Neurophysiol.*, vol. 26, no. 6, pp. 1003–1017, 1963.

[19] T. N. Wiesel, "The postnatal development of the visual cortex and the influence of environment," *Biosci. Rep.*, vol. 2, no. 6, pp. 351–377, 1982.

[20] B. Jahne, *Computer Vision and Applications: A Guide for Students and Practitioners*. New York, NY, USA: Academic, 2000.

[21] K. Nakayama and G. H. Silverman, "The aperture problem. perception of nonrigidity and motion direction in translating sinusoidal lines," *Vis. Res.*, vol. 28, no. 6, pp. 739–746, 1988.

[22] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. DARPA Image Understand. Workshop*, Apr. 1981, pp. 121–130.

[23] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–203, 1981.

[24] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Amer.*, vol. 2, no. 2, pp. 284–299, 1985.

[25] P. A. M. Dirac, *The Principles of Quantum Mechanics*. Oxford, U.K.: Oxford Univ. Press, 1981.

[26] R. N. Bracewell, *The Fourier Transform and Its Applications*. New York, NY, USA: McGraw-Hill, 1980.

[27] A. Stocker, "Analog integrated 2-D optical flow sensor," *Analog Integr. Circuits Signal Process.*, vol. 46, pp. 121–138, 2006.

[28] C. M. Higgins, V. Pant, and R. Deutschmann, "Analog VLSI implementation of spatio-temporal frequency tuned visual motion algorithms," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 3, pp. 489–502, Mar. 2005.

[29] D.-E. Nilsson, "Eye ancestry: Old genes for new eyes," *Curr. Biol.*, vol. 6, no. 1, pp. 39–42, 1996.

[30] M. F. Land and D.-E. Nilsson, *Animal Eyes*. Oxford, U.K.: Oxford Univ. Press, 2012.

[31] A. Borst, J. Haag, and D. F. Reiff, "Fly motion vision," *Annu. Rev. Neurosci.*, vol. 33, pp. 49–70, 2010.

[32] P. R. Martin and S. G. Solomon, "Information processing in the primate visual system," *J. Physiol.*, vol. 589, no. 1, pp. 29–31, 2011.

[33] M. Joesch, F. Weber, H. Eichner, and A. Borst, "Functional specialization of parallel motion detection circuits in the fly," *J. Neurosci.*, vol. 33, no. 3, pp. 902–905, 2013.

[34] W. R. Taylor and R. G. Smith, "The role of starburst amacrine cells in visual signal processing," *Vis. Neurosci.*, vol. 29, no. 1, pp. 73–81, 2012.

[35] M. Joesch, B. Schnell, S. V. Raghu, D. F. Reiff, and A. Borst, "On and off pathways in Drosophila motion vision," *Nature*, vol. 468, no. 7321, pp. 300–304, 2010.

[36] W. Reichardt, "Autocorrelation, a principle for the evaluation of sensory information by the central nervous system," *Sensory Communication*. Cambridge, MA, USA: MIT Press, 1961, pp. 303–317.

[37] J. H. Maunsell and D. C. Van Essen, "Functional properties of neurons in middle temporal visual area of the macaque monkey," *J. Neurophysiol.*, vol. 49, no. 5, pp. 1127–1147, 1983.

[38] W. T. Newsome and E. B. Pare, "A selective impairment of motion perception following lesions of the middle temporal visual area (MT)," *J. Neurosci.*, vol. 8, no. 6, pp. 2201–2211, 1988.

[39] A. Borst and T. Euler, "Seeing things in motion: Models, circuits, mechanisms," *Neuron*, vol. 71, no. 6, pp. 974–994, 2011.

[40] D. E. Angelaki and K. E. Cullen, "Vestibular system: The many facets of a multimodal sense," *Annu. Rev. Neurosci.*, vol. 31, pp. 125–150, 2008.

[41] B. G. Breitmeyer and L. Ganz, "Implications of sustained and transient channels for theories of visual pattern masking, saccadic suppression, information processing," *Psychol. Rev.*, vol. 83, no. 1, pp. 1–36, 1976.

[42] A. D. Milner, M. A. Goodale, and A. J. Vingrys, *The Visual Brain in Action*. Oxford, U.K.: Oxford Univ. Press, 2006. vol. 2.

[43] K. Hausen, "Motion sensitive interneurons in the optomotor system of the fly," *Biol. Cybern.*, vol. 45, no. 2, pp. 143–156, 1982.

[44] D. H. Hubel and T. N. Weisel, "Receptive fields, binocular interaction, function architecture in the cat's visual cortex," *J. Physiol. (Lond.)*, vol. 160, pp. 106–154, 1962.

[45] H. B. Barlow and W. R. Levick, "The mechanism of directionally selective units in rabbit's retina," *J. Physiol.*, vol. 178, no. 3, p. 477, 1965.

[46] A. B. Watson and A. J. Ahumada, Jr., "Model of human visual-motion sensing," *J. Opt. Soc. Amer. A*, vol. 2, no. 2, pp. 322–341, 1985.

[47] C. Fermuller, H. Ji, and A. Kitaoka, "Illusory motion due to causal time filtering," *Vis. Res.*, vol. 50, no. 3, pp. 315–329, 2010.

[48] Y. Weiss, E. P. Simoncelli, and E. H. Adelson, "Motion illusions as optimal percepts," *Nature Neurosci.*, vol. 5, no. 6, pp. 598–604, 2002.

[49] E. P. Simoncelli and D. J. Heeger, "Representing retinal image speed in visual cortex," *Nature Neurosci.*, pp. 461–462, 2001.

[50] E. P. Simoncelli, "Local analysis of visual motion," *Vis. Neurosci.*, pp. 1616–1623, 2003.

[51] R. Etienne-Cummings, J. Van der Spiegel, and P. Mueller, "Hardware implementation of a visual-motion pixel using oriented spatiotemporal neural filters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 9, pp. 1121–1136, Sep. 1999.

[52] N. Franceschini, "Small brains, smart machines: From fly vision to robot vision and back again," *Proc. IEEE*, vol. 102, no. 5, pp. 751–781, May 2014.

[53] A. G. Andreou, K. Strohbehn, and R. E. Jenkins, "Silicon retina for motion computation," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 1991, vol. 3, pp. 1373–1376.

[54] R. Etienne-Cummings, S. A. Fernando, J. Van der Spiegel, and P. Mueller, "Real-time 2D analog motion detector VLSI circuit," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 1992, vol. 4, pp. 426–431.

[55] T. Horiuchi, J. Lazzaro, A. Moore, and C. Koch, "A delay-line based motion detection chip," *Adv. Neural Inf. Process. Syst.*, vol. 3, pp. 406–412, 1991.

[56] T. Horiuchi *et al.*, "Computing motion using analog VLSI vision chips: An experimental comparison among different approaches," *Int. J. Comput. Vis.*, vol. 8, no. 3, pp. 203–216, 1992.

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

[58] R. Sarpeshkar, W. Bair, and C. Koch, "Visual motion computation in analog VLSI using pulses," *Adv. Neural Inf. Process. Syst.*, vol. 5, pp. 781–788, 1993.

[59] R. Sarpeshkar, J. Kramer, G. Indiveri, and C. Koch, "Analog VLSI architectures for motion processing: From fundamental limits to system applications," *Proc. IEEE*, vol. 84, no. 7, pp. 969–987, Jul. 1996.

[60] M. Gottardi and W. Yang, "A CCD/CMOS image motion sensor," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, Feb. 1993, pp. 194–195.

[61] J. Kramer, "Compact integrated motion sensor with three-pixel interaction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 4, pp. 455–460, Apr. 1996.

[62] 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.

[63] A. J. S. Yakovleff and A. Moini, "Motion perception using analog VLSI," *Analog Integr. Circuits Signal Process.*, vol. 15, no. 2, pp. 183–200, 1998.

[64] X. Arreguit, F. A. Van Schaik, F. V. Bauduin, M. Bidiville, and E. Raeber, "A CMOS motion detector system for pointing devices," *IEEE J. Solid-State Circuits*, vol. 31, no. 12, pp. 1916–1921, Dec. 1996.

[65] R. Etinne-Cummings, J. Van der Spiegel, P. Mueller, and M.-Z. Zhang, "A foveated visual tracking chip," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, Feb. 1997, pp. 38–39.

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

[67] A. Moini *et al.*, "An insect vision-based motion detection chip," *IEEE J. Solid-State Circuits*, vol. 32, no. 2, pp. 279–284, Feb. 1997.

[68] A. Moini, *Vision Chips*. New York, NY, USA: Springer-Verlag, 2000.

[69] R. R. Harrison and C. Koch, "An analog VLSI model of the fly elementary motion detector," *Adv. Neural Inf. Process. Syst.*, vol. 10, pp. 880–886, 1998.

[70] C. M. Higgins, R. A. Deutschmann, and C. Koch, "Pulse-based 2-D motion sensors," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 6, pp. 677–687, Jun. 1999.

[71] G. Indiveri, A. M. Whatley, and J. Kramer, "A reconfigurable neuromorphic VLSI multi-chip system applied to visual motion computation," in *Proc. 7th Int. Conf. Microelectron. Neural Fuzzy Bio-inspired Syst.*, Los Alamitos, CA, USA, Apr. 1999, pp. 37–44.

[72] H.-C. Jiang and C.-Y. Wu, "A 2-D velocity- and direction-selective sensor with BJT-based silicon retina and temporal zero-crossing detector," *IEEE J. Solid-State Circuits*, vol. 34, no. 2, pp. 241–247, Feb. 1999.

[73] G. L. Barrows and C. Neely, "Mixed-mode VLSI optic flow sensors for in-flight control of a micro air vehicle," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 4109, pp. 52–63, 2000.

[74] S.-C. Liu, "A neuromorphic aVLSI model of global motion processing in the fly," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 12, pp. 1458–1467, Dec. 2000.

[75] V. Pant and C. M. Higgins, "A biomimetic VLSI architecture for small target tracking. In Circuits and Systems," in *Proc. Int. Symp. Circuits Syst.*, May 2004, vol. 3, pp. 5–8.

[76] C. M. Higgins and C. Koch, "A modular multi-chip neuromorphic architecture for real-time visual motion processing," *Analog Integr. Circuits Signal Process.*, vol. 24, no. 3, pp. 195–211, 2000.

[77] R. R. Harrison and C. Koch, "A robust analog VLSI Reichardt motion sensor," *Analog Integr. Circuits Signal Process.*, vol. 24, no. 3, pp. 213–229, 2000.

[78] C. M. Higgins and S. A. Shams, "A biologically inspired modular VLSI system for

visual measurement of self-motion," *IEEE Sensors J.*, vol. 2, no. 6, pp. 508–528, Dec. 2002.

[79] K. Yamada and M. Soga, "A compact integrated visual motion sensor for its applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 1, pp. 35–42, Mar. 2003.

[80] E. Ozalevli and C. M. Higgins, "Multi-chip implementation of a biomimetic VLSI vision sensor based on the Adelson-Bergen algorithm," in *Proc. Joint Int. Conf. Artif. Neural Netw. Neural Inf. Process.*, 2003, pp. 433–440.

[81] M. Massie, C. Baxter, J. P. Curzan, P. McCarley, and R. Etienne-Cummings, "Vision chip for navigating and controlling micro unmanned aerial vehicles," in *Proc. Int. Symp. Circuits Syst.*, May 2003, vol. 3, pp. 786–789.

[82] E. Ozalevli and C. M. Higgins, "Reconfigurable biologically inspired visual motion systems using modular neuromorphic VLSI chips," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 1, pp. 79–92, Jan. 2005.

[83] E. Ozalevli, P. Hasler, and C. M. Higgins, "Winner-take-all-based visual motion sensors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 8, pp. 717–721, Aug. 2006.

[84] R. R. Harrison, "A biologically inspired analog IC for visual collision detection," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 11, pp. 2308–2318, Nov. 2005.

[85] P. A. Shoemaker and D. C. O'Carroll, "Insect-based visual motion detection with contrast adaptation," *Defense and Security*. Philadelphia, PA, USA: SPIE, 2005, pp. 292–303.

[86] S. Mehta and R. Etienne-Cummings, "A simplified normal optical flow measurement CMOS camera," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 6, pp. 1223–1234, Jun. 2006.

[87] R. Moeckel and S.-C. Liu, "Motion detection circuits for a time-to-travel algorithm," in *IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 3079–3082.

[88] C. Bartolozzi, N. K. Mandloi, and G. Indiveri, "Attentive motion sensor for mobile robotic applications," in *IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 2813–2816.

[89] F. L. Roubieu, F. Expert, G. Sabiron, and F. Ruffier, "Two-directional 1-g visual motion sensor inspired by the fly's eye," *IEEE Sensors J.*, vol. 13, no. 3, pp. 1025–1035, Mar. 2013.

[90] A. A. Stocker, *Analog VLSI Circuits for the Perception of Visual Motion*. New York, NY, USA: Wiley, 2006.

[91] G. Orchard, R. Benosman, R. Etienne-Cummings, and N. V. Thakor, "A spiking neural network architecture for visual motion estimation," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2013, pp. 298–301.

[92] 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.

[93] G. Orchard, N. V. Thakor, and R. Etienne-Cummings, "Real-time motion estimation using spatiotemporal filtering in FPGA," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2013, pp. 306–309.

[94] R. Moeckel, R. Jaeggi, and S.-C. Liu, "Steering with an aVLSI motion detection chip," in *IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 1036–1039.

[95] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, "Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output," *Proc. IEEE*, vol. 102, no. 10, Oct. 2014, DOI: 10.1109/JPROC.2014.2346153.

[96] F. Barranco, C. Fermuller, and Y. Aloimonos, "Contour motion estimation for asynchronous event-driven cameras," *Proc. IEEE*, vol. 102, no. 10, Oct. 2014, DOI: 10.1109/JPROC.2014.2347207.

[97] A. P. Alivisatos *et al.*, "The brain activity map project and the challenge of functional connectomics," *Neuron*, vol. 74, no. 6, pp. 970–974, 2012.

[98] H. Markram *et al.*, "Introducing the human brain project," *Procedia Comput. Sci.*, vol. 7, pp. 39–42, 2011.

[99] T. Serrano-Gotarredona, T. Prodromakis, and B. Linares-Barranco, "A proposal for hybrid memristor-CMOS spiking neuromorphic learning systems," *IEEE Circuits Syst. Mag.*, vol. 13, no. 2, pp. 74–88, 2nd Quart., 2013.

[100] E. C. Sobel, "The locust's use of motion parallax to measure distance," *J. Compar. Physiol. A*, vol. 167, no. 5, pp. 579–588, 1990.

[101] H. Reichardt, "Über die geschwindigkeitsverteilung in einer geradlinigen turbulenten couettesträmung," *Zeitschrift für Angewandte Mathematik und Mechanik (J. Appl. Math. Mech.)*, vol. 36, no. S1, pp. S26–S29, 1956.

[102] S. Martinez-Conde, S. L. Macknik, and D. H. Hubel, "The role of fixational eye movements in visual perception," *Nature Rev. Neurosci.*, vol. 5, no. 3, pp. 229–240, 2004.

[103] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

ABOUT THE AUTHORS

Garrick Orchard received the B.Sc. degree in electrical engineering from the University of Cape Town, Cape Town, South Africa, in 2006 and the M.S.E. and Ph.D. degrees in electrical and computer engineering from The Johns Hopkins University, Baltimore, MD, USA, in 2009 and 2012, respectively.

He is currently a Postdoctoral Research Fellow at the Singapore Institute for Neurotechnology (SINAPSE), National University of Singapore, Singapore, where his research focuses on developing neuromorphic vision algorithms and systems for real-time sensing on aerial platforms. His other research interests include mixed-signal very large scale integration (VLSI) design, compressive sensing, spiking neural networks, visual motion perception, and legged locomotion.

Dr. Orchard was a recipient of the The Johns Hopkins University Applied Physics Laboratory (JHUAPL) Hart Prize for Best Research and Development Project, and won the best live demonstration prize at the 2012 IEEE Biomedical Circuits and Systems Conference (BioCAS). He was named a Paul V. Renoff Fellow in 2007 and a Virginia and Edward M. Wysocki Senior Fellow in 2011.

Ralph Etienne-Cummings (Fellow, IEEE) received the B.Sc. degree in physics, from Lincoln University, Oxford, PA, USA, in 1988 and the M.S.E.E. and Ph.D. degrees in electrical engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 1991 and 1994, respectively.

He is currently a Professor of Electrical and Computer Engineering, and Computer Science at The Johns Hopkins University (JHU), Baltimore, MD, USA. He is the former Director of Computer Engineering at JHU and the Institute of Neuromorphic Engineering. He is also the Associate Director for Education and Outreach of the National Science Foundation (NSF)-sponsored Engineering Research Centers on Computer Integrated Surgical Systems and Technology at JHU. His research interest includes mixed-signal very large scale integration (VLSI) systems, computational sensors, computer vision, neuromorphic engineering, smart structures, mobile robotics, legged locomotion, and neuroprosthetic devices.

Dr. Etienne-Cummings has served as Chairman of the IEEE Circuits and Systems (CAS) Technical Committee on Sensory Systems and on Neural Systems and Application. He was also the General Chair of the 2008 IEEE Biomedical Circuits and Systems Conference (BioCAS). He was also a member of Imagers, MEMS, Medical and Displays Technical Committee of the IEEE international Solid-State Circuits Conference (ISSCC) from 1999 to 2006. He is the recipient of the NSF's Career Award and the U.S. Office of Naval Research Young Investigator Program Award. In 2006, he was named a Visiting African Fellow and a Fulbright Fellowship Grantee for his sabbatical at the University of Cape Town, Cape Town, South Africa. He was invited to be a lecturer at the National Academies of Science Kavli Frontiers Program, in 2007. He has won publication awards, including the 2003 Best Paper Award of the *EURASIP Journal of Applied Signal Processing* and "Best Ph.D. in a Nutshell" at the 2008 IEEE BioCAS, and has been recognized for his activities in promoting the participation of women and minorities in science, technology, engineering, and mathematics.