NEUROMORPHIC SYSTEMS

Neuromorphic systems is the field of engineering that attempts to imitate the operation and structure of biological sensing and information processing nervous systems. The word "neuromorphic," originally introduced by Carver Mead (1, 2), is understood here as something that attempts to artificially recreate biological nervous systems. The way simple living beings sense and process information so very efficiently, relying on slow and often defective components called "neurons" and yet consuming so very little power, is truly amazing. Human beings have no difficulty in recognizing sophisticated objects within complex moving scenes and commanding our arms and fingers to grasp them naturally and effortlessly. However, we have still not yet succeeded in building machines that can do similar tasks at speeds and with power consumption similar to those of their biological counterparts.

In this article, our aim is to provide an overview of the core elements of artificial neuromorphic systems as they are understood today (although we realize that the result will almost inevitably be incomplete). First, neuromorphic systems can be divided into two main (and not necessarily overlap-free) categories: "sensory subsystems" (corresponding to biological organs such as eyes, ears, noses, and skin) and "processing subsystems" (corresponding to the biological brain). But neuromorphic systems must also take into account three separate, complementary processes: (a) information computation, (b) information communication, and (c) information storage. These three processes are habitually performed in standard man-made computers, but there they are typically implemented on separate, physical "pieces of hardware" (either chips or circuits within chips). In neuromorphic systems, such separation is not so clear-cut and all three processes are often implemented in one and the same physical element over and over again. For example, a synapse connecting two neurons performs computing, communication, and storage tasks.

In biological neural systems, information is typically exchanged through electrical spikes, except when some local tissue is used such as the retina, where continuous signals transmit information to neighboring cells. Here we will focus mainly on artificial neuromorphic systems that encode, transmit, and process information in the form of spikes. We will use the term "events" to reference such spikes. In general, the events discussed will be digital electronic multibit signals capable of traveling very quickly (typically, in fractions of microseconds) between electronic components. Artificial systems of this type are commonly known as "event-driven neuromorphic systems" (3), and their event-driven intercommunication strategy is called "address event representation" (AER) (4–9).

In the following sections, we will very briefly describe some of the principles involved (components, communications, learning) and look at some examples of sensors (retinas, cochleae, skins noses) and event computing systems. Sources or material will also be referenced for further reading.

1. PRINCIPLES

Figure 1 illustrates the conceptual differences between conventional computers and biological neural systems with regard to "processing," "communication," and mass information "storage." In a conventional computer, these three concepts are physically clearly separated, as illustrated in Figure 1a. Data computations are performed by a CPU (or several CPUs). CPUs are highly complex circuits capable of performing an extensive catalog of instructions. The data are stored in a physical memory, as is the algorithm or set of instructions that need to be executed on the data. Between the data memory storage devices and the CPU(s), there are one or more communication networks. If there are sensors or other devices, these are typically "seen" by the CPUs as additional memory locations. The CPUs read the instructions one by one from the "instruction memory," read the data required by the instructions from the "data memory," execute the instructions sequentially, and then save the data again in the data memory. This results in heavy use of the communication network, which is a shared resource. Nowadays, the timescales involved are usually subnanosecond. Computer engineering has evolved during the course of many decades, and there are now many ways of distributing physical memories, communications networks, and processing CPUs as more or less hierarchical structures and layouts. However, in the end, processors, memory, and communication networks are still clearly separated.

In contrast, biological neural sensory and processing systems are structured very differently, as is illustrated in a highly simplified manner in Figure 1b. The main processing element is the neuron, which always performs the same function: It accumulates input spikes and, when enough of these have been received within a given time, it produces its own output spike and resets itself. Neurons are interconnected by "synapses" that transport the spike from its producer neuron to a receiver neuron. Synapses are characterized by a "strength" or "weight" that modulates the effect of the spike on the destination neuron. The time constants involved are on the order of milliseconds or fractions of seconds. Synapses store the system knowledge in their weights and in the system's own connecting structure. They, therefore, perform communication and memory functions simultaneously. Processing is performed together by neurons and synapses. Neurons are typically grouped into populations, such as those in a sensor (like an eye or ear), or as hierarchies of populations within the brain. In biological neural structures, there is therefore no clear distinction between elements performing processing, communication, and memory tasks.

Neuromorphic engineering is the building and operating of man-made hardware that mimics the principles of neural processing in biological systems. In the next section, we give an overview of the discipline and provide some sources for further reading.

1.1. Introduction to Spiking Neurons

The original artificial neural networks proposed in the 1940s (10) and the perceptron concept (11), together with

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*.
DOI: 10.1002/047134608X.W8328

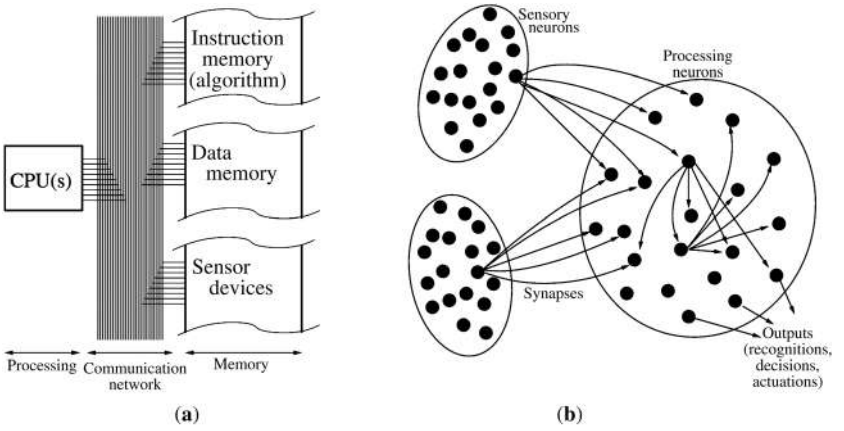

Figure 1. Illustration of the physical realization of the concepts of "processing," "communication," and "storage" in conventional computers (a) and biological neural systems (b).

its powerful back-propagation training rule (12), were the keys that opened the door to the whole modern field of artificial neural networks and machine learning. They were based on the neuron idea expressed in Figure 2a, which here we call the "static neuron." In a "static neuron"-based neural computing system, each neuron receives a static input pattern $(x_1, x_2, \ldots, x_n)$ where each input component x_i is a numerical value. A neuron's output is computed as

$$y_j = h\left(\sum_i w_{ij} x_i\right)$$

where w_{ij} is the "synaptic weight" connecting input x_i to neuron N_j, and $h()$ is a nonlinear activation function. Inputs x_i can be provided by external inputs or by outputs from other neurons. This type of static neuron is typically used in neural-based image processing systems (13), where the system input is made up of all the numerical values of the input image pixels. The neuron outputs to which these input pixels connect are computed using the equation above. These outputs connect to other neurons, whose outputs are computed in the same way, and this continues until all neuron outputs have been computed. Normally, systems of this type are structured as hierarchical feed-forward layers, where neuron layers alternate with max and/or pooling layers (14, 15).

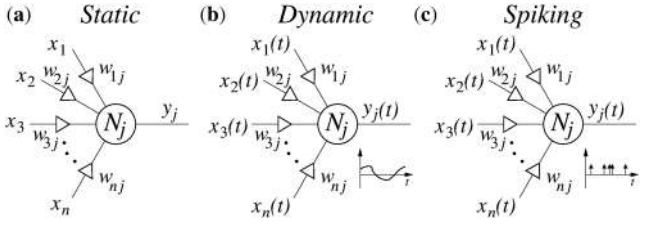

Figure 2. Fundamental types of neuron processing principles. (a) Static type, all inputs and outputs are one single numerical value. (b) All inputs and outputs are dynamic continuous-time functions. (c) All inputs and outputs are represented by asynchronous sequences of instantaneous events, called "spikes," which resemble signals in biological nervous systems.

Figure 2b represents a "dynamic neuron," in which both inputs and outputs are represented by dynamically evolving continuous time functions. The internal operation of this type of neuron is typically described by a differential nonlinear equation dependent on the weighted inputs $w_{ij}x_i(t)$. This produces an internal dynamic neuron state, which is typically mapped nonlinearly to the output variable $y_j(t)$. Dynamic neural systems resulting from dynamic neurons naturally allow for feedback (16). However, stability conditions need to be imposed (17) and this produces systems with a high number of differential equations that need to be solved numerically at each time step.

Figure 2c represents a "spiking neuron" in which both inputs and outputs are represented by sequences of asynchronous spikes, as in their biological counterparts. The internal operation (and also the synapses) of a spiking neuron can be described by a differential nonlinear equation (18). However, computation efficiency can be optimized if these differential equations only need to be updated whenever a new input event is received. In this case, the whole system is purely event-driven (and does not have to rely on any clock, in the case of a hardware implementation, or on a global time step in the case of a software implementation). Computations are only necessary whenever events are communicated. Representing signals with spikes is an additional difficulty. One obvious choice is to map the "static neuron" representation by transposing the numerical values x_i, y_j into neuron spike rate frequencies and then let the system settle to a stable stationary state. However, this approach results in a highly inefficient computational spiking system, since many spikes would be required to represent a numerical neural input/output value. Smarter techniques have been proposed that require just one spike per numerical value (19), or just a few through low-rate encoding (20). Biology itself seems to adopt both approaches: a fast single-spike per neuron response capability combined with a slower but more accurate multiple-spike per neuron signal encoding capability.

In the next section, we will focus on hardware techniques and the implementation of spiking neural systems.

1.2. Silicon Neurons Summary

Hardware implementations of spiking neurons are important for designing neuromorphic computing architectures and can be extremely useful for a wide variety of applications, ranging from the high-speed modeling of spiking neural networks to the real-time processing of sensory signals in autonomous cognitive agents. Hardware emulation of real neurons is still an active topic of research, because of the need for very high levels of efficiency and very large-scale integration in advanced CMOS VLSI processes. Several different types of hardware implementations have been proposed for silicon neuron models, using a wide range of different circuit solutions. One of the first circuits designed to model the function of real neurons was the conductance-based silicon neuron proposed in 1991 (21). A simplified schematic diagram of this circuit is shown in Figure 3.

In this circuit, the relevant neuron conductances were modeled using transconductance amplifiers operated in the weak-inversion regime. This allowed the amplifiers to exhibit a sigmoid transfer function, and this was exploited to model the active and passive neuronal ion channels.

This circuit can produce action potentials very similar to those measured in real cells. Due to its relatively large size and high number of potential device mismatch issues, however, other solutions based on simpler integrate-and-fire (IF) models have been proposed more recently, adopting a variety of circuit design approaches (22).

In the field of computational neuroscience, Brette and Gerstner (23) proposed the "adaptive exponential integrate-and-fire" (AdEx-IF) neuron, a generalized version of IF neuron models that acts as a bridge between biologically realistic but complex conductance-based models and very simplistic integrate and fire models. Like other, alternative, two-variable generalized IF models (24, 25), AdEx-IF can reproduce a wide range of different firing patterns as a function of its parameters (26), thus generating a compact description of many different types of biological neurons.

In general, two-variable generalized IF neuron models lend themselves well to compact VLSI implementation using analog electronic circuits (27–31). Figure 4 shows an example of a silicon neuron circuit with AdEx-IF model properties: it produces rising exponentials with negative exponent at the onset of the stimulation, and rising exponentials with positive exponent just before reaching the spiking threshold (28). The differential-pair integrator circuit (32) in the "leak" block emulates the leak conductance of more complex neuron models. At the same time, the transistors in the "positive feedback" block in Figure 4 model both the sodium activation and sodium inactivation channels in an extremely compact manner. Similarly, the effect of potassium channels in real neurons is emulated using the transistors in the "reset" block. The circuit also emulates the effect of calcium ion channels via the "adaptation" block, which produces the second slow variable of the two-variable system. This is useful for reproducing the wide repertoire of neural dynamics previously demonstrated with theoretical models.

1.3. Address Event Representation (AER) and Communication Mesh Networks

AER has become a popular "virtual wiring" technique for interconnecting spiking neuromorphic systems. The high

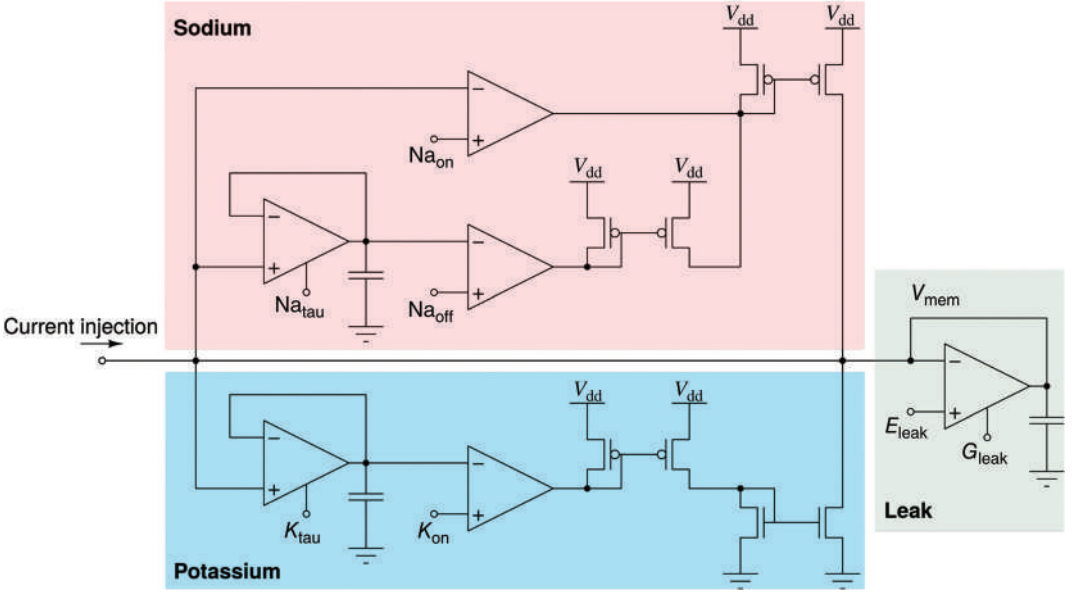

Figure 3. A conductance-based silicon neuron. The "leak" block models the passive leak behavior of the neuron: In the absence of stimulation, the membrane potential V_{mem} leaks to E_{leak} following first-order low-pass filter dynamics. The "sodium" block comprises sodium activation and inactivation circuits that model the sodium conductance dynamics observed in real neurons. The "potassium" block has circuits that reproduce potassium conductance dynamics. The G_{leak}, Na_{Tau}, and K_{Tau} parameters determine the time constants of the passive, sodium, and potassium channel dynamics, respectively, while the Na_{On} and Na_{Off} parameters set the sodium activation and inactivation thresholds, respectively, and the K_{on} parameter sets the potassium activation threshold.

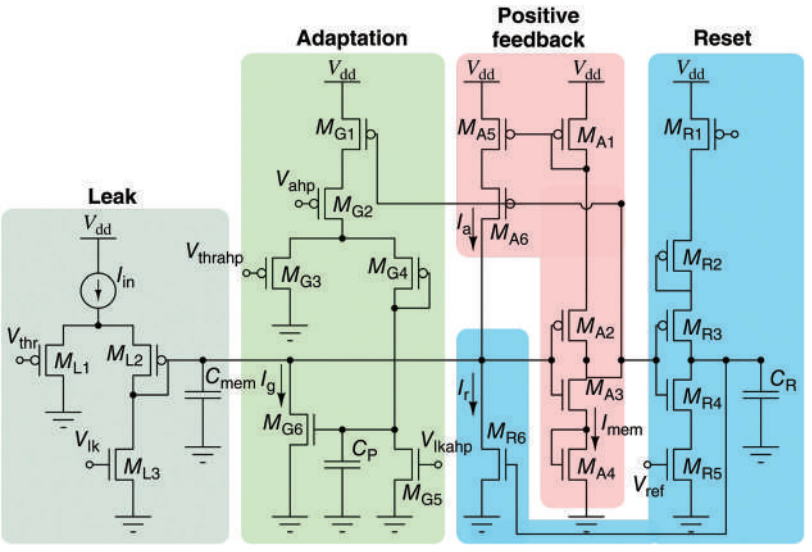

Figure 4. A generalized IF (integrate and fire) neuron equivalent to the AdEx-IF (adaptive exponential integrate and fire) neuron computational model. The log-domain leaky integrator $M_{L1–3}$ implements the neuron's input leak conductance. The positive feedback block $M_{A1–6}$ comprises a noninverting amplifier with current-mode positive feedback, which produces the action potential. The reset block $M_{R1–6}$ resets the neuron to the reset voltage and keeps it low for a user-configurable refractory period. The adaptation block $M_{G1–6}$ produces a negative feedback current I_g that triggers the spike frequency adaptation mechanism. The circuit voltage biases can be used as parameters to change the spiking properties of the silicon neuron.

speeds available in digital interchip communications are exploited in AER to time-multiplex numerous synaptic connections between neurons. These connections only need to be active during spike (also called "event") transmission. In AER, whenever a neuron in a module generates a spike (event), its address or ID (typically its (x, y) coordinates) is written on an intermodule high-speed digital bus. An ID for the module, a sign bit, or additional informative parameters can also be appended to this "address." Figure 5 illustrates several scalable AER approaches reported in literature. Figure 5a shows the concept of Flat-AER (33). Each module contains an array of neurons. Each neuron is identified by an address inside the module together with a module ID, and therefore has its own unique global address. All modules share a single external AER bus connected to a programmable mapper. Each event generated is sent to the mapper, which identifies to which destinations it should go and generates events accordingly. Since all events flow through the mapper, the total communication bandwidth is limited by this block and its input and output AER buses. The mapper must also fan-out to a large number of other modules, causing even more delays. Broadcast-AER (34, 35), which is shown in Figure 5b, was designed to limit physical fan-out per bus (and thus improve speed) and implement intermodule communication by means of intermodule hops. However, it still suffers from the same mapper bottleneck. Figure 5c illustrates prestructured AER (36). Here all links are point-to-point, using splitter and merging modules to fan-out and fan-in. Address spaces are local to each link. In terms of communication efficiency, this is the most effective scheme. However, the network is not easily reconfigurable. Hierarchical-AER (37), as illustrated in Figure 5d, extends the concept of Flat-AER, with its one single mapper, to a hierarchy of mappers. It exploits the fact that most connectivity is typically local by using many fast, local mappers in parallel, thereby improving bandwidth. Like HiAER (37), Tree-AER (38) is a highly efficient tree-based hierarchical AER communication scheme that is guaranteed deadlock-free. Here, neurons also have unique global IDs and each module has its own local router to define system connectivity. The modules are connected in a tree arrangement. Figure 5e illustrates Router–Mesh–AER (39, 40). Neurons have a unique global ID (an ID within their module plus the module ID), but there is no global mapper. Each module has its own router and directs each incoming event through a traveling branch to its destination. Events can be encoded by either their source or their destination addresses, creating different trade-offs (40).

1.4. Learning and Adaptive Intelligence

The capacity of neuromorphic systems to adapt to and learn from the environments in which they operate is critical to attaining and maintaining optimum performance. Online learning embedded in neuromorphic silicon implementations therefore offers distinct advantages over approaches that relay external off-line training (41, 42). Depending on the performance metric, learning systems can be categorized as supervised, unsupervised, reinforcement learning, or different combinations of the three. All these types can be efficiently implemented in neuromorphic hardware. Such implementations typically embed local forms of incremental outer product learning onto crossbar arrays of synapses, where each weight is updated according to the

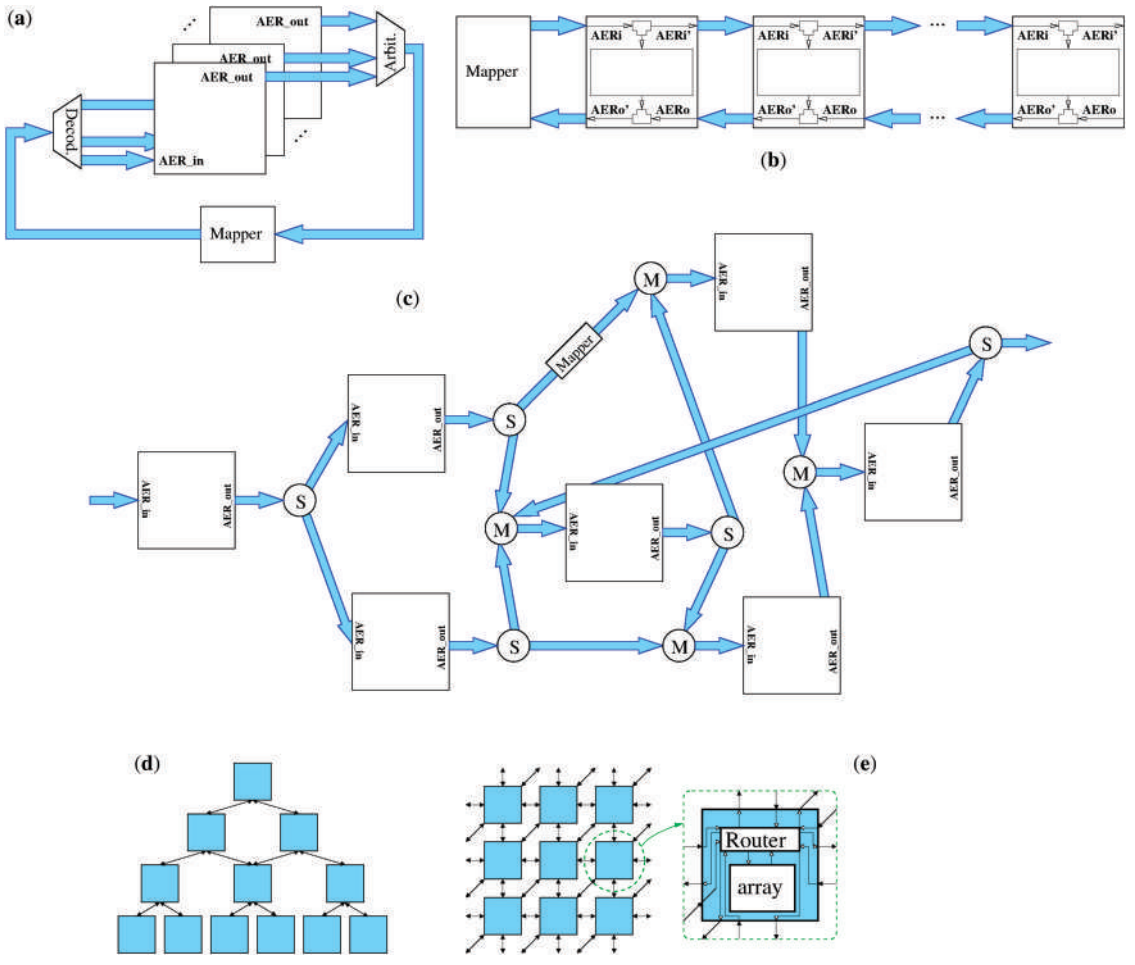

Figure 5. Scalable address-event representation approaches. (a) Flat-AER. (b) Broadcast-AER. (c) Prestructured AER. (d) Hierarchical-AER and Tree-AER. (e) Router-Mesh AER.

input- and output-related activities of the two connected neurons (41).

Efficient online learning is also feasible using event-driven spiking neuromorphic architectures that use address-event representation (AER) (Section 1.3) to achieve dynamically reconfigurable synaptic connectivity and plasticity. AER-based synaptic connectivity may be readily extended with local spike-timing dependent plasticity (STDP) mechanisms implemented directly in the address domain (43) to learn synaptic strength online from real-time data (44). STDP-based models of unsupervised temporally asymmetric Hebbian learning are also applicable to other forms of spike-based learning, such as reinforcement learning of distal reward, using STDP-modulated dopamine signaling (24), and deep learning of multilayered cortical representations, using STDP event-driven contrastive divergence in spiking Boltzmann machines (45). The advantage of AER-based synaptic connectivity for the efficient event-driven implementation of STDP-based online learning is that all information on synaptic strengths resides in local SRTs, in direct proximity to both presynaptic and postsynaptic event streams. The local implementation of event-driven STDP SRTs may therefore be sufficient to support more general implementations of complex nonlocal learning rules capable of benefitting from nested network structures, including global structures – thanks to the long-range, hierarchical connectivity provided by HiAER (37).

1.5. Mapping Generic Algorithms into Integrate-and-Fire Neurons

There has been significant research over the past two decades in developing new platforms for spiking neural computation. Current neural computers are primarily developed to mimic biology. They use neural networks, which can be trained to perform specific tasks to mainly solve pattern recognition problems. These machines can do more than simulate biology; they allow us to rethink our current paradigm of computation. The ultimate goal is to develop brain-inspired general-purpose computation architectures that can breach the current bottleneck introduced by the von Neumann architecture. Recently, a new framework for such a machine was proposed, called STICK (46). In this proposal, the use of neuron-like units with precise timing representation, synaptic diversity, and temporal delays allows setting a complete, scalable compact computation framework. The framework provides both linear and nonlinear operations, allowing representing and solving

any function. Preliminary usability examples in solving real use cases from simple differential equations to sets of nonlinear differential equations leading to chaotic attractors have been shown.

2. NEUROMORPHIC SENSORS WITH EVENT-DRIVEN OUTPUT

2.1. Neuromorphic Vision Retinas

Artificial neuromorphic silicon retinas were first proposed by Mead and Mahowald (47). Each pixel would compute an output proportional to the difference between its sensed light intensity and the average light intensity in its neighborhood. To read out such a preprocessed sensed image of this type, however, a conventional scanning mechanism was required capable of reading out all preprocessed pixel values. This section gives an overview of artificial retinas incorporating an asynchronous pixel-driven address-event-representation readout mechanism.

Luminance/Color Retinas. The simplest light-to-event transformation is achieved by mapping linearly from light intensity (photosensor current) to pixel event frequency. This was the approach followed in Culurciello's Octopus retina (48), but using an integrator with positive feedback to integrate the pixel photosensor current up to a given threshold and triggering a digital event communication circuit with microsecond response time to minimize power consumption. Depending on incident light intensity, each pixel generates an event frequency ranging from 8 mHz to 8 MHz and covering a 9-decade dynamic range (180 dB).

Based on a similar principle, Olsson and Hafliger (49) proposed a spiking pixel containing a vertical stack of photodiodes, each sensitive to a different range within the visible spectrum. Two independent photocurrents were integrated separately, generating two streams of spikes. By analyzing the two different frequencies produced, it was possible to estimate the dichromatic spectral content of the incident light per pixel.

Similarly, Fasnacht and Delbruck (50) also used a stacked two-diode structure to measure relative long- and short-wavelength spectral content. In this case, however, the circuit output was a digital PWM signal. Absolute intensity was encoded by the signal's frequency and the relative photodiode current was encoded by the signal's duty cycle. The signal itself was generated by a self-timed circuit alternately discharging the top and bottom photodiodes.

This light intensity to frequency readout approach has generally been abandoned because of unfair distribution of readout bandwidth to brightly lit pixels; one single highlight in a scene, for example, may contribute most of the output data. The sensing of color by vertical color separation has not been commercially successful compared to using color filter arrays due to the poor color separation capabilities of stacked junctions and the difficulty of achieving complete charge transfer from deeply buried junctions.

Spatial Contrast. Neuromorphic spatial contrast retinas are vision sensors that implement the computational behavior of the ON-center–OFF-surround sustained response from the horizontal and bipolar cells in the biological retina (51). Spatial contrast retinas output trains of address events that are proportional to the computed local spatial contrast in the observed scene.

Computing contrast at the focal plane level alleviates the dynamic range problem and significantly reduces the output data flow while preserving the relevant information on shape for object recognition.

The first spatial-contrast-sensitive silicon retina was implemented in Carver Mead's laboratory (21, 47). Figure 6a illustrates the architecture of Mahowald's retina. The photodetectors in this silicon retina are implemented using parasitic bipolar transistors. The photocurrents are converted to a photovoltage through MOS-diode connected transistors. This voltage is averaged using a hexagonal grid of diffusive resistors (implemented using a network of MOS transistors (1, 52)). Local contrast is computed as the difference between the local photovoltage and the local average voltage computed by the diffusive network. The smoothing factor depends on the value of the resistors (or the diffusion constant in semiconductors). Boahen subsequently published a more sophisticated biharmonic spatial contrast retina (53). Figure 6b shows the schematics of Boahen's pixel. In this retina, two cross-coupled diffusive networks with two different smoothing constants are implemented. The difference in the diffusion lengths of the two resistive grids generates the antagonistic ON-center–OFF-surround response. However, none of these retinas was equipped with a pixel-driven AER readout until Mortara et al. reported an unarbitrated readout scheme (9) and Boahen described a collision-free arbitrated scheme (4).

Based on Boahen's 1992 topology, Zaghloul and Boahen published a spatiotemporal contrast retina (54, 55) by adding temporal filtering and adaptation to the temporal average. The main problem that limited the performance of these prototypes was their fixed-pattern noise (FPN), caused by the mismatching of the electrical parameters for the CMOS transistors. Spatial contrast retinas with in-pixel calibration to reduce FPN were later developed (56, 57). Figure 6c illustrates the schematics of the spatial contrast retina implemented by Leñero-Bardallo et al. (57). It is based on Boahen's biharmonic retina, but incorporates in-pixel calibration circuitry and the voltage biases of the diffusive networks are self-adapting. Figure 6d–g shows some experimental results of natural images captured with this retina, after calibration. The images were obtained by histogramming the events produced by 32×32 pixels while observing a scene for 30 ms. An average of 4552 events were obtained per image. However, the need for calibration and the large pixel area ($81 \times 76\,\mathrm{mm}^2$ in a 0.35 mm technology (57)) has reduced commercial interest in this type of retinas.

Other interesting contrast-sensitive retinas that have been developed (58–60) compute spatial contrast using only the information from the nearest-neighbor pixels. However, these devices do not perform fully asynchronous computation as biological vision systems do, but rely on a global reset to begin the contrast computation, thereby introducing an artificial frame time.

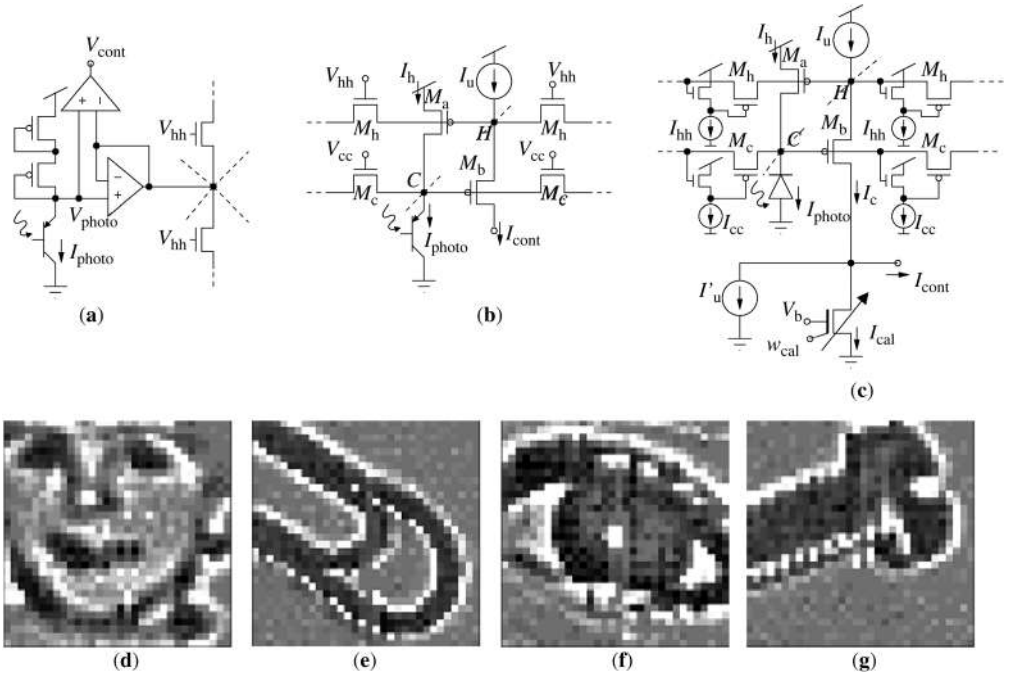

Figure 6. (a) Mahowald's contrast pixel. (b) Boahen's original pixel. (c) Leñero's pixel. (d–g) Natural images captured with Leñero's spatial contrast retina after calibration.

Motion/DVS Retinas. In motion or dynamic vision sensors (DVS), each pixel calculates the time derivative of the light it senses, optionally performs some processing (e.g., detecting a given change of light between consecutive events) and, when a certain level (threshold) is reached, emits an "event." The event usually consists of the (x, y) coordinates of the pixel within the two-dimensional photosensor matrix. The output of a DVS thus comprises a flow of (x, y) coordinates corresponding to the different pixels that have detected changes in the brightness of the light they are sensing. This concept was first proposed by Landolt et al. (61) and Kramer (62). Prototype sensors of this kind suffered from high interpixel mismatch and responded to a threshold on the derivative, resulting in low temporal contrast sensitivity. Later, an improved sensitivity sensor was reported by Lichtsteiner et al. (63). Figure 7a shows how, in these DVS sensors, the photocurrent I_{ph} sensed by a photosensor is first transformed into voltage through logarithmic conversion with optional voltage amplification A_v, resulting in $V_{ph} = A_v\, V_o \log(I_{ph}/I_o)$. In DVS pixels, V_{ph} is memorized on a capacitor after each event, and the pixel outputs an event when the change of V_{ph} from the memorized value exceeds a threshold. Parameter I_o suffers from interpixel mismatch. Depending on the circuit implementation of amplifier A_v, this parameter may also introduce mismatch. A low-mismatch low-power amplification based on subthreshold stacked diodes was proposed by Serrano-Gotarredona and Linares-Barranco (64). The time derivative is then calculated, and some additional processing may be performed. The resulting time derivative is

$$\frac{dV_{ph}}{dt} = A_v V_o \frac{1}{I_{ph}} \frac{dI_{ph}}{I_{ph}}$$

where high-mismatch parameter I_o has been canceled. The time derivative of V_{ph} is obtained by sensing the current through a capacitor with its terminal voltage difference set to V_{ph}, as illustrated in Figure 7b. This yields $I_D = C_1 dV_{ph}/dt$. This time derivative can be used directly to determine the relative change of light in a pixel, since it is normalized with respect to light and will thus provide a measure of temporal contrast. This can be done by replacing the square box in Figure 7b by a resistor R, resulting in $V_D = -RI_D$. It is also possible to postprocess the time derivative to obtain, for example, an accumulated computation of it. This is the case of reported DVS sensors (63, 64) where, as shown in Figure 7c, the postprocessing consists of an integrate-and-fire operator: The derivative current I_D is integrated on capacitor C2 until the accumulated integral V_o reaches a given threshold (sensed by a comparator circuit, not shown), after which it is reset by the switch shown in

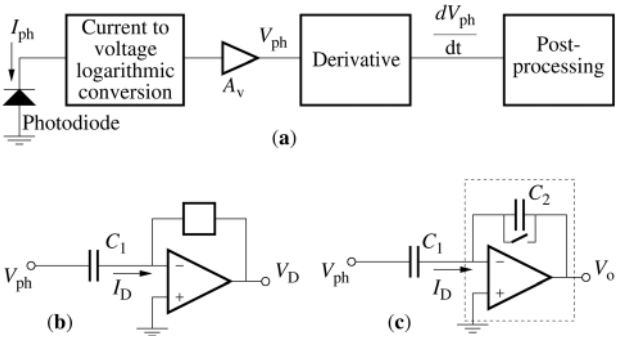

Figure 7. Dynamic vision sensor basics. (a) Conceptual block diagram. (b) Example circuit for time derivative. (c) Example circuit for time derivative and postprocessing.

the figure. At this instant, an output event is generated for the pixel and sent out as an "address event."

Combined DVS/Luminance. The temporal contrast motion-detection retina pixels discussed in the previous section respond to relative changes in light intensity but do not collect information about absolute light levels. As a result, image data in the form of gray levels, and, consequently, information about static (parts of) scenes, are absent from the outputs generated by such sensors. Attempts have been made to combine the strengths of temporal contrast pixels – namely, high temporal resolution, wide dynamic range, and, most importantly, suppression of redundancy in the data acquired (in comparison with conventional image sensors) – with an ability to acquire absolute light intensity information.

The asynchronous time-based image sensor (ATIS) described in Reference 65 combines relative change detection with absolute exposure measurement at the single-pixel level. The DVS change detector asynchronously initiates the measurement of a new exposure value immediately after a brightness change of a certain magnitude has been detected by the pixel. The exposure measurement circuit converts the integrated photocharge into the timing of asynchronous pulse edges using a single-slope photocurrent-to-time converter, as illustrated in Figure 8. Pulses are transmitted off-chip using AER communication. As a result, gray-level image information is acquired and transmitted continuously, albeit only from parts of the scene where there is new visual information.

Another recent approach to combining dynamic and static information in a single pixel is a solution known as the dynamic and active pixel vision sensor (DAVIS) (66). This pixel combines conventional active pixel sensor (APS) frame-based sampling of intensity with asynchronous detection of log intensity changes. DAVIS has the advantages of using the same photodiode as the DVS circuit and a small readout circuit that only adds a few transistors to the pixel. It increases the DVS pixel area by about 5%, resulting in a pixel area that is about half of the original ATIS pixel at the same process feature size. APS output has the limited dynamic range and redundant data capture disadvantages of conventional frames, but it allows conventional images to be captured concurrently with DVS events and potentially has the capacity to bring together conventional machine vision and bioinspired event-based approaches. The combination of color APS with DVS was also recently reported in Reference 67; the device proposed was based on macropixels comprising one DAVIS pixel with three APS pixels covered with RGB color filters.

Alternative schemes of combined DVS/luminance sensing schemes have also been reported by Kim et al. (68–70) and Pardo et al. (71).

An overview of recent developments in the field can be found in References 3 and 72.

2.2. Review of Neuromorphic Audio Cochleae

In human hearing, incoming sounds cause the eardrum to vibrate, leading in turn to vibration of the bones in the middle ear and the generation of hydrodynamic waves in the inner ear, or cochlea. These waves are coupled with the vibrations of the basilar membrane (BM), which sits in the middle of a fluid chamber in the cochlea.

The BM was the first cochlea structure to be modeled electronically in silicon, by Lyon and Mead (73). Its tapered shape (narrow and stiff at the start, becoming wide and flexible at the end) means that different frequencies are detected at different places along the membrane. The biophysics of this membrane is modeled as a large number of coupled filter stages with best characteristic frequencies ranging from tens of hertz to tens of kilohertz with log frequency spacing. Filter architectures range from the cascaded form (73–76) used to model the phenomenological output of the cochlea to a resistively coupled bank of bandpass filters (76–79) for modeling the role of the BM and the cochlear fluid more explicitly.

The organ of Corti that sits atop the BM contains both the inner and outer hair cells (*IHCs* and *OHCs*, respectively). The IHCs transduce the vibrations of the membrane into a neural signal that is transmitted by the auditory nerve fibers. The biological function of the IHCs is typically modeled as a half-wave rectifier in an electronic circuit based on known recorded physiological responses (80). The OHCs implement local automatic gain control (AGC) by altering the length and width of their cell bodies. Various electronic implementations of the local biological AGC function have been described in References 78, 79 and 81.

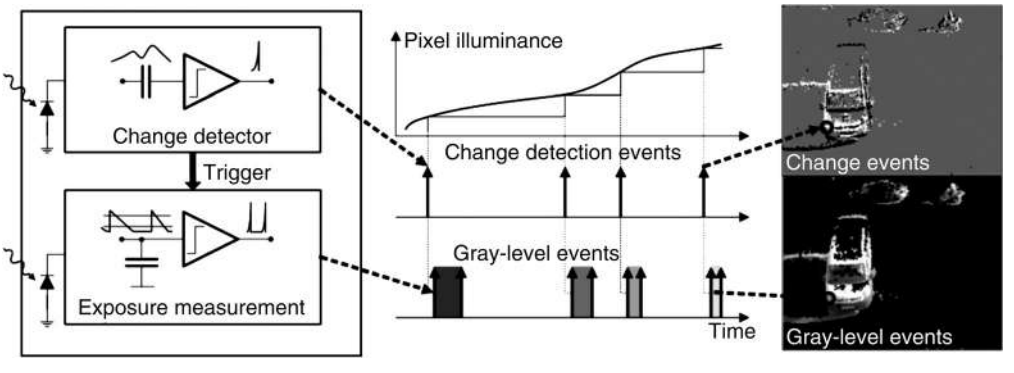

Figure 8. Illustration of pixel operation combining change detector and exposure measurement. *Left:* Simplified pixel conceptual block diagram. *Center:* Time evolution of light (*top*), change detector events (*center*), and exposure measurement events (*bottom*). *Right:* Corresponding updates within the full visual scene.

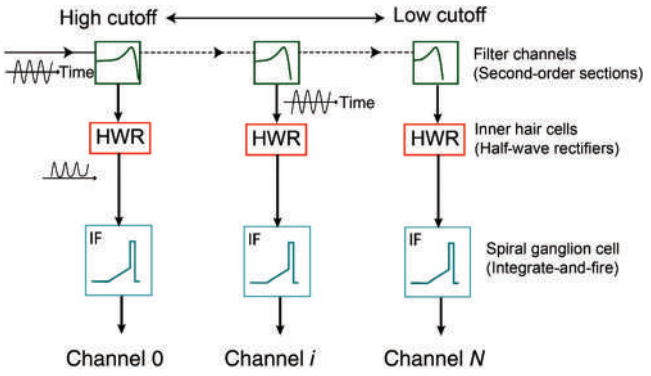

Figure 9. Schematic illustration of artificial cochlea structure.

IHC outputs are transformed into asynchronous spikes through circuits that quantify them into encoded data using address-event representation (79, 82, 83). The core of the latest 64-channel binaural AER cochlea from Reference 83 consumes only 55 uW. An example block diagram of a cascaded architecture with AER outputs is shown in Figure 9. A review of cochlea designs can be found in Reference 84 and the historical development of a more extensive list of silicon cochlea designs over the last two decades is described in Reference 3. These earlier analog cochlea implementations have been used in applications such as pitch detection (85) and localization (86). Applications involving spiking cochleae are described in Section 4.2.

2.3. Neuromorphic Tactile Sensing

The application domains of touch sensing technology are currently widening. Touch sensors are crucial in robotics. In prosthetics (87), exoskeletons, robot-assisted surgery (88), and teleoperation, they convey sensory feedback to the user, greatly improving the system's usability and effectiveness; in service robotics (89), they are essential for safe interaction and for improving manipulation (90) and planning (91).

A tactile sensing element, or *taxel*, is either a structural unit that produces a signal as a response to a mechanical stimulus or a material (or aggregation of materials) with, for example, piezoelectric, piezoresistive, or optical properties that intrinsically converts mechanical stress/strain into an optical or electrical signal. In the latter case, e-skin offers multimodal sensing capabilities (69), bendability, flexibility, stretchability, and, hopefully, an ability to shrink and wrinkle comparable to that of human skin (92).

The POSFET (piezoelectric oxide silicon field-effect transistor) (93) device is a sensotronic unit (see Figure 10) in which signals are transduced using a film of piezoelectric material that biases the gate of a MOS transistor. In comparison with other touch devices, the POSFET offers faster response time and better pressure sensitivity.

Taxels can cover large areas, such as the outer body surface of a robot or a screen. In most cases, taxel activation is highly localized in time and space, and the resulting event-driven sensing strategy is greatly beneficial for the efficient encoding, transmission, and processing of tactile information, halving the amount of signal transmission needed by a heavily actuated robot and reducing it to less than 20% in other applications.

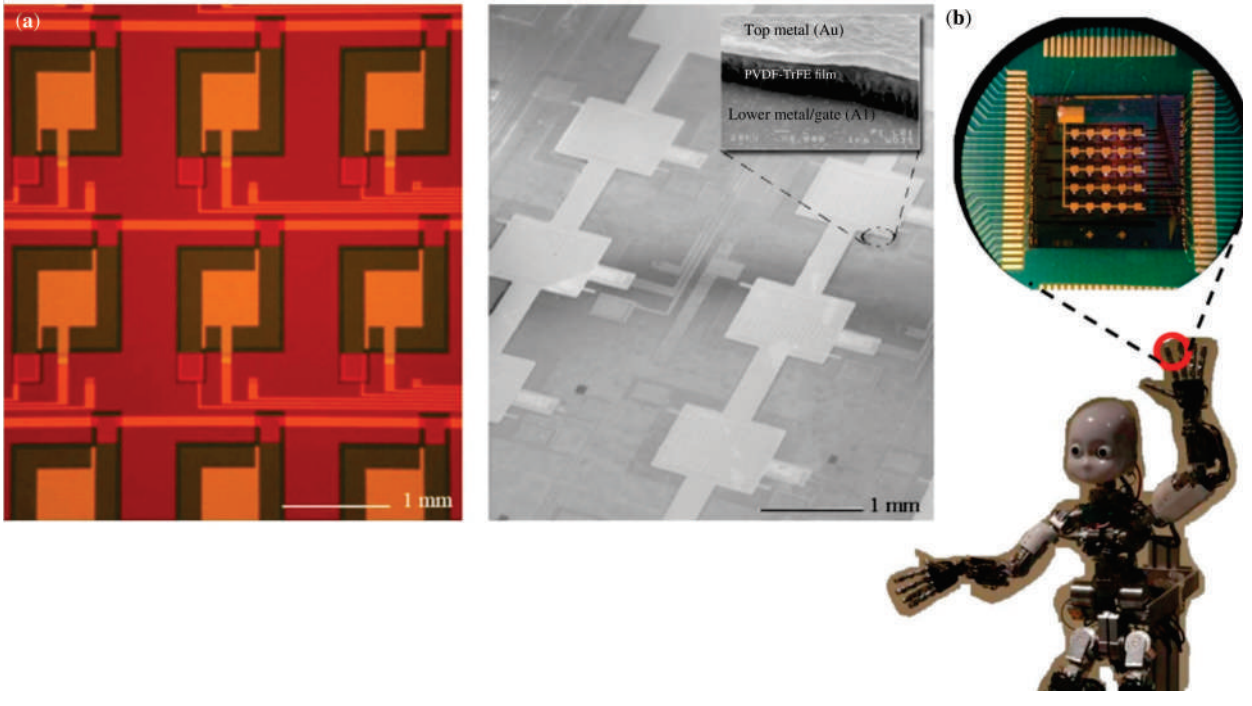

Figure 10. (a) Two images of POSFET (piezoelectric oxide silicon field-effect transistor) device arrays. (b) Example of the integration of a POSFET tactile sensing array into the fingertip of iCub. (Courtesy of Istituto Italiano di Tecnologia and EU Roboskin, grant FP7-231500.)

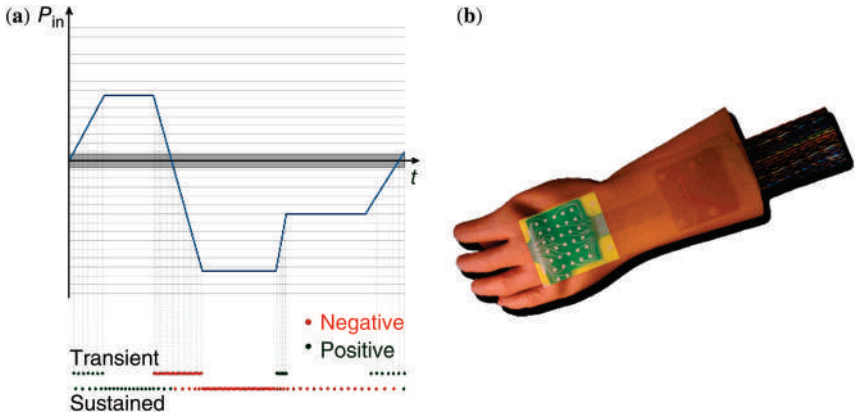

Figure 11. (a) Illustration of the encoding performed by the event-driven taxel. A positive and a negative neuron are used for encoding pressure and release, respectively. The transient taxel responds to changes of the input and the sustained taxel responds to the absolute level of the input force. (Courtesy of NSI group, iCub Facility, Istituto Italiano di Tecnologia). (b) Example of the integration of a tactile sensing array into a human-like artificial prosthetic limb. (Courtesy of COSMIC Lab., DITEN, University of Genova.)

Little new research work is currently being carried out in event-driven touch sensing. Existing tactile systems can be transformed into event-driven devices by postprocessing clock-generated signals. This method decreases data transmission, but is limited by the clock frequency and does not exploit the low-level data redundancy and adaptive signal sampling offered by event-driven signal acquisition. This approach has been addressed on FPGAs and CPUs (94, 95) and also on dedicated ASICs (96).

The event-driven paradigm should be applied at the level of sensory acquisition: As in dynamic vision sensors (63–65), the taxel triggers an event when it detects relevant information. As in human mechanoreceptors and the dual mode of operation proposed in vision sensors (65, 67), two main types of taxels should be implemented (97): a "transient" taxel that detects changes in the pressure applied to the skin and a "sustained" taxel that encodes its response to for an absolute value. Figure 11 shows how the two types of taxel work, encoding the change in the applied pressure value and the pressure value in their instantaneous firing rate, respectively.

2.4. Neuromorphic Olfaction

Taking the structure and dynamics of their biological counterpart as their model, neuromorphic olfactory systems detect and categorize chemical signatures using cross-selective gas sensor arrays and signal processing engines customized to accommodate the high dimensionality and great variability intrinsic to natural odor scenes. These scenes are highly complex, with signals of interest often mixed in with variable backgrounds, degraded by many sources of occlusion and interference, and with concentration ranges spanning several orders of magnitude. The olfactory systems of diverse species have evolved common (98) (and perhaps optimal) computational principles to tackle such complexity, and these principles offer clues as to how some of the major problems in machine olfaction can be overcome (99).

The gas sensors currently used for neuromorphic olfaction take a variety of material forms, such as conductive polymers (100), optical microbeads (101), metal–oxide chemiresistors (102), and G protein-coupled receptors (103). Like olfactory sensory neurons in biology, these sensors are relatively nonspecific and bind to a range of primary analytes in the environment. Specificity is achieved by incorporating diversity in the sensor array, either by instantiating a large number of sensors with different tuning profiles or through indirect techniques such as temperature modulation, whereby the varying selectivity of each sensor at different temperatures is leveraged. This diversity allows the sensor array to span a coding space large enough to represent the myriad combinations of analyte features that constitute natural odor signals.

Posttransduction processing, usually triggered by turbulent events in olfactory environments, is carried out by neuromorphic circuits modeling downstream computations of biological olfaction. The progressive decorrelation of odor representations observed *in vivo*, for example, is modeled by a hierarchical process in which chemiresistive signals are progressively categorized, resulting in representations that are initially coarse but that become increasingly refined over time. This process offers robust recognition and generalization capacities (42). In another study, increased separability between data samples of different categories and robust associations between analytes in the same class were illustrated in standardized data sets using soft winner-take-all computations inspired by the antennal lobe and mushroom body of the insect olfactory system (104). In yet another study, a neuromorphic implementation of glomerular-layer microcircuits in mammalian olfaction exhibited noise robustness, nontopographical decorrelation computations, and concentration normalization effects (105). Crucially, structural features observed in the biological system helped the researchers in this study to make effective use of hardware resources, significantly lowering the energy consumption of costly network-wide computations, as can be seen in Figure 12.

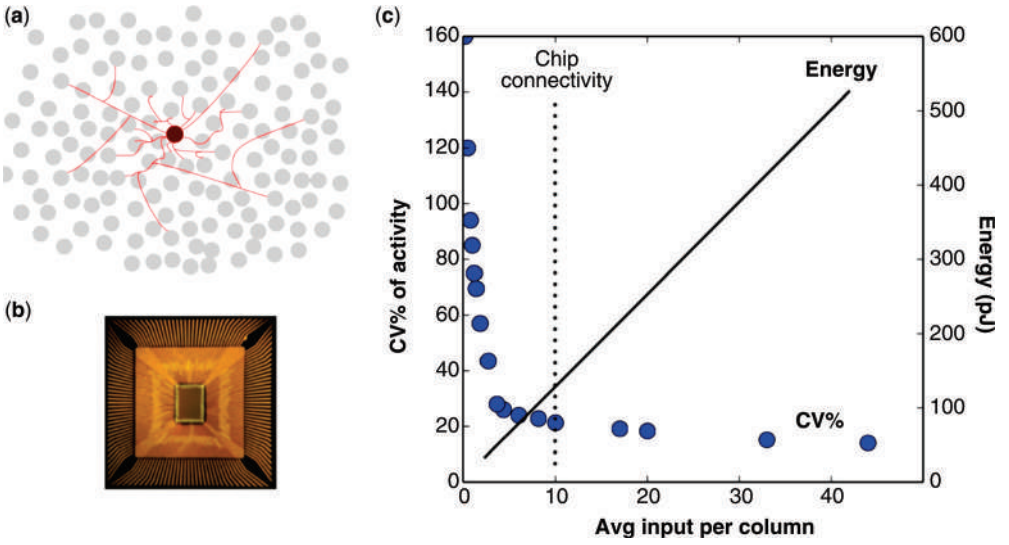

Figure 12. (a) Connectivity structure between olfactory bulb columns shows high local clustering with a few long-range connections. (b) A neuromorphic model of 48 glomerular columns in a digital ASIC measuring 3 mm × 2 mm. (c) Coefficient of variation (CV% – the standard deviation as a percentage of the mean) of activity across the columns (primary *Y*-axis) and energy consumption of the interconnection network (secondary *Y*-axis) as a function of the density of connections (*X*-axis). A high clustering coefficient and a small average path length in the network causes rapid information spread, enabling global computations such as odor intensity normalization to be carried out at a fraction of energy costs compared to a fully connected network (105).

3. EXAMPLES OF LARGE-SCALE NEUROMORPHIC PROCESSING SYSTEMS

Large-scale neural simulations are severely hampered by the current digital computing paradigm. Simulating a human-scale cortex model (10^{10} neurons and 10^{14} synapses) – the goal of the Human Brain Project (HBP) (106) – would require an exascale supercomputer (10^{18} flops) with millions of computing *cores* consuming a quarter-million households' worth of power (0.5 GW) (107). Simulations of this type are challenging because each neuron *distributes* its output to thousands of other neurons and, in turn, *aggregates* inputs from thousands more. The next section briefly describes some recent ongoing attempts to implement scalable approaches to large-scale neural brain emulation in HW.

3.1. SpiNNaker

SpiNNaker (a contraction of Spiking Neural Network Architecture) is a digital neuromorphic platform developed at the University of Manchester, UK. The key concept in SpiNNaker is the use of conventional mobile phone processors in a massively parallel configuration using AER communication via a packet-switched fabric. Neurons and synapses are modeled in software, resulting in a very flexible system capable of accommodating a very wide range of neuron models and learning rules.

The central component in SpiNNaker is a custom microchip incorporating 18 ARM968 processing units and a packet router (108). Each ARM968 core has 32 kbytes of tightly coupled memory to hold code and a further 64 kbytes of tightly coupled memory to hold local data. This chip is packaged with a standard 128 Mbyte low-power SDRAM memory so that large systems can be assembled by connecting multiple packages directly to their immediate neighbors in a 2D mesh (39).

For practical reasons, large SpiNNaker systems are built from a circuit board incorporating 48 packages – 864 ARM cores – and using FPGAs for high-speed serial communications to extend the mesh from board to board, as shown in Figure 13. The largest system currently planned will have over a million ARM cores on 1200 boards, assembled into 10′19″ rack cabinets.

The hardware platform is supported by software tools (109, 110) running on a conventional host machine that will accept a neural network description in a standard language such as PyNN (111) or Nengo (112) and will map the network onto the machine, typically allowing the network to run at biological speeds.

3.2. The BrainScaleS-System: A Mixed-Signal Approach to Neuromorphic Computing

Information processing in the brain is based on local analog computing and asynchronous spike-based network communications in continuous time. Although it is not fully understood why Nature has evolved this particular solution, it represents an ideal model for electronic implementation. The mixed-signal approach to neuromorphic systems was pioneered by the FACETS (113) and BrainScaleS (114) projects and is now being further developed in the neuromorphic computing subproject of the Human Brain Project (115). The name *BrainScaleS System* has been maintained for all systems in this project lineup. As a physical (electron-based) model of biological (ion-based) microcircuits, the mixed-signal approach combines an energy-efficient analog implementation of local nonlinear processing in neurons and synapses with a perfectly

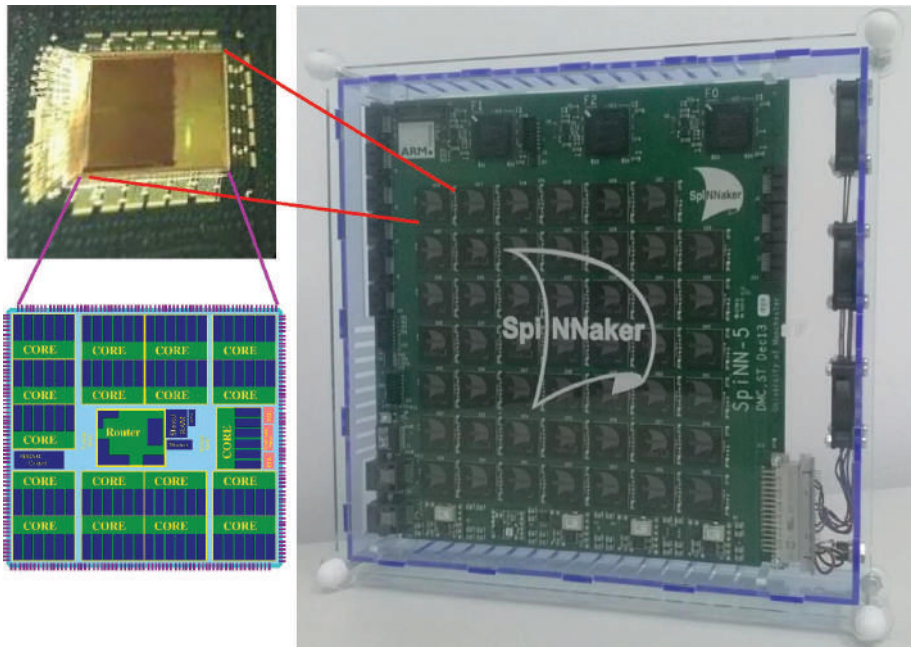

Figure 13. *Right:* SpiNNaker 48-chip board including three Spartan6 FPGAs for SATA interfacing purposes. *Top left:* Photograph of SpiNNaker chip packaged together with a 128 MB SDRAM chip. (Courtesy of Unisem Europe Ltd.). *Bottom left:* Layout diagram of SpiNNaker chip showing 18 ARM cores, each with tightly coupled SRAM, and the asynchronous router in the center.

scalable network fabric based on stereotypical action potentials (spikes).

From 2005 to 2015, the BrainScaleS system development went through two chip generations. The first generation, represented by the *Spikey* chip (116), featured 384 leaky-integrate-and-fire (LIF) neurons and 98304 conductance-based synapses on a single 5 mm × 5 mm chip, manufactured in the 180 nm process node by UMC (Taiwan). All time constants were scaled by a factor of 100 000 with respect to biological real time, offering accelerated access to network activities bridging many timescales. The chip featured three on-chip plasticity mechanisms: spike-timing-dependent plasticity (STDP), short-term synaptic depression (STD), and short-term synaptic potentiation (STP). Synapse weights were stored with a precision of 4 bits on individual SRAM cells in each synapse circuit. A block diagram of the network architecture is shown in Figure 14.

The second-generation chip is called *Hicann* (High Input Count Analog Neural Network). Manufactured in the same technology node as the first-generation chip, a single chip measures 10 mm × 5 mm. The synapse implementation and the plasticity models are identical to the first-generation chip. The major changes lie in the neuron circuits, which implement the adaptive exponential LIF model (*AdEx*), and in a very high synaptic input count per neuron of up

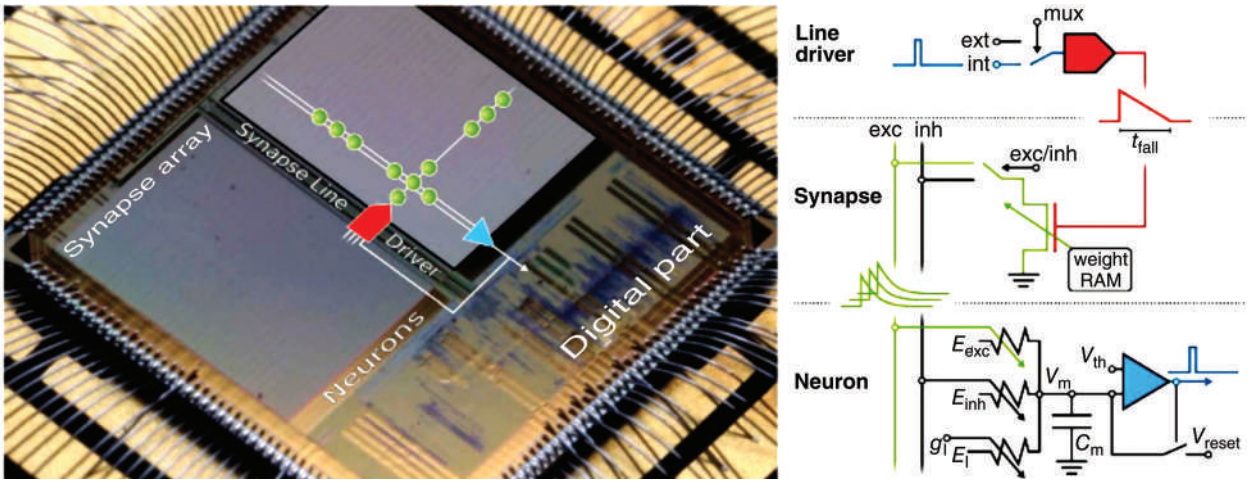

Figure 14. Mixed-signal network implementation in the Spikey-Chip, a predecessor of the BrainScaleS system. (From Reference 116.)

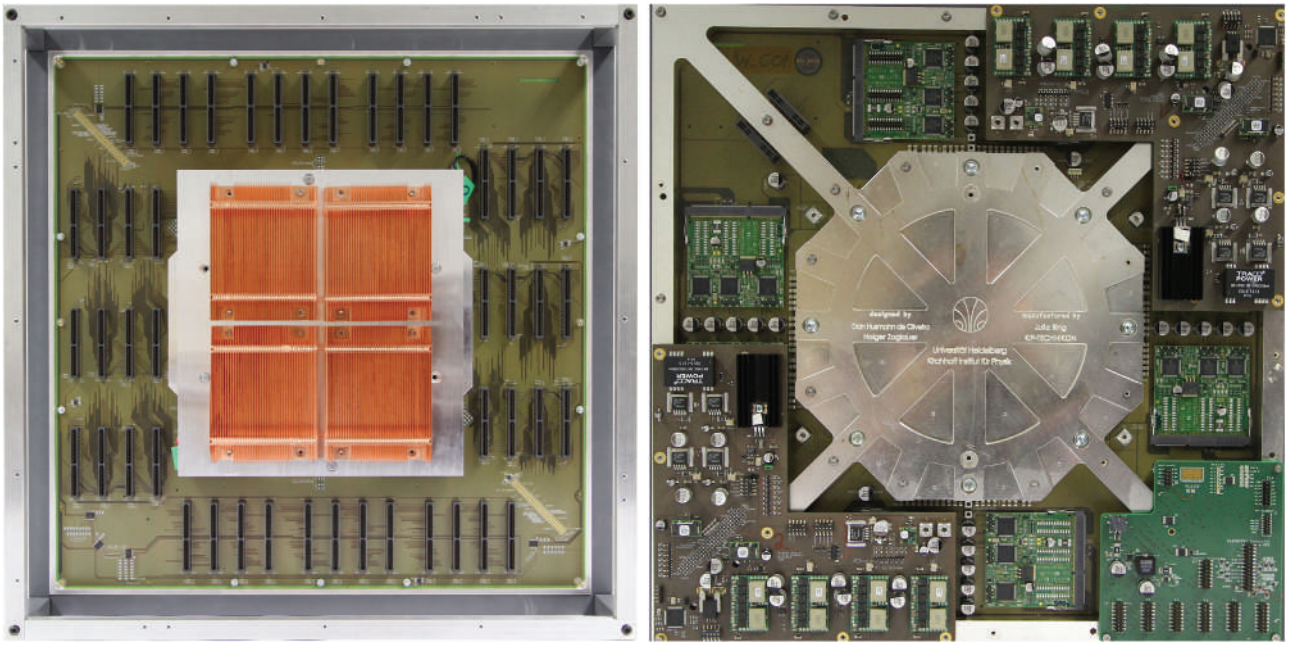

Figure 15. Front side and backside of the BrainScaleS wafer module.

to 16 000. Neuron parameters are stored on analog floating gate circuits. The acceleration factor with respect to biological real time has been set to 10 000. A single *Hicann* chip features 512 neurons and 114 688 plastic synapses. Another major development from the first- to the second generation is the realization of very large systems through wafer scale integration (117). A single 8 in. wafer carries a total of 384 interconnectable *Hicann* chips corresponding to a total of 44 million synapses and up to 196 608 neurons per wafer. In addition to the network wafer, a complete wafer module implements 48 FPGA communication modules providing off-wafer connectivity to other wafers and to host computers, as well as power supply and monitoring capabilities (Figure 15). A total of 20 such wafer modules have been assembled as part of the Human Brain Project (115) Neuromorphic Computing Platform, offering remote access to the system.

As a sideline of the second-generation chip development, the *Hicann* chip concept has been modified to implement multicompartment neurons (118). Multicompartment neurons feature passive dendritic branching, active dendrites (dendritic spikes), and back-propagating action potentials.

In 2015, work began on the development of third-generation chips. The most significant of several new features is the implementation of an on-chip plasticity processor (119). The processor has access to on-chip network activity and can control synaptic weights, network connectivity, and neuron parameters through local algorithms. The most relevant computational feature of the processor is its capability to implement structural plasticity, reward-based learning, and neuronal homeostasis.

3.3. TrueNorth

The IBM SyNAPSE TrueNorth neurosynaptic processor (120) integrates a 2D mesh of 64 × 64 cores, each having 256 crossbar-connected digital spiking neurons on a single 1M-neuron, 256M-synapse chip (Figure 16). The 2D mesh topology of TrueNorth extends to networks larger in size, with 4 × 4 tiles of TrueNorth chips integrated on printed circuit boards being combined in a single-rack system comprising 4096 chips with 4 billion neurons and 1 trillion synapses consuming an estimated 4 kW of power (121).

Implemented in 28 nm CMOS, the TrueNorth chip integrates 5.4 billion CMOS transistors. Internally asynchronous and globally synchronous, TrueNorth routes neural spike events across cores via address-event representation communication on a 2D mesh. Each core implements 256

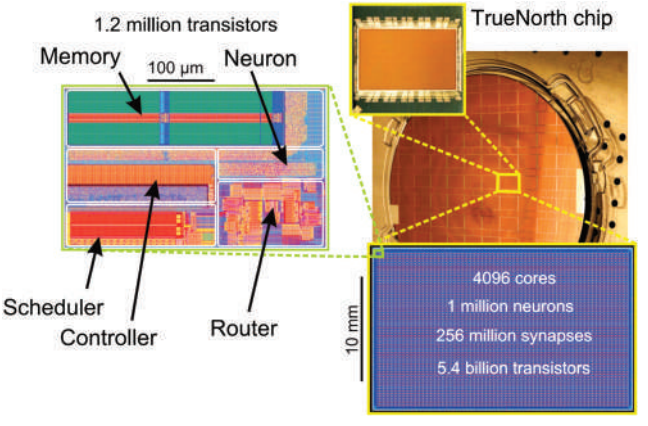

Figure 16. The IBM SyNAPSE TrueNorth neurosynaptic processor (120). *Right:* TrueNorth chip layout, silicon wafer, and chip photograph. *Left*: Layout of one of the 4096 cores. Limited connectivity across cores is provided through address-event representation (AER) asynchronous communication between cores using a mesh topology. TrueNorth offers a throughput of 1 G synaptic events per second (SynEPS) at 26 pJ of energy consumption per synaptic event (120).

spiking neurons with linear accumulate-and-fire neural dynamics using a 256-neuron finite-state machine on a self-timed clock and a 256×256 binary synapse crossbar array. Binary synapses in common for a neuron output are scaled by an individually programmable 8 bit factor to provide greater dynamic range in synaptic connectivity. TrueNorth also provides for additional functions at neuronal level, such as different noise sources and different neural activation functions that can be instantiated when programming the neurosynaptic core (122).

TrueNorth supports full connectivity across cores. Although each neuron is limited to making external synaptic connections to just one other core at a limited distance on the 2D mesh, the mesh can be expanded to accommodate larger numbers of neurons replicating the source neuron and thus allowing larger fan-out over greater distances (120). The trade-off between efficiency and flexibility in TrueNorth can be exploited to realize large-scale neural networks that can be discriminatively trained using either supervised learning (120) or samples from generative models implementing Boltzmann machines (123).

3.4. Neurogrid

Neurogrid uses multilevel digital distribution and multilevel analog aggregation, two key strategies for minimizing cortical wiring, to build a neuromorphic system with billions of synaptic connections. The difficult problem of mapping the brain's three-dimensional wiring onto a silicon chip's two-dimensional surface is alleviated by sharing wires among a population of silicon neurons – instead of dedicating a wire to each neuron (1, 124). The traffic on these shared wires, collectively called the flat address-event bus (Section 1.3), is given by the total number of synaptic connections times the silicon–neuron population's average spike-rate; each address-event signals the arrival of a spike at a particular synapse (125). The address-event bus has been successfully used to build networks with a thousand neurons and a few hundred connections per neuron (33), but it has not been scaled beyond millions of synaptic connections, the point at which the bus' signaling rate becomes saturated. To relieve this bottleneck, Neurogrid emulates multilevel distribution in axonal arbors and multilevel aggregation in dendritic arbors.

Multilevel distribution and aggregation minimize the number of axonal fibers in a nerve tract by shortening axonal branches in two different ways, both translating directly into traffic savings in the address-event bus that emulates the nerve tract. Multilevel distribution moves branch points as close as possible to the axonal terminals (Figure 17a, *top*), replacing multiple branch segments with a single fiber at a higher level in a hierarchical branching pattern (126). Neurogrid emulates this strategy by using routers to interconnect silicon–neuron arrays in a tree-like (rather than a mesh-like) fashion, allowing address-events to be replicated close to their target arrays (38), hereby cutting traffic at the tree's root by a factor equal to the number of copies that arrive at its leaves (Figure 17a, *bottom*). Multilevel aggregation moves the axon's terminals as close as possible to its branch points by extending dendritic branches (Figure 17b, *top*). The number of dendritic branches required is minimized by having each one summate signals from many axons (126). Neurogrid emulates this strategy by modeling multiple overlapping dendritic trees using a single two-dimensional resistive grid, thereby cutting traffic at the tree's root by a (compound) factor equal to the number of neighboring neurons that receive input (Figure 17b, *bottom*). The resistive grid replicates the linear transformation of postsynaptic potentials triggered by the axons' bouton clusters (127) into currents delivered to the dendrites' trunks.

Emulating the brain's hierarchical branching patterns enables Neurogrid to accommodate columnarly organized cortical networks with thousands of synaptic connections per neuron efficiently (128–133). Each cortical area is modeled by a group of *Neurocore*s, with each of its cell layers (or cell types) mapped onto a different Neurocore's two-dimensional silicon neuron array (Figure 17c). Circular pools of neurons centered at the same (x, y) location on these Neurocores model a cortical column (134). Intercolumn axonal projections are routed by using the presynaptic neuron's address to retrieve the target columns' centers from an off-chip random-access memory (first distribution level). This RAM is programmed to replicate the neocortex's function-specific intercolumn connectivity (135, 136). Intracolumn axonal branches are routed by copying the address-event to all of a cortical area's Neurocores using the interchip tree network. Unneeded copies are filtered using an on-chip RAM (second distribution level). This RAM is programmed to replicate the neocortex's stereotyped intracolumn connectivity (137). Finally, pool-spanning dendritic branches – arborizing over a circular disk centered on the cell body – are realized using the resistive grid mentioned earlier (multilevel aggregation). An elegant transistor-level implementation (52) makes it possible to adjust the grid's space constant electronically to match the pool's radius. For instance, a spike may be routed to 10 columns, copied to each of those columns' six layers, and evoke postsynaptic potentials in a 100 neighboring neurons (i.e., a 5.6 neuron radius) in all but one layer, making a total of 5000 synaptic connections.

With 16 Neurocores, Neurogrid is able to simulate cortical networks with up to a million neurons organized in up to 16 different cell layers and connected by billions of synaptic connections in real time (Figure 17d). It achieves a record-breaking 20 T flop/J on this task – five orders of magnitude better energy efficiency than a PC (129). The average energy each synaptic activation consumes is minimized by following the principle: *Amortize the cost of longer distance communications, which are more energetically expensive, over a greater number of synapses*.

3.5. HiAER-IFAT

Hierarchical address-event routing (HiAER) (37) provides a multiscale tree-based extension of AER synaptic routing (Section 1.3) for dynamically reconfigurable long-range synaptic connectivity in neuromorphic computing systems. By distributing random-access addressing of synaptic events at multiple scale levels in a tree-based connection hierarchy, HiAER offers both flexibility and expandability in synaptic connectivity at both local and global levels.

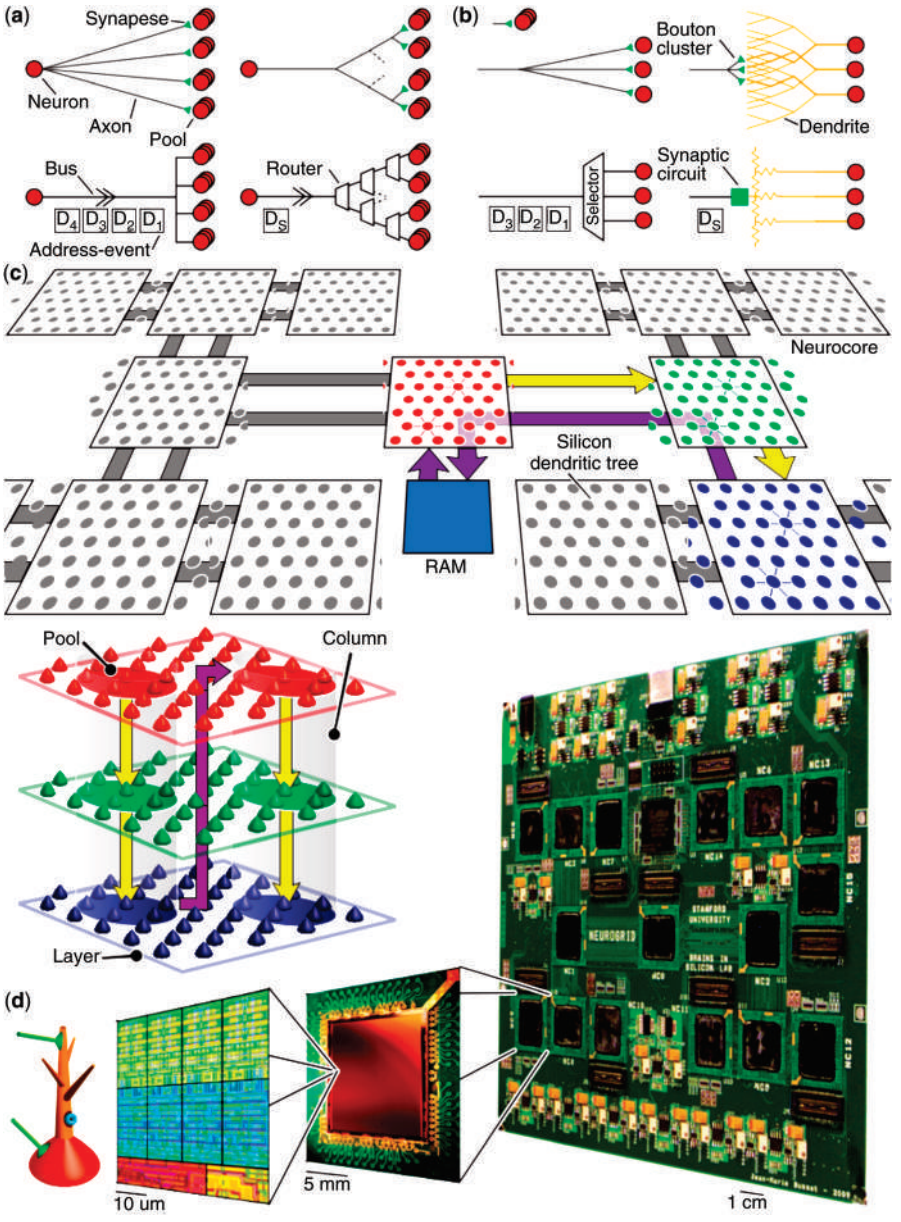

Figure 17. Modeling the cortex on Neurogrid. (a) Multilevel distribution (a) and multilevel aggregation (b) in neural (*top*) and neuromorphic (*bottom*) networks: The traffic on a digital bus that emulates spike distribution by an axonal arbor is reduced by mimicking axonal and dendritic branching patterns. (c) Mapping cortical columns: Cell layers (red, green, and blue), intercolumn connections (purple), and intracolumn connections (yellow) are mapped onto a different Neurocores, off-chip RAM (on a daughterboard), and on-chip RAM (in each Neurocore), respectively. (d) (from left to right) *Silicon neuron:* Models four ligand-gated and four voltage-gated ion channel populations, a dendrite compartment, and a soma compartment (color-coded in the schematic cell). *Neurocore:* Holds a 256×256 silicon neuron array as well as routing circuitry for inter-Neurocore communication. *Neurgrid:* Holds 16 Neurocores, connected in a binary tree network, and can simulate up to a million of neurons connected by billions of synapses in real time. (Created by Ben Varkey Benjamin.)

Queueing theory results show that HiAER offers scalable synaptic event throughput, independent of neural network size, for given synaptic fan-out and nominal axonal delay, with no spatial restrictions on synaptic connectivity (37). Another distinguishing feature of HiAER is that its synaptic connections encode not only programmable synaptic strength (probability of presynaptic release and postsynaptic conductance) but also programmable axonal delay, implemented in the timing of events routed from source to destination.

The HiAER synaptic event routing infrastructure serves as a communication backbone to integrate-and-fire array transceivers (IFAT) (33, 138, 139) and other event-driven spiking neural network hardware systems (36, 140–142). The HiAER-IFAT hardware system illustrated in Figure 18 integrates HiAER reconfigurable synaptic

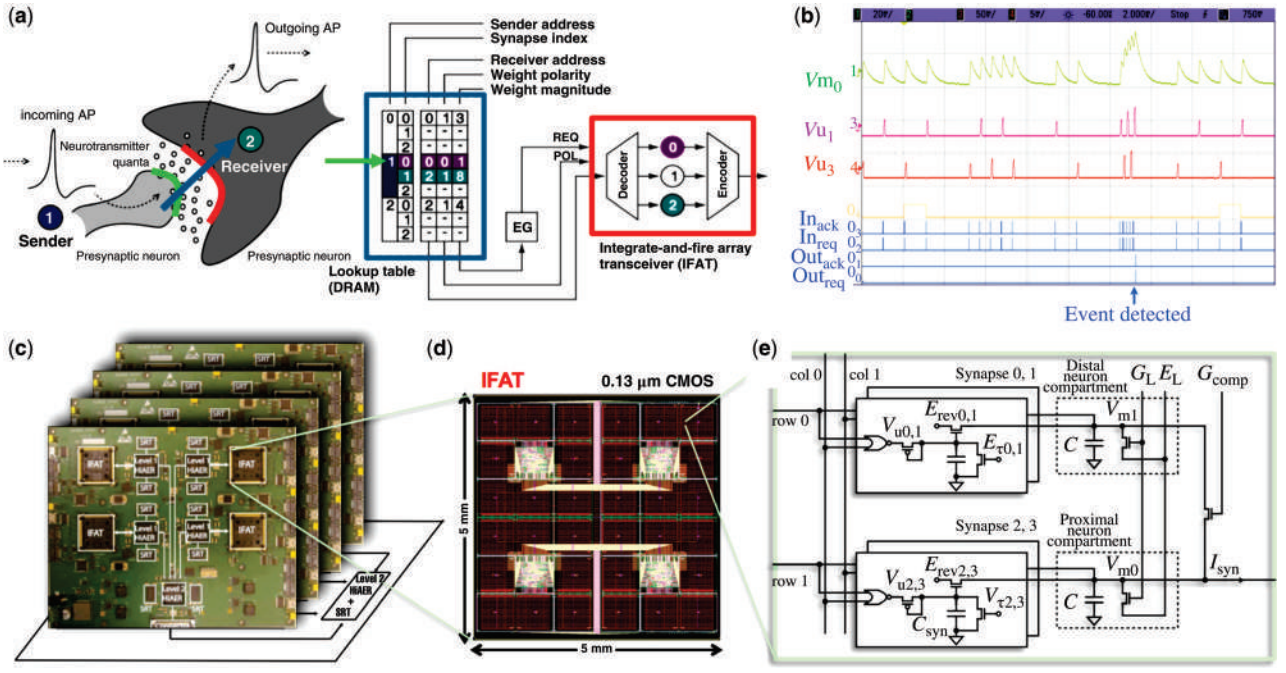

Figure 18. Integrate-and-fire array transceiver (IFAT) with hierarchical address-event routing (HiAER) synaptic connectivity for scalable and reconfigurable neuromorphic neocortical processing (139). (a) Dynamic reconfigurable synaptic connectivity across IFAT arrays of addressable neurons is implemented by routing neural spike events through DRAM synaptic routing tables (SRT). Only the level 1 (L1) leaf node in HiAER synaptic connectivity is shown for simplicity. (b) The IFAT neural array multiplexes and integrates incoming spike synaptic events ($\mathrm{In_{ack}}$, $\mathrm{In_{req}}$) generating continuous-time analog dynamics of synaptic (V_u) and neural (V_m) variables to produce outgoing spike neural events ($\mathrm{Out_{ack}}$, $\mathrm{Out_{req}}$). (c) Full-size HiAER-IFAT network with four boards, each with four IFAT modules, serving 1M neurons and 1G synapses, and spanning four levels in connection hierarchy. Each IFAT chip module comprises a 65k-neuron Tezzaron 130 nm CMOS IFAT microchip. (d) Xilinx Spartan-6 FPGA (level 1 HiAER), and two 2 Gb DDR3 SDRAM SRTs serving 65M synapses. (e) Each neural cell models conductance-based membrane dynamics in proximal and distal compartments for synaptic input with programmable axonal delay, conductance, and reversal potential. IFAT chip-measured energy consumption is 48 pJ per spike event (139), several orders of magnitude more efficient than emulation on CPU/GPU platforms.

routing, implemented using FPGAs and DRAM, with IFAT event-driven conductance-based continuous-time neural dynamics implemented in custom, very-large scale, low-power, mixed-signal integrated circuits (139, 143).

Each quadruple set of HiAER level 1 nodes (leaves in the hierarchy) shares one Xilinx Spartan 6 FPGA (XC6SLX45T) with two 2 Gb DDR3 DRAMs (Micron MT41J128M16) for synaptic routing table (SRT) storage. Four such units are provided on the board, along with an extra unit serving four HiAER level 2 nodes, as shown in Figure 18. The nodes across the FPGAs are interconnected through L1 bus parallel communication links as shown. Each FPGA is also equipped with a local 200 MHz clock generator, an external clock input, and USB and JTAG ports for diagnostics and programming. An additional 200 MHz master clock generator can provide all 4 + 1 HiAER nodes with a global clock. The system interfaces to the outside, at HiAER level 3, through the L2 bus. Several boards can be combined to form a spike-based neuromorphic computer with 262 144 analog integrate-and-fire neurons and high-speed peripherals using different variants of address-event routing protocols (94, 95, 144). At the single-board level, approximately linear throughput scaling has been demonstrated for global synaptic event routing at 36 million synaptic events per second (synEPS) per 16k-neuron node in the hierarchy (145).

Each IFAT chip has four independent ports, each with 16k two-compartment integrate-and-fire neurons (139, 143) and assigned to a single HiAER level 1 node. The IFAT neural array transceives incoming synaptic spike events to outgoing neural spike events generated through the internal, analog, continuous-time dynamics of synaptic and neural state variables (Figure 18) (139). Internally continuous-time analog, but externally asynchronous digital, the IFAT interfaces directly with HiAER to emulate large-scale biophysical models of cortical neural dynamics with reconfigurable synaptic connectivity (145). Each neuron in the IFAT array models two (proximal and distal) membrane compartments, of which one is excitable for spike generation and event registration. Coupled to each of the two compartments are two independent types of conductance-based synapses that are dynamically instantiated through the time-multiplexing of HiAER synaptic input events. Programmable control over synaptic reversal potentials and conductance time constants provides for nonlinear pooling functions in shunting inhibition and temporal coding in synchrony detection (143) – elements that are crucial to spike-based neural computation but are missing from the simplified linear integrate-and-fire models that are most commonly implemented in analog or digital neuromorphic VLSI.

4. EXAMPLE APPLICATIONS

4.1. Event-Driven Vision

The foundations of current (conventional) machine vision date back more than 150 years to Muybridge's early use of the *camera obscura* to produce a single image and capture the first "movie." The perception that visual motion is smooth and continuous when viewed above a certain frame rate is, however, more related to characteristics of the human eye and brain than to the quality of the acquisition and encoding of the visual information as a series of images. Whatever frame rate is set, frames are inadequate for visual computation pertaining to change or motion because they are unrelated to the scene's dynamic: There is no relation whatsoever between the dynamics present in a scene and the frame rate chosen to control the pixel's data acquisition process. Oversampling or undersampling will inevitably occur, and, moreover, both will usually happen at the same time. The scene will be under- and oversampled at the same time because all the pixels in an image sensor share a common timing source that controls exposure for all of them.

Neuromorphic event-driven artificial vision takes us beyond the widespread, ingrained belief that acquiring series of images at a given rate is a good way to capture visual motion (146, 147). As explained in Section 2.1, each pixel adapts its own sampling rate to the visual scene. Despite the disadvantages of increased pixel size and reduced fill factor, the advantages of acquiring dynamic vision data in this way, that is, ultrahigh-speed operation combined with reduced power consumption, transmission bandwidth, and memory requirements, actually extend beyond the acquisition stage. All subsequent processing benefits from the fact that the sensors encode visual dynamics into spatiotemporal patterns of "events," representing the relevant features of motion such as moving object contours and trajectories virtually in continuous time.

The mathematics used to describe features in space and time are simple and elegant (148–152), yielding highly efficient algorithms and computational rules that allow the real-time operation of sensory processing systems while minimizing demand for computing – and, consequently, for electrical power. No event-based theory yet exists with deductive power analogous to the z-transform familiar to conventional Nyquist-based signal processing. But thanks to the increased temporal resolution of their acquisition and encoding process, event-based spatiotemporal patterns show a high degree of orthogonality between seemingly similar features that can be exploited through the simultaneous processing of event clouds in the frequency and time domains. As a result, demanding machine vision tasks such as real-time 3D reconstruction (148, 153, 154), complex multiobject tracking (150, 155), or fast visual feedback loops for sensory–motor action (156, 157) can run at kilohertz rates on cheap, battery-powered processing hardware and allow "always-on" visual input for user interaction. Environmental context awareness on smart mobile devices, which is currently prohibited due to high power consumption requirements, also becomes possible.

To appreciate the impact a new "event-driven neuromorphic" paradigm of vision acquisition and processing could have on machine vision, it is interesting to consider the historical background on which the present-day machine vision devices are based. Like every scientific field, conventional computer vision relies on assumptions.

The first assumption, as already briefly discussed, is that dynamic scenes are observed using a stroboscopic acquisition that produces a collection of static images or frames, where the frame rate is selected high enough to encompass the frequencies of interest. Consequently, images are now the core element in the whole field of machine vision and, basically, everything is designed to acquire, operate on, and display frames. Temporal resolution is usually fixed and scene independent, there is an abundance of redundant information, and dynamic range (the difference between the darkest and the brightest pixel) is limited due to the same fixed exposure time to which every pixel in the sensing array is subjected.

The second assumption of conventional computer vision is that luminance, in the form of gray levels or colors acquired as absolute values of light intensity, is the main source of information. Luminance is so omnipresent in the field that it is used for every visual task and in every application, including stereo vision, motion estimation, recognition, and navigation.

The unnatural acquisition of both relevant and nonrelevant data, following the criteria that the fixed sample rate must be sufficient to capture the highest frequency content in the scene, means that today's conventional machine vision unavoidably leads to a trade-off between latency and power: Low latency can only be achieved by increased power consumption.

The higher temporal resolution and sparseness of output offered by event-driven neuromorphic vision (63–65, 72) open a window on new aspects of visual computation by drawing attention to space–time domains (151, 158). Event-driven computation makes it possible to build a bridge between the computational and biological worlds, both of which process data using the temporal properties and arrival times of spikes or events. The role played by time in vision has been known to biologists and computational scientists for decades, but event-based computation makes it possible to derive new mathematical approaches without needing to use precise neuron modeling (46). Recent developments in event-based visual computation show that several computationally ill-posed problems can be rewritten within the time framework and lead to new methodologies and real-time implementations at frequencies of hundreds of kilohertz (148, 149, 153, 154, 159), whereas many frame-based techniques can reach only 25 Hz even on full desktop PC processors. The real advantage – and the main difficulty – of event-based computation is that it addresses vision problems in an incremental framework where each incoming event leads to a small computation.

Another advantage of using time is that any computation based on precisely timed events can be expressed as time coincidence of events, as in neural computation. Even in the case of luminance, as used in the DVS or ATIS camera (see Section "Combined DVS Luminance"), correlations can be expressed as a set of three coincidences between pixels (65). This is because luminance is encoded

in time and thus any correlation rewritten in time spaces can essentially be reduced to the detection of three coincidences between two pixels. The same applies to stereo vision: When high temporal resolution is available, matching merely means detecting coincidences of activations between pixels (148, 153, 154). If two pixels from two cameras emit events at the same time, they are most probably observing the same 3D point. A series of recent papers has shown that the use of precisely timed information on change detection casts new light on problems such as tracking (155, 160–163), SLAM (Simultaneous Localization and Mapping) (164, 165), object recognition (152, 166), time-to-contact (167), and optical flow (146, 149, 159, 168). jAER is an open-source software project that shows how many of these methods can be applied to DVS and DAVIS output (169). Machine vision has never really pursued this field of research because it has focused on a technology in which frames were the main source of information. More importantly, event-driven computation gives machine vision the status of a true science like physics, which observes a natural phenomenon, models it, and provides mathematical solutions that can lead to an implementation. Events from a neuromorphic event-driven camera and those from a biological retina are very similar. The fact that this approach differs so greatly from a pure engineering approach based on ad hoc solutions may explain why advances in machine vision over the last few years have resulted almost exclusively from the application of deep learning data regression technology.

4.2. Event-Driven Audio

The spiking outputs of event-based silicon cochleae have been used in a number of conventional audio tasks such as source localization, speaker identification, and audio digit recognition. In source localization, spike timing is used by an event-driven source localization algorithm that estimates the interaural timing difference (ITD) from the arrival time differences of the spikes generated by the two ears of a binaural cochlea. Since the resolution of the spike times is less than 1 μs, a spike-timing localization algorithm using the spike times produces a minimum spatial angular resolution similar to the minimum resolution obtained by cross-correlating signals from the two microphones (170). However, the computational load of the spike-timing localization algorithm can be up to 40× lower than that of the cross-correlation algorithm on the two analog microphone signals (82).

In speaker identification and digit recognition research, a machine learning classifier such as a support vector machine is used on the input features extracted from the spike trains (e.g., the spike interspike intervals and the individual cochlea channel firing rates) (171). Although the performance figures for these simple features are currently inferior to those of state-of-the-art signal processing methods in the audio tasks listed above, the advantage of using event-based sensors and networks lies in the shorter latency of the classified output. In a classification process using a fully connected feed-forward network, for example, answers have been reported after an average of four input spikes (172).

Outputs from spiking cochlea have also been combined with spiking outputs from DVS or other retina sensors for tasks such as guiding a robot (173). Currently, the use of deep networks for classification with these spikes is actively ongoing. Results have already shown that these networks are more forgiving of the nonidealities of silicon spikes and can be trained to extract useful information from the outputs of event-based sensors. The results in Reference 174, obtained using the MNIST database of handwritten digits and the TIDIGITs audio database, show that a high degree of accuracy can be obtained in digit recognition performance (>98%) using a deep network. This compares to accuracy values of around 64% using only 1000 spikes from a spiking retina and 83% from cochlea spikes. This suggests that high accuracy from event-based sensors will also be possible in the near future (174).

4.3. Industrial Successes

The field of neuromorphic systems engineering is still in a development stage, with researchers trying to better understand neuroscientific principles by mimicking them in artificial man-made artifacts. Nonetheless, a number of specific developments have found applications in practical scenarios.

One of them is the neuromorphic low-power image positioning sensor (175), which is presently used in space applications for studying solar flares (176).

Another successful commercial application of a neuromorphic vision sensor is the Logitech pointing device/mouse, which uses a neuromorphic image sensor without any moving mechanical parts (177).

Event-driven dynamic vision sensors such as those described in Sections “Motion DVS Retinas” and “Combined DVS Luminance” have been marketed to the public for research and development since 2009 (178), and more recently since 2015 targeting wider markets (179). A commercial application of this technology is in prosthetic devices for the blind (180).

BIBLIOGRAPHY

1. C. Mead. *Analog VLSI and Neural Systems*. Addison-Wesley, 1989.
2. C. Mead. *Proc. IEEE* **1990**, *78*, pp 1629–1636.
3. S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas. *Event-Based Neuromorphic Systems*. John Wiley & Sons, Inc.: New York, 2015.
4. K. A. Boahen. Retinomorphic Vision System, in *Proc. of the Microelectronics for Neural Networks (MicroNeuro'96);* Lausanne, 1996; pp 2–14.
5. K. Boahen. *IEEE Trans. Circuits Syst. II* **2000**, *47*, pp 416–434.
6. S. R. Deiss, R. J. Douglas, and A. M. Whatley. In *Pulsed Neural Networks*, Maass, W.; Bishop, C. M., Eds.; MIT Press, 1998; pp 157–178.
7. J. Lazzaro, J. Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie. *IEEE Trans. Neural Netw.* **1993**, *4*, pp 523–528.
8. M. Mahowald. VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function. PhD Dissertation, Caltech, Pasadena, 1992.

9. A. Mortara, E. A. Vittoz, and P. Venier. *IEEE J. Solid-State Circuits* **1995**, *30*, pp 660–669.
10. W. McCulloch and W. Pitts. *Bull. Math. Biophys.* **1943**, *5*, pp 115–133.
11. F. Rosenblatt. *Psychol. Rev.* **1958**, *65*, pp 386–408.
12. P. J. Werbos. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences; 1975.
13. M. Riesenhuber and T. Poggio. *Nat. Neurosci.* **1999**, *2*, 1019–1025.
14. G. E. Hinton, S. Osindero, and Y. W. Teh. *Neural Comput*. **2006**, *18*, pp 1527–1554.
15. A. Krizhevsky, I. Sutskever, and G. E. Hinton. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, pp 1097–1105.
16. D. W. Tank and J. J. Hopfield. *IEEE Trans. Circuits Syst.* **1986**, *33*, pp 533–541.
17. M. Cohen and S. Grossberg. *IEEE Trans. Syst. Man Cybern.* **1983**, *13*, pp 815–826.
18. W. Maass and C. M. Bishop. *Pulsed Neural Networks*. MIT Press, 2001.
19. S. Thorpe, D. Fize, and C. Marlot. *Nature* **1996**, *381*, pp 520–522.
20. J. Perez-Carrasco, et al. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, pp 2706–2719.
21. M. A. Mahowald and C. A. Mead. *Sci. Am.* **1991**, *264*, pp 76–82.
22. G. Indiveri, et al. *Front. Neurosci.* **2011**, *5*, pp 1–23.
23. R. Brette and W. Gerstner. *J. Neurophysiol.* **2005**, *94*, pp 3637–3642.
24. E. M. Izhikevich. *Cereb. Cortex* **2007**, *17*, pp 2443–2452.
25. E. M. Izhikevich. *IEEE Trans. Neural Netw.* **2004**, *15*, pp 1063–1070.
26. R. Naud, N. Marcille, C. Clopath, and W. Gerstner. *Biol. Cybern.* **2008**, *99*, pp 335–347.
27. K. M. Hynna and K. Boahen. *Neural Comput*. **2007**, *19*, pp 327–350.
28. P. Livi and G. Indiveri. *Proc. IEEE Int. Symp. Circuits Syst.* 2009. doi: 10.1109/ISCAS.2009.5118408.
29. S. Millner, A. Grubl, K. Meier, J. Schemmel, and M.-O. Schwartz. *Adv. Neural Inf. Process. Syst.* **2010**, *23*, pp 1642–1650.
30. A. van Schaik, C. Jin, and T. J. Hamilton. A log-domain implementation of the Izhikevich neuron model, in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS'10);* 2010; pp 4253–4256.
31. A. van Schaik, et al. A Log-Domain Implementation of the Mihalas–Niebur Neuron Model, in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS'10);* 2010; pp 4249–4252.
32. C. Bartolozzi and G. Indiveri. *Neural Comput*. **2007**, *19*, pp 2581–2603.
33. R. J. Vogelstein, U. Mallik, J. T. Vogelstein, and G. Cauwenberghs. *IEEE Trans. Neural Netw.* **2007**, *18*, pp 253–265.
34. S. A. Bamford, A. F. Murray, and D. J. Willshaw. *IEEE Trans. Neural Netw.* **2010**, *21*, pp 286–304.
35. P. Merolla, J. Arthur, B. Shi, and K. Boahen. *IEEE Trans. Circuits Syst. I* **2007**, *54*, pp 301–311.
36. R. Serrano-Gotarredona, et al. *IEEE Trans. Neural Netw.* **2009**, *20*, pp 1417–1438.
37. S. Joshi, S. Deiss, M. Arnold, Y. J. Park, and G. Cauwenberghs. Scalable Event Routing in Hierarchical Neural Array Architecture with Global Synaptic Connectivity, in *Proc. of the 12th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA);* February 2010.
38. P. Merolla, J. Arthur, R. Alvarez, J.-M. Bussat, and K. Boahen. *IEEE Trans. Circuits Syst. I* **2014**, *61*, pp 820–833.
39. S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown. *IEEE Trans. Comput.* **2013**, *62*, pp 2454–2467.
40. C. Zamarreño-Ramos, A. Linares-Barranco, T. Serrano-Gotarredona, and B. Linares-Barranco. *IEEE Trans. Biomed. Circuits Syst.* **2013**, *7*, pp 82–102.
41. G. Cauwenberghs and M. Bayoumi, Eds. *Learning on Silicon*. Kluwer Academic Publishers, 1999.
42. B. Raman, J. Hertz, K. Benkstein, and S. Semancik. *Anal. Chem.* **2008**, *80*, pp 8364–8371.
43. R. J. Vogelstein, F. Tenore, R. Philipp, M. S. Adlerstein, D. H. Goldberg, and G. Cauwenberghs. *Adv. Neural Inf. Process. Syst.* **2003**, *15*, pp 1171–1178.
44. N. Qiao, et al. *Front. Neurosci.* **2015**, *9*, p 141.
45. E. Neftci, S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs. *Front. Neurosci.* **2014**, *7*, 272.
46. X. Lagorce and R. Benosman. *Neural Comput*. **2015**, *27*, pp 2261–2317.
47. C. A. Mead and M. Mahowald. *Neural Netw*. **1988**, *1*, pp 91–97.
48. E. Culurciello, R. Etienne-Cummings, and K. A. Boahen. *IEEE J. Solid-State Circuits* **2003**, *38*, pp 281–294.
49. J. A. M. Olsson and P. Hafliger. Two Color Asynchronous Event Photo Pixel, in *Proc. of the IEEE International Symposium on Circuits and Systems;* 2008; pp 2146–2149.
50. D. B. Fasnacht and T. Delbruck. *IEEE Int. Symp. Circuits Syst.* **2007**, *2007*, pp 3091–3094.
51. P. Sterling and J. B. Demb, *The Synaptic Organization of the Brain*, Shepherd, G. M., Ed.; Oxford University Press: New York, 1998.
52. A. G. Andreou and K. A. Boahen. *Analog Integr. Circuits Signal Process.* **1996**, *9*, pp 141–166.
53. K. A. Boahen and A. G. Andreou. *Adv. Neural Inf. Process. Syst.* **1992**, *4*, pp 762–772.
54. K. Zaghloul and K. Boahen. *IEEE Trans. Biomed. Eng.* **2004**, *51*, pp 657–666.
55. K. Zaghloul and K. Boahen. *IEEE Trans. Biomed. Eng.* **2004**, *51*, pp 667–675.
56. J. Costas-Santos, T. Serrano-Gotarredona, R. Serrano-Gotarredona, and B. Linares-Barranco. *IEEE Trans. Circuits Syst. I* **2007**, *54*, 1444–1458.
57. J. A. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco. *IEEE Trans. Circuits Syst. I* **2010**, *57*, pp 2632–2643.
58. M. Barbaro, P. Y. Burgi, A. Mortara, P. Nussbaum, and F. Heitger. *IEEE J. Solid-State Circuits* **2002**, *37*, pp 160–172.
59. M. Gottardi, N. Massari, and S. A. Jawed. *IEEE J. Solid-State Circuits* **2009**, *44*, pp 1582–1592.
60. P. F. Ruedi, et al. *IEEE J. Solid-State Circuits* **2003**, *38*, pp 2325–2333.
61. O. Landolt, A. Mitros, and C. Koch. Visual Sensor with Resolution Enhancement by Mechanical Vibrations, in *Proc. of the 19th Conference on Advanced Research in VLSI;* Salt Lake City, UT, Mar. 2001; pp 249–264.

62. J. Kramer. *IEEE Trans. Circuits Syst II* **2002**, *49*, pp 612–628.
63. P. Lichtsteiner, C. Posch, and T. Delbruck. *IEEE J. Solid-State Circuits* **2008**, *43*, pp 566–576.
64. T. Serrano-Gotarredona and B. Linares-Barranco. *IEEE J. Solid-State Circuits* **2013**, *48*, pp 827–838.
65. C. Posch, D. Matolin, and R. Wohlgenannt. *IEEE J. Solid-State Circuits* **2011**, *46*, pp 259–275.
66. R. Berner, C. Brandli, M. Yang, S.-C. Liu, and T. Delbruck. A 240×180 10 mW 12 µs latency sparse-output vision sensor for mobile applications, in *Proc. of the Symposium on Very Large Scale Integration (VLSI);* Kyoto, Japan, 2013; pp C186–C187.
67. C. Li, C. Brandli, R. Berner, H. Liu, M. Yang, S.-C. Liu, and T. Delbruck. An RGBW Color VGA Rolling and Global Shutter Dynamic and Active-Pixel Vision Sensor, in *International Image Sensor Workshop (IISW'15);* Vaals, The Netherlands; 2015.
68. D. Kim, Z. Fu, J. H. Park, and E. Culurciello. *IEEE Trans. Electron Devices* **2009**, *56*, pp 2586–2593.
69. J. Kim, M. Lee, H. J. Shim, R. Ghaffari, H. R. Cho, D. Son, Y. H. Jung, M. Soh, C. Choi, S. Jung, K. Chu, D. Jeon, S.-T. Lee, J. H. Kim, S. H. Choi, T. Hyeon, and D.-H. Kim. *Nat. Commun.* **2014**, *5*, 5747.
70. D. Kim and E. Culurciello. *IEEE Sensors J*. **2013**, *13*, pp 2102–2108.
71. F. Pardo, J. A. Boluda, and F. Vegara. *IEEE J. Solid-State Circuits* **2015**, *50*, pp 786–798.
72. C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck. *Proc. IEEE* **2014**, *102*, pp 1470–1484.
73. R. F. Lyon and C. Mead. *IEEE Trans. Acoustics Speech Signal Process.* **1988**, *36*, pp 1119–1134.
74. R. Sarpeshkar, R. F. Lyon, and C. Mead. *Analog Integr. Circuits Signal Process.* **1998**, *16*, pp 245–274.
75. A. van Schaik, E. Fragnière, and E. Vittoz. *Adv. Neural Inf. Process. Syst.* **1996**, *8*, pp 671–677.
76. L. Watts, D. Kerns, R. Lyon, and C. Mead. *IEEE J. Solid-State Circuits* **1992**, *27*, pp 692–700.
77. E. Fragniere. A 100-Channel Analog CMOS Auditory Filter Bank for Speech Recognition, in *ISSCC Digest of Tech. Papers;* 2005; pp 140–589.
78. T. Hamilton, C. Jin, A. van Schaik, and J. Tapson. *IEEE Trans. Biomed. Circuits Syst.* **2008**, *2*, pp 30–43.
79. B. Wen and K. Boahen. *IEEE Trans. Biomed. Circuits Syst.* **2009**, *3*, pp 444–455.
80. A. R. Palmer and I. J. Russell. *Hearing Res*. **1986**, *24*, pp 1–15.
81. A. Katsiamis, E. Drakakis, and R. Lyon. *IEEE J. Solid-State Circuits* **2009**, *44*, pp 1006–1022.
82. S.-C. Liu, A. van Schaik, B. A. Minch, and T. Delbruck. *IEEE Trans. Biomed. Circuits Syst.* **2014**, *8*, pp 453–464.
83. M. Yang, C. H. Chien, T. Delbruck, and S. C. Liu. A 0.5 V 55µW 64×2-channel binaural silicon cochlea for event-driven stereo-audio sensing, in *2016 IEEE International Solid-State Circuits Conference (ISSCC);* San Francisco, CA, 2016; pp 388–389.
84. S.-C. Liu and T. Delbruck. *Curr. Opin. Neurobiol.* **2010**, *20*, pp 1–8.
85. J. Lazzaro and C. Mead. *Proc. Natl. Acad. Sci.* **1989**, *86*, pp 9597–9601.
86. J. Lazzaro and C. Mead. *Neural Comput*. **1989**, *1*, pp 47–57.
87. C. Hartmann, J. Linde, S. Dosen, D. Farina, L. Seminara, L. Pinna, M. Valle, and M. Capurro. Towards Prosthetic Systems Providing Comprehensive Tactile Feedback for Utility and Embodiment, in *Biomedical Circuits and Systems Conference (BioCAS);* 2014 IEEE, October; pp 620–623, doi: 10.1109/BioCAS.2014.6981802.
88. J. Konstantinova, A. Jiang, K. Althoefer, P. Dasgupta, and T. Nanayakkara. *IEEE Sensors J*. **2014**, *14-8*, pp 2490–2501.
89. A. Del Prete, F. Nori, G. Metta, and L. Natale. Control of Contact Forces: The Role of Tactile Feedback for Contact Localization, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS);* 2012; pp 4048–4053, doi: 10.1109/IROS.2012.6385803.
90. Z. Kappassov, J. A. Corrales, and V. Perdereau. *Robot. Auton. Syst.* **2015**, *74*, pp 195–220.
91. A. Roncone, M. Hoffmann, U. Pattacini, and G. Metta. Learning peripersonal space representation through artificial skin for avoidance and reaching with whole body surface, in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS);* 2015.
92. V. J. Lumelsky, M. S. Shur, and S. Wagner. *IEEE Sensors J*. **2001**, *1*, doi: 10.1109/JSEN.2001.923586.
93. R. S. Dahiya, G. Metta, M. Valle, A. Adami, and L. Lorenzelli. *Appl. Phys. Lett.* **2009**, *95*, pp 034105-3.
94. H. K. O. Berge and P. Hafliger. High-Speed Serial AER on FPGA, in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS'07);* 2007; pp 857–860.
95. C. Zamarreno-Ramos, T. Serrano-Gotarredona, and B. Linares-Barranco. *IEEE Trans. Biomed. Circuits Syst.* **2012**, *6*, pp 486–497.
96. P. M. Ros, M. Crepaldi, A. Bonanno, and D. Demarchi. Wireless Multi-Channel Quasi-Digital Tactile Sensing Glove-Based System, in *Euromicro Conference on Digital System Design (DSD);* Sept 4–6, 2013; pp 673–680, doi: 10.1109/DSD.2013.76.
97. S. Caviglia, M. Valle, and C. Bartolozzi, Asynchronous, Event-Driven Readout of POSFET Devices for Tactile Sensing, in *Proc. of the 2014 IEEE International Symposium on Circuits and Systems (ISCAS);* 2014; pp 2648–2651, doi: 10.1109/ISCAS.2014.6865717.
98. B. Ache and J. Young. *Neuron* **2005**, *48*, pp 417–430.
99. S. Marco and A. Gutierrez-Galvez. *IEEE Sensors J*. **2012**, *12*, pp 3189–3214.
100. R. Beccherelli, E. Zampetti, S. Pantalei, M. Bernabei, and K. Persaud. *Sensors Actuators B Chem*. **2010**, *146*, pp 446–452.
101. T. Dickinson, J. White, J. Kauer, and D. Walt. *Nature* **1996**, *382*, pp 697–700.
102. J. Gonzalez-Jimenez, J. Monroy, and J. Blanco. *Sensors* **2011**, *11*, pp 6145–6164.
103. K. Persaud. *IEEE Sensors J*. **2012**, *12*, pp 3108–3112.
104. M. Schmuker, T. Pfeil, and M. Nawrot. *Proc. Natl. Acad. Sci.* **2014**, *111*, pp 2081–2086.
105. N. Imam, T. Cleland, R. Manohar, P. Merolla, J. Arthur, F. Akopyan, and D. Modha. *Front. Neurosci.* **2012**, *6*. 83.
106. H. Markram. *Sci. Am.* **2012**, *306*, pp 50–55.
107. P. Kogge. *IEEE Spectr*. **2011**, *48*, pp 48–54.
108. E. Painkras, L. A. Plana, J. D. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, S. B. Furber. *IEEE J. Solid-State Circuits* **2013**, *48*, pp 1943–1953.
109. A. D. Brown, S.B. Furber, J.S. Reeve, J.D. Garside, K.J. Dugan, L.A. Plana, and S. Temple. *IEEE Trans. Comput.* **2015**, *64*, pp 1769–1782.
110. S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana. *Proc. IEEE* **2014**, *102*, pp 652–665.

111. A. Davison, D. Brüderle, J. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger. *Front. Neuroinf.* **2008**, *2*, pp 1–10.
112. T. C. Stewart, B. Tripp, and C. Eliasmith. *Front. Neuroinform.* **2009**, *3*, pp 1–9.
113. http://facets.kip.uni-heidelberg.de (accessed Sept. 1, 2016).
114. http://brainscales.kip.uni-heidelberg.de (accessed Sept. 1, 2016).
115. http://electronicvisions.github.io/hbp-sp9-guidebook (accessed Sept. 1, 2016).
116. T. Pfeil, A. Grübl, S. Jeltsch, E. Müller, P. Müller, M. A. Petrovici, et al. *Front. Neurosci.*, **2013**, *7*, 11.
117. J. Schemmel, J. Fieres, and K. Meier. Wafer-Scale Integration of Analog Neural Networks, in *IEEE International Joint Conference on Neural Networks (IJCNN'08);* IEEE World Congress on Computational Intelligence; June, 2008; pp 431–438.
118. S. Millner, A. Hartel, J. Schemmel, and K. Meier. Towards Biologically Realistic Multi-Compartment Neuron Model Emulation in Analog VLSI, in *Proc. of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN);* 2012.
119. S. Friedmann, N. Frémaux, J. Schemmel, W. Gerstner, and K. Meier. *Front. Neurosci.* 2013, doi: arXiv:1303.6708.
120. P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha. *Science* **2014**, *345*, pp 668–673.
121. D. S. Modha. *Introducing a Brain-inspired Computer: TrueNorth's Neurons to Revolutionize System Architecture;* IBM Research, 2014. Available at http://www.research.ibm.com/articles/brain-chip.shtml.
122. A. S. Cassidy, P. Merolla, J. V. Arthur, S. K. Esser, B. Jackson, R. Alvarez-icaza, P. Datta, J. Sawada, T. M. Wong, V. Feldman, A. Amir, D. B. dayan Rubin, E. Mcquinn, W. P. Risk, and D. S. Modha. Cognitive Computing Building Block: A Versatile and Efficient Digital Neuron Model for Neurosynaptic Cores, in*Proc. of the International Joint Conference on Neural Networks (IJCNN);* IEEE, 2013.
123. S. Das, B. Pedroni, P. Merolla, J. Arthur, A. Cassidy, D. Modha, G. Cauwenberghs, and K. Kreutz-Delgado. Gibbs Sampling with Low-Power Spiking Digital Neurons, in *IEEE International Symposium on Circuits and Systems;* 2015.
124. M. A. Mahowald and R. Douglas. *Nature* **1991**, *354*, pp 515–518.
125. M. Mahowald. *An Analog VLSI System for Stereoscopic Vision*, vol. *265;* Springer, 1994.
126. D. Chklovskii, et al. *Neuron* **2004**, *43*, pp 609–618.
127. T. Binzegger, R. Douglas, and K. Martin. *J. Neurosci.* **2007**, *27*, pp 12242–12254.
128. B. V. Benjamin, J. V. Arthur, P. Gao, P. Merolla, and K. Boahen. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* **2012**, *2012*, pp 771–774.
129. B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen. *Proc. IEEE* **2014**, *5*, pp 699–716.
130. S Choudhary, et al. Silicon Neurons That Compute, in *Artificial Neural Networks and Machine Learning (ICANN'12);* 2012; pp 121–128.
131. P. Gao, B. V. Benjamin, and K. Boahen. *IEEE Trans. Circuits Syst. I* **2012**, *59*, pp 2383–2394.
132. S. Menon, S. Fok, A. Neckar, O. Khatib, and K. Boahen. *Biomed. Robot. Biomechatron.* **2014**, *2014*, pp 181–186.
133. R. Silver, K. Boahen, S. Grillner, N. Kopell, and K. L. Olsen. *J. Neurosci.* **2007**, *27*, pp 11807–11819.
134. V. B. Mountcastle. *J. Neurophysiol.* **1957**, *20*, pp 408–434.
135. D. H. Hubel and T. N. Wiesel. *Proc. R. Soc. Lond. B Biol. Sci.* **1977**, *198*, pp 1–59.
136. D. D. Stettler, A. Das, J. Bennett, and C. D. Gilbert. *Neuron* **2002**, *36*, pp 739–750.
137. T. Binzegger, R. Douglas, and K. Martin. *J. Neurosci.* **2004**, *24*, pp 8441–8453.
138. D. H. Goldberg, G. Cauwenberghs, and A. G. Andreou. *Neural Netw.* **2001**, *14*, pp 781–793.
139. T. Yu, J. Park, S. Joshi, C. Maier, and G. Cauwenberghs. 65k-neuron integrate-and-fire array transceiver with address-event reconfigurable synaptic routing, in *IEEE Biomedical Circuits and Systems Conference (BioCAS);* 2012; pp 21–24.
140. G. Indiveri, A. M. Whatley, and J. Kramer. A Reconfigurable Neuromorphic VLSI Multi-Chip System Applied to Visual Motion Computation, in *Proc. of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems;* 1999; pp 37–44.
141. S.-C. Liu and R. Douglas. *IEEE Trans. Neural Netw.* **2004**, *15*, pp 1305–1314.
142. D. Sridharan, B. Percival, J. Arthur, and K. A. Boahen. *Adv. Neural Inf. Process. Syst.* **2008**, *20*, pp 1401–1408.
143. T. Yu, J. Park, S. Joshi, C. Maier, and G. Cauwenberghs. *Conf. Proc. IEEE Eng. Med. Biol. Soc*. 2012; pp 775–778.
144. D. B. Fasnacht, A. M. Whatley, and G. Indiveri. A Serial Communication Infrastructure for Multi-Chip Address Event Systems, in *Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS'08);* 2008; pp 648–651.
145. J. Park, T. Yu, C. Maier, S. Joshi, and G. Cauwenberghs. Live Demonstration: Hierarchical Address-Event Routing Architecture for Re-Configurable Large Scale Neuromorphic Systems, in *Proc. of the IEEE International Symposium on Circuits and Systems;* 2012; pp 707–711.
146. T. Delbrück. Frame-Free Dynamic Digital Vision, in *Proc. of the International Symposium on Secure-Life Electronics;* University of Tokyo, Tokyo, Japan, 2008; pp 21–26.
147. C. Posch, R. Benosman, and R. Etienne-Cummings. *IEEE Spectr.* **2015**, *52*(12), pp 44–49.
148. R. Benosman, S. H. Ieng, P. Rogister, and C. Posch. *IEEE Trans. Neural Netw. Learn. Syst.* **2011**, *22*, pp 1723–1734.
149. R. Benosman, S.-H. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan. *Neural Netw*. **2012**, *27*, pp 32–37.
150. C. Clady, S. H. Ieng, and R. Benosman. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *66*, pp 91–106.
151. X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. Benosman. HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition, *IEEE Trans. Pattern Recognit. Mach. Intell.* 2016. Available at http://ieeexplore.ieee.org/document/7508476/.
152. G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, and R. Benosman. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, pp 2028–2040.
153. J. Carneiro, S. H. Ieng, C. Posch, and R. Benosman. *Neural Netw*. 2013, pp 27–38.
154. P. Rogister, R. Benosman, S. Leng, P. Lichtsteiner, and T. Delbrück. *IEEE Trans. Neural Network. Learn. Syst.* **2012**, *23*, pp 347–353.

155. X. Lagorce, C. Meyer, S. H. Ieng, D. Filliat, and R. Benosman. *IEEE Transactions Neural Netw. Learn. Syst.* **2015**, *26*, pp 1710–1720.
156. J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. J. Douglas, and T. Delbruck. A pencil balancing robot using a pair of AER dynamic vision sensors, in *Pro. of the IEEE International Symposium on Circuits and Systems;* Taipei, 2009; pp 781–784.
157. T. Delbrück and M. Lang. *Front. Neurosci.* **2013**, *7*, pp 223.
158. X. Lagorce, S. H. Ieng, X. Clady, M. Pfeiffer, and R. B. Benosman. *Front. Neurosci.* **2015**, *9*, 46.
159. R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 407–417.
160. Drazen, D., Lichtsteiner, P., Häfliger, P., Delbrück, T., and Jensen, A. *Exp. Fluids* **2011**, *51*, pp 1465–1469.
161. M. Litzenberger, C. Posch, D. Bauer, A. N. Belbachir, P. Schon, B. Kohn, et al. Embedded Vision System for Real-Time Object Tracking Using an Asynchronous Transient Vision Sensor, in *12th Signal Processing Education Workshop; 4th Digital Signal Processing Workshop;* 2006; pp 173–178, doi: 10.1109/DSPWS.2006.265448.
162. G. Gritsch, M. Litzenberger, N. Donath, and B. Kohn. Real-Time Vehicle Classification Using a Smart Embedded Device with a 'Silicon Retina' Optical Sensor, in *Proc. of the 11th International IEEE Conference on Intelligent Transportation Systems;* 2008; pp 534–538.
163. M. Litzenberger, B. Kohn, A. N. Belbachir, N. Donath, G. Gritsch, H. Garn, et al. Estimation of Vehicle Speed Based on Asynchronous Data from a Silicon Retina Optical Sensor, in *Proc. of the IEEE Conference on Intelligent Transportation Systems (ITSC);* 2006; pp 653–658, doi: 10.1109/ITSC.2006.1706816.
164. E. Mueggler, B. Huber, and D. Scaramuzza. Event-Based, 6-DOF Pose Tracking for High-Speed Maneuvers, in *Proc. of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS);* 2014; pp 2761–2768, doi: 10.1109/IROS.2014.6942940.
165. D. Weikersdorfer, D. B. Adrian, D. Cremers, and J. Conradt. Event-Based 3D SLAM with a Depth-Augmented Dynamic Vision Sensor, in *2014 IEEE International Conference on Robotics and Automation (ICRA);* 2014; pp 359–364, doi: 10.1109/ICRA. 2014.6906882.
166. P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer. *Front. Neurosci.* **2013**, *7*, p 178.
167. X. Clady, C. Clercq, S.-H. Ieng, F. Houseini, M. Randazzo, L. Natale, et al. *Front. Neurosci.* **2014**, *8*, p 9.
168. B. Rueckhauer and T. Delbruck. *Front. Neurosci.* **2016**, *10*, 176.
169. jAER Open Source Project. Available at http://jaerproject.org (accessed March 6, 2016).
170. H. Finger and S. C. Liu. Estimating the Location of a Sound Source with a Spike-Timing Localization Algorithm, in *IEEE International Symposium on Circuits and Systems (ISCAS);* Rio de Janeiro, 2011; pp 2461–2464.
171. C. H. Li, T. Delbruck, and S.-C. Liu. Real-Time Speaker Identification Using the AEREAR2 Event-Based Silicon Cochlea, in *Proc. of the IEEE International Symposium on Circuits and Systems;* Seoul, Korea (South), 2012: pp 1159–1162.
172. D. Neil and S. C. Liu. *IEEE Trans. Very Large Scale Integr. Syst.* **2014**, *22*, pp 2621–2628.
173. V. Chan, C. T. Jin, and A. van Schaik. *Front. Neurosci.* **2012**, *6*, pp 1–9.
174. D. Neil and S. C. Liu. Effective Sensor Fusion with Event-Based Sensors and Deep Network Architectures, in *Proc. of the IEEE International Symposium on Circuits and Systems;* May 2016.
175. K. Strohbehm, et al. *Proc. of the 13th National Solar Observatory/Sacramento Peak Summer Workshop;* **1993**; vol *1*.
176. P. Bernasconi, et al. Flare Genesis Experiments. http://sd-www.jhuapl.edu/FlareGenesis/flare.html.
177. X. Arreguit, A. van Schaik, F. V. Bauduin, M. Bidiville, and E. Raeber. *IEEE J. Solid-State Circuits* **1996**, *31*, pp 1916–1921.
178. http://www.inilabs.com (accessed March 6, 2016).
179. http://www.chronocam.com/ (accessed March 6, 2016).
180. http://www.pixium-vision.com (accessed March 6, 2016).

C. Bartolozzi

Istituto Italiano di Tecnologia, Genoa, Italy

R. Benosman

Pierre and Marie Curie University, Paris, France

K. Boahen

Stanford University, Stanford, CA, USA

G. Cauwenberghs

University of California San Diego, La Jolla, CA, USA

Tobi Delbrück, Giacomo Indiveri and Shih-Chii Liu

ETHZ, Zurich, Switzerland

S. Furber

University of Manchester, Manchester, UK

N. Imam

Cornell University, Ithaca, NY, USA

Bernabé Linares-Barranco, Teresa Serrano-Gotarredona

Instituto de Microelectrónica de Sevilla (IMSE-NM), CSIC and University of Seville, Sevilla, Spain

K. Meier

University of Heidelberg, Heidelberg, Germany

C. Posch

Chronocam Ltd., Paris, France

M. Valle

University of Genoa, Genova, Italy