CAVIAR: A 45k Neuron, 5M Synapse, 12G Connects/s AER Hardware Sensory–Processing–Learning–Actuating System for High-Speed Visual Object Recognition and Tracking

Rafael Serrano-Gotarredona, Matthias Oster, Patrick Lichtsteiner, Alejandro Linares-Barranco,
Rafael Paz-Vicente, Francisco Gómez-Rodríguez, Luis Camuñas-Mesa, Raphael Berner, Manuel Rivas-Pérez,
Tobi Delbrück, Shih-Chii Liu, Rodney Douglas, Philipp Häfliger, Gabriel Jiménez-Moreno, Anton Civit Ballcels,
Teresa Serrano-Gotarredona, *Member, IEEE*, Antonio J. Acosta-Jiménez, and Bernabé Linares-Barranco

Abstract—This paper describes CAVIAR, a massively parallel hardware implementation of a spike-based sensing–processing–learning–actuating system inspired by the physiology of the nervous system. CAVIAR uses the asychronous address–event representation (AER) communication framework and was developed in the context of a European Union funded project. It has four custom mixed-signal AER chips, five custom digital AER interface components, 45k neurons (spiking cells), up to 5M synapses, performs 12G synaptic operations per second, and achieves millisecond object recognition and tracking latencies.

R. Serrano-Gotarredona was with the Consejo Superior de Investigaciones Científicas, Seville Microelectronics Institute, Seville 41012, Spain. He is now with the Austriamicrosystems, Valencia, Spain (e-mail: rserrano@imse.cnm.es).

M. Oster was with the Institute of Neuroinformatics (INI), ETH Zürich/University of Zürich, Zürich CH-8057, Switzerland. He is now with the Varian Medical Systems, Baden CH-5405, Switzerland (e-mail: matthias.oster@gmail.com).

P. Lichtsteiner was with the Institute of Neuroinformatics (INI), ETH Zürich/University of Zürich, Zürich CH-8057, Switzerland. He is now with the Espros Photonics Corporation, Baar CH-6340, Switzerland (e-mail: patrick.lichtsteiner@espros.ch).

A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, M. Rivas-Pérez, G. Jiménez-Moreno, and A. Civit Ballcels are with the Computer Architecture and Technology Department, University of Seville, Seville 41012, Spain (e-mail: alinares@atc.us.es; rpaz@atc.us.es; gomezroz@atc.us.es; mrivas@us.es; gaji@atc.us.es; civit@atc.us.es).

L. Camuñas-Mesa, T. Serrano-Gotarredona, A. Acosta-Jiménez, and B. Linares-Barranco are with the Consejo Superior de Investigaciones Científicas, Seville Microelectronics Institute, Seville 41092, Spain (e-mail: luiscamu@imse.cnm.es; terese@imse.cnm.es; acojim@imse.cnm.es; bernabe@imse.cnm.es).

R. Berner, T. Delbrück, S.-C. Liu, and R. Douglas are with the Institute of Neuroinformatics (INI), ETH Zürich/University of Zürich, Zürich CH-8057, Switzerland (e-mail: bernerr@ee.ethz.ch; tobi@ini.phys.ethz.ch; shih@ini.phys.ethz.ch; rjd@ini.phys.ethz.ch).

P. Häfliger is with the Informatics, University of Oslo, Oslo NO-0316, Norway (e-mail: hafliger@ifi.uio.no).

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

Digital Object Identifier 10.1109/TNN.2009.2023653

I. INTRODUCTION

BRAINS perform powerful and fast vision processing in a way conceptually different from that of machine vision systems. Machine vision systems process sequences of still frames from a camera. For performing scale- and rotation-invariant 3-D object recognition, for example, sequences of computationally demanding operations need to be performed on each acquired frame. The computational power and speed required for such tasks make it difficult to develop real-time autonomous systems for such applications.

On the other hand, vision sensing and object recognition in brains are performed without using the "frame" concept, at least not in the usual sense of implying a fixed-rate sequence of still images. Throughout this paper, we intentionally avoid the use of the expression "image processing," because in our hardware technology, there never is an "image" or a "frame," but rather a continuous flow of visual information in the form of temporal spikes.

The visual cortex is structured as a sequence of layers (8–10 layers in the human cortex [1], [13]), starting from the retina, which does its own preprocessing in a more compact and analog architecture. Although cortex has massive feedback and recurrent connections, it is known that a very fast and purely feedforward recognition path exists within the ventral stream of the visual cortex [1], [2]. Here we exploited this feedforward path concept to build a fast vision recognition system. A conceptual block diagram of such a cortically inspired feedforward hierarchically structured autonomous system for sensing/processing/decision–actuation can be seen in Fig. 1(a) [1]–[11]. The pattern of connectivity in cortex follows a basic structure: each neuron in a layer connects to a "cluster of neurons" or "projective field" in the next layer [12], [13].

In most cases, these projective fields can be approximated by computing 2-D convolutions. A single layer of a single convolution kernel can detect and localize a preprogrammed or prelearned object, independent of its position. Using multiple

kernels of different sizes and rotations can make the computation scale and rotation invariant. Multilayered convolutional networks are capable of complex object recognition [3]–[8].

Spiking neurons receive synaptic input from other cells in the form of electrical spikes, and they autonomously decide when to generate their own output spikes. Hardware that combines spike-based multineuron modules to compute projective fields can enable powerful and fast frame-free vision processing. If the components generate short-latency meaningful, nonredundant spikes, then spike-based systems can efficiently compute "on-demand" compared to conventional approaches. The processing delay depends mainly on the number of layers, and not on the complexity of objects and shapes to be recognized. Their latency and throughput are not limited by a conventional sampling rate.

In recent years, significant progress has been made towards the understanding of the computational principles exploited by visual cortex. Many artificial systems that implement bioinspired software models use biological-like (convolution-based) processing that outperforms more conventionally engineered machines [3]–[11], [14]–[17]. However, these systems generally run at extremely low speeds because the models are implemented as software programs on conventional computers. For real-time solutions, direct hardware implementations of these models are required. However, hardware engineers face a large hurdle when trying to mimic the bioinspired layered structure and the massive connectivity within and between layers. A growing number of research groups worldwide are mapping some of these computational principles onto real-time spiking hardware through the development and exploitation of the so-called address–event representation (AER) technology. In this paper, we report on the results of our European Union consortium project "Convolution AER Vision Architecture for Real-Time" (CAVIAR), where the largest ever built multichip multilayer AER real-time frame-free vision system to date has been developed.

The purpose of this paper is to introduce to various communities, including computational neuroscience and machine vision, the promising and effective AER hardware technology that allows the construction of modular, multilayered, hierarchical, and scalable (visual) sensory processing learning and actuating systems. Throughout this paper, we will illustrate the power and potential of the AER hardware technology through the demonstrator assembled in the CAVIAR project.

The AER is a spike-based representation technique for communicating asynchronous spikes between layers of neurons in different chips. The spikes in AER are carried as addresses of sending or receiving neurons on a digital bus. Time "represents itself" as the asynchronous occurrence of the event. AER was first proposed in 1991 by Mead's Lab at California Institute of Technology (Caltech, Pasadena) [24]–[28], and has been used since then by a wide community of hardware engineers. Unarbitrated and simpler event readout have been used [29], [30], and more elaborate and efficient arbitrated versions have also been proposed, based on winner-take-all (WTA) [31], or the use of arbiter trees [32], which have evolved to row parallel [33] and burst-mode word-serial [34]–[36] readout schemes by Boahen's Lab. The AER has been used in image

and vision sensors, for simple light intensity to frequency transformations [38], time-to-first-spike codings [40]–[42], foveated sensors [43], [44], spatial contrast sensors [23], [45], temporal intensity difference [39] and temporal contrast sensors [19], [20], and motion sensing and computation systems [46]–[50]. AER has also been used for auditory systems [51]–[53], competition and WTA networks [54]–[56], and even for systems distributed over wireless networks [57]. For AER-based 2-D convolution, Vernier *et al.* [58] and Choi *et al.* [59] reported on 2-D convolution chips with hard-wired elliptic or Gabor-shaped kernels for orientation extraction. AER has made it feasible to emulate large scale neurocortical-like multilayered realistic structures since the development of scalable and reprogrammable kernel 2-D convolution chips, either with some minor restrictions on symmetry [60], or without any restrictions on shape or size [18]. Of great importance for the spread and success of AER systems has also been the availability of open-source reusable silicon IP [37], a better understanding by the community of asynchronous logic design, and the development of conventional synchronous interfacing logic and computer interfaces [61]–[64].

In CAVIAR, an AER infrastructure was developed to support a set of AER modules (chips and interfaces) [Fig. 1(b)] that are connected in series and parallel to embody the abstract layered architecture in Fig. 1(a). The following modules were developed: 1) a temporal contrast retina (motion sensing camera) chip; 2) a programmable kernel 2-D convolution processing chip; 3) a 2-D WTA object chip; 4) spatio–temporal processing and learning chips; 5) AER remapping, splitting, and merging field-programmable gate array (FPGA)-based modules; and 6) computer–AER interfacing FPGA modules for generating and/or capturing AER. These modules were then used for building a multilayer artificial vision demonstrator system for detecting and tracking balls moving at high speeds.

The overall architecture of the CAVIAR vision system is illustrated in Fig. 1(b) and in more detail in Fig. 13. Moving objects in the field of view of the retina cause spikes. Each spike from the retina causes a splat of each convolution chip's kernel onto its own integrator array. When the integrator array pixels exceed positive or negative thresholds they in turn emit spikes. In the CAVIAR system experiments, we generally used circular kernels such as the ones in Fig. 3(c) and (d), which detect circular objects of particular sizes. The resulting convolution spike outputs are noise filtered by the WTA object chip. The WTA output spikes, whose addresses represent the location of the "best" circular object, are fed into a configurable delay line chip that spreads time into space. This spatial pattern of temporal delayed spikes is then learned by the learning chip. The WTA spikes also control a mechanical or electronic tracking system that stabilizes the programmed object in the field-of-view center.

The rest of this paper is structured as follows. Section II describes the temporal contrast retina, Section III the programmable kernel 2-D convolution chip, Section IV the 2-D WTA chip, Section V the learning chips, Section VI the different interfaces, and finally, Section VII describes the complete CAVIAR vision system and shows experimental results. Section VIII concludes the paper and gives future outlooks.
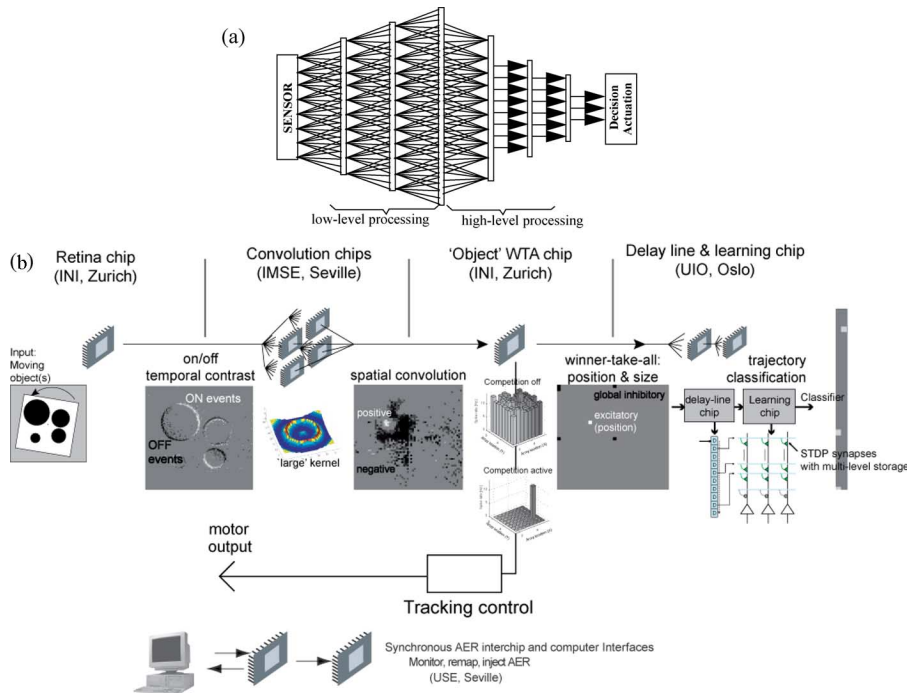
Fig. 1. CAVIAR system overview. (a) A bioinspired system architecture performing feedforward sensing + processing + actuation tends to have the following conceptual hierarchical structure: 1) a sensing layer; 2) a set of low-level processing layers usually implemented through projection fields (convolutions) for feature extraction and combination; 3) a set of high level processing layers that operate on "abstractions" and progressively compress information through, for example, dimension reduction, competition, and learning; 4) once a reduced set of signals/decisions is obtained they are conveyed to (usually mechanical) actuators. (b) The CAVIAR system components and multilayer architecture; an example output of each component is shown in response to the rotating stimulus and the basic functionality is illustrated below each chip component.

<div style="display:flex;gap:2em">

<div>

TABLE I
TEMPORAL CONTRAST VISION SENSOR PROPERTIES ADAPTED FROM [20]

Functionality	Asynchronous temporal contrast
Pixel size (μm)	40x40
Chip size (mm)	6x6
Fill factor	9.4%
Fabrication process	4M 2P 0.35μm CMOS
Pixel complexity	26 transistors (14 analog), 3 capacitors
Array size	128x128
Interface	15-bit non-greedy AER
Power consumption	24mW
Dynamic range	120dB, 2 lux to >100klux scene illumination with f/1.2 lens. Moonlight capable with high contrast scene
Response latency	15μs @ 700mW/m^2
Max events/sec	~2 Meps
Standard deviation σ of temporal contrast threshold	2.1% scene contrast

</div>

<div>

TABLE II
CONVOLUTION CHIP PROPERTIES

Functionality	Dynamic 2D Convolution with programmable kernel
Pixel size (μm)	90x90
Chip size (mm)	4.0x5.4
Fabrication process	4M 2P 0.35μm CMOS
Pixel complexity	364 transistors, 1 capacitor
Array size	32x32
max kernel size	31x31
kernel weight resolution	4 bit
calibration resolution	5 bit
Interface	15-bit word parallel non-greedy AER
Power consumption	66-150mW, depending on kernel size
forgetting rate	adjustable
Max out events/sec	25 Meps

</div>

</div>

II. AER TEMPORAL CONTRAST RETINA

The temporal contrast silicon retina is an asynchronous vision sensor that emits spike address–events (AEs) (Fig. 2 and Table I) [19], [20]. Each AE from the chip is the address k of a pixel and signifies that the log intensity at pixel k changed by an amount T since the last event from that pixel. T is a global event threshold that we typically set to about 15% contrast. In addition, one bit of the address encodes the sign of the change (ON or OFF). This representation of "change in log intensity" generally encodes scene reflectance change. The compressive logarithmic transformation in each pixel allows for wide dynamic range operation (120 dB, compared with for example, 60 dB for a high-quality traditional image sensor). This wide dynamic range means that the sensor can be used with uncontrolled natural lighting. The asynchronous response property also means that the events have a latency down to 15 μs with bright lighting and typically about 1 ms under indoor illumination, resulting in an effective frame rate of typically several kilohertz. The temporal redundancy reduction greatly reduces the output data rate for scenes in which most pixels are not changing. The design of the pixel also allows for unprecedented uniformity of response: the mismatch between pixel contrast thresholds is 2.1% contrast. The event threshold can be set down to 10% contrast, allowing the device to sense natural scenes rather than only artificial high-contrast stimuli. The vision sensor also has integrated digitally controlled biases that greatly reduce chip-to-chip variation in parameters and temperature sensitivity [21].
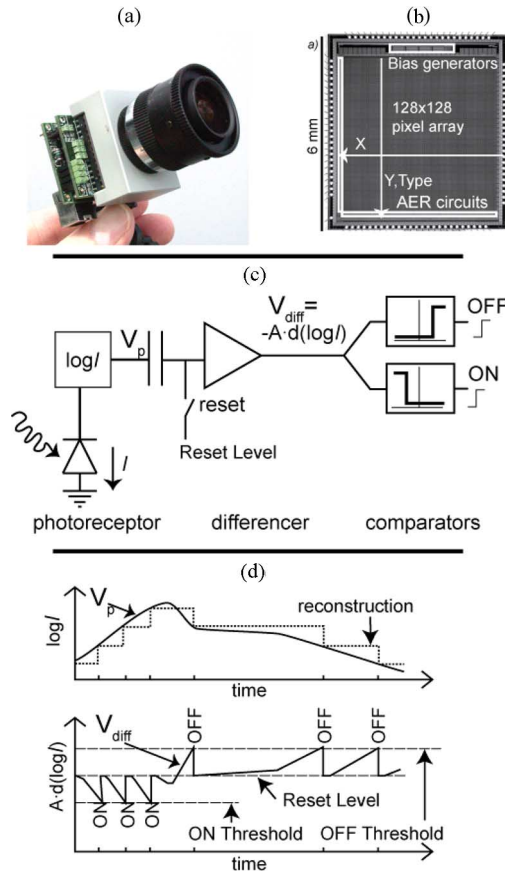
Fig. 2. Temporal contrast silicon retina vision sensor. (a) Silicon retina USB2 system. The vision sensor with its lens and USB2.0 interface. (b) Chip micrograph. A die photograph labeled with the row and column from a pixel that generates an event with x, y, type output, where type is ON or OFF. (c) Simplified pixel core schematic that responds with events to fixed-size changes of log intensity. (d) Principle of operation. How the ON and OFF events are internally represented and output in response to an input signal. Figure adapted from [20].
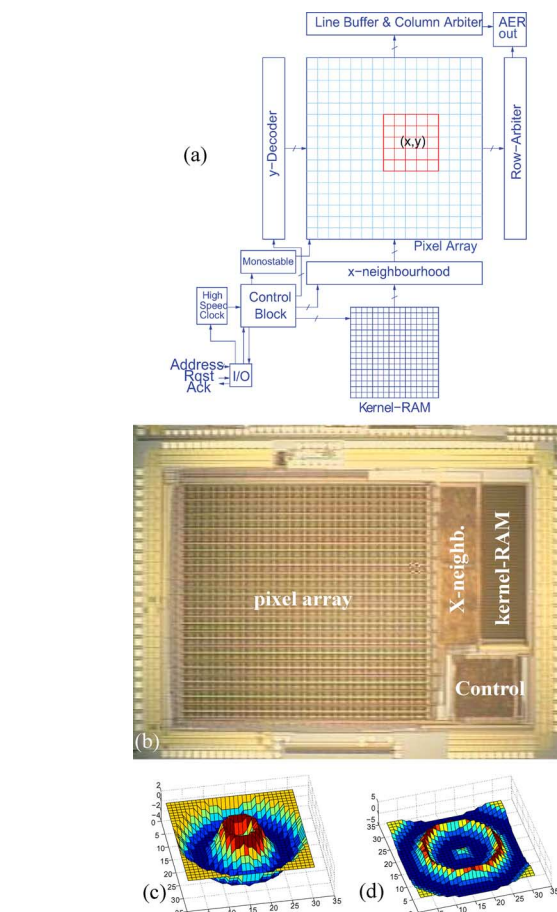

Fig. 3. Convolution chip. (a) Architecture of the convolution chip. (b) Microphotograph of fabricated chip. (c) Kernel for detecting circumferences of radius close to four pixels and (d) close to nine pixels.

III. AER PROGRAMMABLE KERNEL 2-D CONVOLUTION CHIP

The convolution chip is an AER transceiver with an array of event integrators, already reported elsewhere [18]. Table II summarizes the chip performance figures and specifications. For each incoming event, integrators within a projection field around the addressed pixel compute a weighted event integration. The weight of this integration is defined by the convolution kernel [18], [60]. Each incoming event computation splats the kernel onto the integrators.

Fig. 3(a) shows the block diagram of the convolution chip. The main parts of the chip are as follows.

1) An array of 32×32 pixels where each pixel contains a binary weighted signed current source and an integrate-and-fire signed integrator. The current source is controlled by the kernel weight read from the RAM and stored in a dynamic register for each input event.

2) A 32×32 kernel static RAM where each kernel weight value is stored with signed 4-b resolution.

3) A digital controller that handles all sequence of operations.

4) For each incoming event, a monostable generates a pulse of fixed duration that enables the integration simultaneously in all active pixels.

5) An x-neighborhood block that performs a displacement of the kernel in the x direction.

6) Arbitration and decoding circuitry that generate the output AEs and which uses Boahen's burst mode word parallel AER [33].

The chip operation sequence is as follows.

1) The digital control block stores the (x, y) address of an incoming event and acknowledges reception of the event through the Rqst and Ack signals.

2) The control block computes the x-displacement that has to be applied to the kernel and the limits in the y addresses where the kernel has to be copied.

3) The control block copies the kernel from the kernel RAM row by row to the corresponding rows in the pixel array.

4) The control block activates the generation of a monostable pulse. This way, in each pixel a current weighted by the corresponding kernel weight is integrated during a fixed time interval.

5) Kernel weights in the pixels are erased.

A pixel (Fig. 4) contains two digitally controlled pulsing current sources (pulsing CDAC) which provide a current pulse of fixed width [equal to the width of the signal "event pulse" coming from the monostable in Fig. 3(a)] and amplitude dependent on the kernel weight stored in the dynamic register
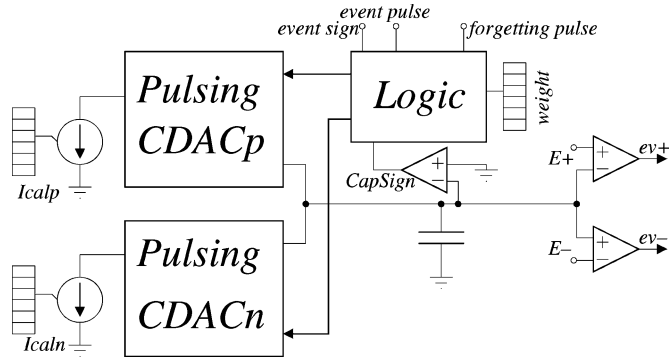
Fig. 4. Simplified block diagram of convolution chip pixel.

Fig. 5. Multichip assembly of convolution chips. All chips "see" the same input space (up to 128×128 pixels), but each chip can process only 32×32 pixels. Each pixel stores its limit coordinates ($x_{\min}, x_{\max}, y_{\min},$ and $y_{\max}$). In general, when an event is received at coordinate (x_o, y_o), up to four chips process it.

"weight" in Fig. 4. Depending on the combination of kernel weight sign and input event sign, the current pulse has to be positive (provided by CDACp) or negative (provided by CDACn). The current of each CDAC is proportional to a locally trimmable current (Icalp or Icaln) to compensate for interpixel mismatches. Calibration values are loaded from an external source over a serial interface. Current pulses are integrated onto a capacitor, whose voltage is monitored by two comparators. If an upper (lower) threshold $E+$ ($E-$) is reached, the pixel sends a positive (negative) output event, and resets the capacitor voltage to the intermediate resting level. This event is arbitrated and decoded in the periphery of the chip. In parallel, all pixels receive a periodic signal "forgetting pulse" which discharges (charges) the capacitor voltage to the intermediate resting voltage if "*CapSign*" is high (low), by generating fixed amplitude current pulses at CDACn (CDACp).

Both the size of the pixel array and the size of the kernel storage RAM are 32×32. The input address space can be up to 128×128 (14 b) and the chip is programmed to receive input from a part of this space. Fig. 3(b) shows the microphotograph of the fabricated chip. AER events can be fed-in up to a peak rate of 50 million events per second (Meps). The chip can generate output events at a maximum rate of 25 Meps. Input event throughput depends on kernel size and internal clock frequency. The event cycle time is given by $(4 + 2 \times n_k)T_{\text{clock}}$, where n_k is the number of programmed kernel lines (from 1 to 32) and T_{clock} is the internal clock period. The internal clock is tunable and could be set up to 200 MHz ($T_{\text{clock}} = 5$ ns) before observing operation degradation although in our setup we generally used 100 MHz. Maximum sustained input event throughput can, therefore, vary between 33 Meps for a one line kernel down to 3 Meps for a full 32 line kernel. Further details are given in Table II and elsewhere [18].

Each convolution chip can process an input space of up to 128×128 pixels, but can produce outputs for only 32×32 pixels. This is useful for multichip assembly. For example, Fig. 5 illustrates how an array of 4×4 chips, each with 32×32 pixels, could be used to process a visual input of 128×128 pixels. Each chip stores into internal registers its own limit coordinates $[x_{\min}, x_{\max}, y_{\min}, y_{\max}]$ within the total 128×128 pixel space. All chips share the same input AER bus (this is done in practice using AER splitters). Maximum kernel size can be 31×31 (i.e., $2s + 1 \times 2r + 1$; see Fig. 5), which means
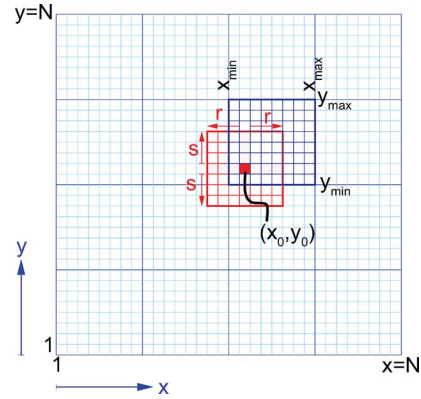
that pixels up to 30 positions apart from a chip might need to be processed by it. For example, in Fig. 5, we can see how an event with address (x_0, y_0) is processed simultaneously by four neighboring chips. The output events produced by all chips are merged on a single AER bus by an external merger.

For the vision system described in Section VII, we assembled four convolution chips on a single printed circuit board (PCB). The PCB has one AER input bus connector and one AER output bus connector. The input bus goes to a 1-to-4 splitter, implemented on a complex programmable logic device (CPLD) chip, that feeds the input AER ports of the four chips. The chips' output AER ports connect to a merger circuit, implemented on another CPLD circuit, whose output goes to the PCB output AER connector. The four chips can be programmed to "see" the same input space and each compute a different 2-D filter (convolution) on the same 32×32 pixel space, or the four chips can be programmed to process the same kernel while operating on an expanded 64×64 pixel space. In Section VII, we used this latter option, so that the PCB would work as one single convolution processor of array size 64×64, and maximum kernel size of 31×31.

IV. AER 2-D WTA CHIP

The AER WTA transceiver chip [66]–[70] is designed to simultaneously determine the "what" and "where" of the convolution chip outputs. The "whats" are the best matched features in the case of multiple convolution chips, each with a different kernel, and the "wheres" are the spatial locations of these features (Fig. 6 and Table III). The WTA chip implements this feature competition using four populations of spiking neurons which receive the outputs of individual convolution chips and computes the winner (strongest input) in two dimensions. First, it performs a WTA operation on the inputs from a feature map to determine the strongest input (which codes the location of the strongest feature in the feature map), and second, it performs a second level of WTA operation on the sparse feature maps to determine the strongest feature out of all preprogrammed features. The parameters of the network are configured so that it implements a hard WTA with only one neuron active at a time. The spike rate of the winning neuron is proportional to its input
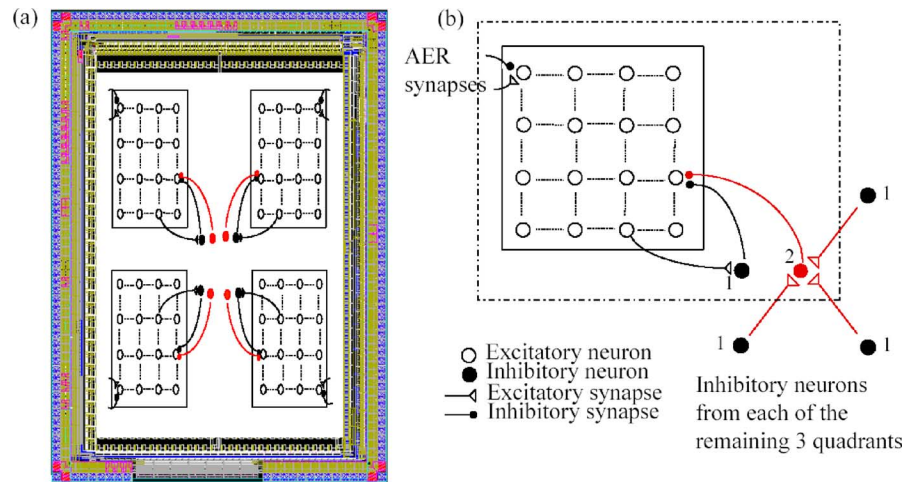
Fig. 6. Layout and architecture of WTA object chip. (a) Architecture and underlying layout of the four populations or quadrants receiving inputs from four convolution chips. (b) The architecture of one of the quadrants, each with 256 neurons. Notice that (a) does not show the inputs to the global inhibitory neurons 2.

spike rate. The 2-D WTA chip reduces the data flow rate to the learning chip by preserving only information about the locations of the best matched features. The learning chip can then, for example, track the 3-D movement of an object in space by programming the same feature shape at different sizes in the different convolution chips.

The WTA chip has an array of integrate-and-fire neurons that can be configured into four populations or quadrants of 16×16 neurons or a single population of 32×32 neurons. Fig. 6(a) shows the network architecture and the underlying layout of the four population quadrants. Each quadrant as shown in Fig. 6(b) has 254 excitatory neurons (unfilled circles) and two global inhibitory neurons. The excitatory neurons receive external AER inputs representing its feature map from a convolution chip through four input synapses, which comprise one excitatory synapse, one excitatory depressing synapse [71], and two inhibitory synapses[1] (only two of the four synapses are shown for an exemplar neuron in each quadrant in Fig. 6). The connectivity across the four quadrants can be configured to enable WTA competition within a feature map (first level of competition) and across feature maps (second level of competition). Each excitatory neuron also has two sets of local non-AER synapses which form the connections to and from global inhibitory neurons 1 and 2, and a local self-excitatory synapse. To enable WTA competition within a feature map or quadrant, the excitatory neurons in that quadrant are configured to drive their global inhibitory neuron labeled 1 [solid black circle in Fig. 6(b)], which in return inhibits these excitatory neurons. To additionally enable WTA competition across feature maps, the global inhibitory neuron labeled 2 [solid gray (red) circle in Fig. 6(b)] in a quadrant is excited by the global inhibitory neurons labeled 1 from the remaining three quadrants. In return, it inhibits all excitatory neurons within its own quadrant. The second-level competition works as follows. The activity of the global inhibitory neuron 1 in each quadrant reflects the highest input rate to the excitatory neurons within the quadrant in the case of a hard WTA competition. Thus,

---

[1]In the CAVIAR system described in this paper, we only use the excitatory synapse. The other three synapses were included for other eventual applications.

TABLE III
WTA CHIP SPECIFICATIONS

Functionality	4 quadrant WTA
Pixel size (µm)	87 x 140
chip size (mm)	3.1 x 4.8
Fabrication process	4M 2P 0.35µm CMOS
Pixel complexity	160 transistors + 6 capacitors
Array size	4x16x16 pixels x 4 dynamic synapses
Minimum detectable regular input rate difference	10%
Interface	12-bit word parallel non-greedy AER
Power consumption	118 mW
Max events/sec (I/O)	10 Meps

this neuron will activate the global inhibitory neurons 2 of the remaining three quadrants the most if its quadrant receives the highest input rate out of the four quadrants. Hence, it can indirectly suppress the activity of the excitatory neurons of the remaining quadrants through their global inhibitory neuron 2. This computation is discussed in detail in [67].

An on-chip global digital-to-analog converter (DAC) block allows us to set individual local synaptic parameters for each neuron by using part of the address space to transmit the DAC value. This block was also used to decrease the amount of mismatch across the neurons. The spiking activity of the neurons is monitored through the addresses of the output AER spikes while an on-chip scanner [72] consisting of a clocked set of shift registers allows us to monitor the membrane potentials of the neurons externally.

During the WTA operation, each excitatory input spike charges the membrane of the postsynaptic neuron until one neuron in the array—the winner—reaches threshold and is reset. All other neurons are then inhibited via the global inhibitory neuron of its population. Self-excitation provides hysteresis for the winning neuron by facilitating the selection of this neuron as the next winner.

Because of the moving stimulus, the network must determine the winner using a rapid estimate of the instantaneous input firing rates. The number of spikes that the neuron must integrate before eliciting an output spike can be adjusted by varying the
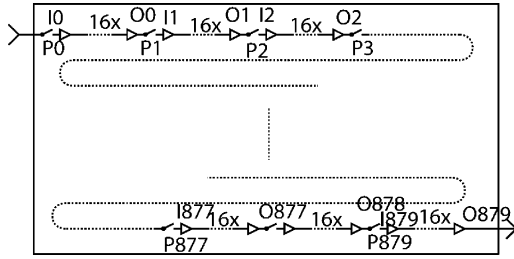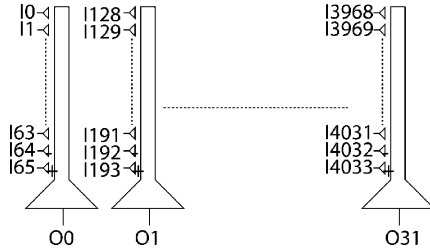
Fig. 7. Block diagram of the delay line chip.

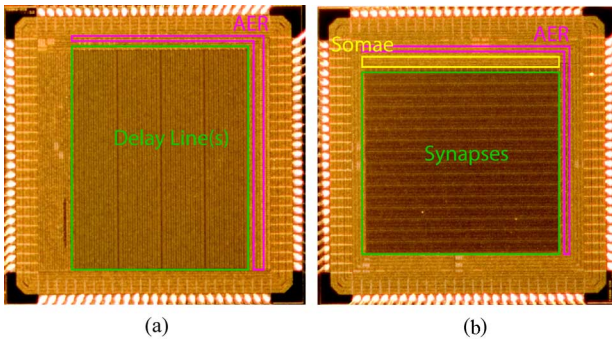Fig. 8. Block diagram of the learning chip.

Fig. 9. Die photographs of (a) delay line and (b) learning chips.

Functionality	Configurable Delay Line
Delay Element size (μm)	27 x 11
chip size (mm)	3.15 x 3.15
Element Complexity	11 transistors (1 analog), 3 capacitors
Fabrication process	4M 2P 0.35μm CMOS
Array size	1x14080 - 880x16 (configurable)
Interface	16-bit greedy AER
Power consumption	~60mW (including PCB)
delay dynamic range	6.25ns-2.5ms
delay standard deviation	17%
Max events/sec (I/O)	~10 Meps

system. The first chip expands the time axis of dynamic spatial patterns into a spatial dimension by simply generating multiple copies of the pattern at different time delays. In this AER framework, this is realized with a delay line chip. The second chip implements competitive learning to classify spatial patterns. The patterns may be spike patterns, formed by coincident spikes at different spatial locations, or activity patterns, formed by coincident average spiking activity at different locations.

A. Expanding Time Into Space

The delay line chip (Figs. 7 and 9 and Table IV) expands time into a spatial dimension. It is composed of an asynchronous delay line with 14 080 delay elements (triangles) and 880 addressable access points. Higher order bits in the address events determine if one is injecting an event $(I\ldots)$ or closing or opening a switch $(P\ldots)$. Output address events are produced at points $O\ldots$. Thus, the delay line can be programmed to be clipped at the access points into several delay lines.

Expanding time into a spatial dimension is what we do all the time as we plot graphs with a time axis. Fig. 16 illustrates this with the outputs of the retina, convolution, and WTA stages as they track a circle on a rotating disk. By representing time in space, the 2-D motion can be represented in a 3-D figure where the circular motion of the circle's center forms a spiral. The result is a snapshot of the circle trajectory within a time window.

B. From Spike-Based to Rate-Based Hebbian Learning

The learning chip (Figs. 8 and 9 and Table V) is an implementation of a spike-based timing-dependent learning rule. It contains 32 neurons, each with 64 learning synapses, and an inhibitory and excitatory nonlearning synapse. In the sense of spike-timing-dependent plasticity the chip can be tuned to increase synaptic weights when presynaptic spikes precede postsynaptic spikes within a certain time window. This time window may be prolonged and then the net behavior of the synapse becomes rate dependent, rather than spike timing dependent (given that the inputs are roughly Poisson distributed spike trains, see, for example, [73] and [74]). The learning rule as given in [73] is

$$\frac{d\vec{\omega}}{dt} = \mu y \left(\vec{c} - \frac{1}{\nu^2} \vec{\omega} \right).$$ (1)

This learning rule implicitly normalizes the weight vector to length ν, which is a free parameter, that is, any attractor of the dynamics of $\vec{\omega}$ has length ν. Another parameter is the learning

strengths of the input synapses. The neuron and synapse parameters must be programmed appropriately for the incoming input spike rate to the neurons and the duration of the input so that the network operates properly as a WTA network. The WTA circuit can reliably select the winner given a difference of input firing rate of only 10% if it receives stationary input spike trains with a regular firing rate. A quantitative analysis of the constraints on the parameter space in regards to a spiking WTA system receiving spiking inputs of various statistics is described in [66], [69], and [70]. These constraints have been derived for the condition of a hard WTA network in the case of both regular and Poisson input spike trains. Specifications of the WTA chip are summarized in Table III.

V. AER LEARNING CHIPS

The learning system in the CAVIAR processing chain supplies the ability to learn to classify spatio–temporal patterns representing the trajectory of a moving object [73]. It is capable of both spike-based learning (or spike-timing-dependent plasticity [82]) to learn to classify spatio–temporal spike patterns and rate-based Hebbian learning to learn spatio–temporal activity patterns. This is achieved in a two-chip AER processing

TABLE V
LEARNING CHIP PROPERTIES

Functionality	Spike Based Learning
Synapse size (μm)	121x16
chip size (mm)	3.15 x 3.15
Synapse Complexity	46 transistors (26 analog), 4 capacitors
Fabrication process	4M 2P 0.35μm CMOS
Array size	64x32
Interface	16-bit greedy AER
Power consumption	~60mW (including PCB)
Weight storage technology	weak 6-level static memory
Pre/Postsynaptic correlation time window	~10ns-1s
Max events/sec (I/O)	~10 Meps

rate μ. Symbol y represents the neuron's output spike train, defined as a sum of Dirac delta functions. Thus, $\vec{\omega}$ only changes at times of postsynaptic spikes. $\vec{c}$ is a dynamic variable that provides some measure of recent presynaptic activity. It is incremented with every presynaptic spike, decaying with the same time constant τ as the membrane potential of the neurons, and resetting after a postsynaptic spike. It is mainly the time constant τ in balance with the weight vector length ν that determines the transition from rate-based to spike-based learning behavior.

The on-chip weight storage is implemented with a six-level static memory cell which can be set so that the stored weight is weakly driven to the closest of six stable voltage levels within a few hundred milliseconds [75]. Short-term analog computations are stored on a capacitor at analog resolution and the result is only slowly discretized for long-term storage.

C. Competitive Hebbian Learning of Spatio–Temporal Patterns

In the following setups, the inhibitory synapse of the learning neurons is configured to receive the outputs from all other neurons, to do global cross inhibition. The neurons will thus operate as classifiers in the manner of a WTA network. Since the learning rule specifies that weight updates only happen if there is postsynaptic activity, the learning will also be competitive: only the winner will adapt its weights to respond even better to that particular stimulus. Each neuron will thus specialize on a certain input pattern and win the WTA competition if the input is closer to its pattern than to any of the other neurons' patterns. This results in a learned discretization of an input space with minimal information loss. The learning chip's output is then a compressed representation of the system input that can simplify the task of a higher level supervised learning stage (e.g., implemented in software on a computer) to control appropriate actions as a response to specific situations. This is, however, beyond the scope of this paper.

More details about the theoretical background for the learning chip's behavior are given in Appendix II and a practical example with real-world input from the CAVIAR processing chain is described in Section VII-D.

The combined two-chip system can classify spatio–temporal patterns: neurons in the learning chip can become sensitive not only to spatial patterns, but also to stimulus direction and speed.

VI. AER INTERFACES

When developing and tuning complex hierarchical multichip AER systems it is crucial to have available proper computer interfaces for reading AER traffic and visualizing it, for injecting synthesized or recorded AER traffic into AER buses, and for address transformations or remappings as events travel between chips. We developed five PCB solutions, all shown in Fig. 10(a)–(e):

A. PCI-AER

Fig. 10(a) shows the PCI-AER interfacing PCB capable of sequencing timed AER events out from a computer or *vice versa* capturing and time-stamping events from an AER bus and storing them in computer memory. It uses a Spartan-II FPGA, and can achieve a peak rate of 15 Meps and 10 Meps sustained rate, using PCI bus mastering.

B. USB-AER

Fig. 10(b) shows a photograph of the USB- AER board. Its block diagram is shown in Fig. 11. It is a codesign platform that does not require a PCI slot and can be controlled through a USB port. It uses a Spartan II 200 FPGA with a Silicon Labs C8051F320 microcontroller. Depending on the logic configuration (firmware) loaded into the FPGA, it can perform five different functions: 1) transform a sequence of frames into AER in real time (video-frames to AER converter, using rate coding) [80], 2) histogram AER events into sequences of frames in real time (AER to video-frames converter), 3) remap addresses using lookup tables (AER mapper), 4) capture and time-stamp events for offline analysis (AER data logger), and 5) reproduce time-stamped sequences of events in real time (AER data player). This board can also work without a USB connection (standalone mode) by loading the firmware through MMC/SD cards (used in commercial digital cameras). This PCB can handle AER traffic of up to 25 Meps between its two AER ports. It also includes a video graphics array (VGA) display output for visualizing histogrammed frames in real time.

The AER remapping capability is the most useful feature for assembling multimodules AER systems. A block diagram of the mapping firmware is shown in Fig. 12. This firmware can produce a sequence of up to eight output events per input event. These events are stored in the 2-MB SRAM memory of the board. The memory can be prewritten with the proper lookup table through USB. These output events are accessed through simple bit operations of the input events. The mapper is also able to decide with some probability value if an output event is produced according to a probabilistic function implemented with a linear feedback shift register (LFSR).

C. Switch-AER

The third PCB, based on a simple CPLD, is shown in Fig. 10(c). It splits one AER bus into two, three, or four buses, and *vice versa*, merges two, three, or four buses into a single bus, with proper handling of the handshaking signals.
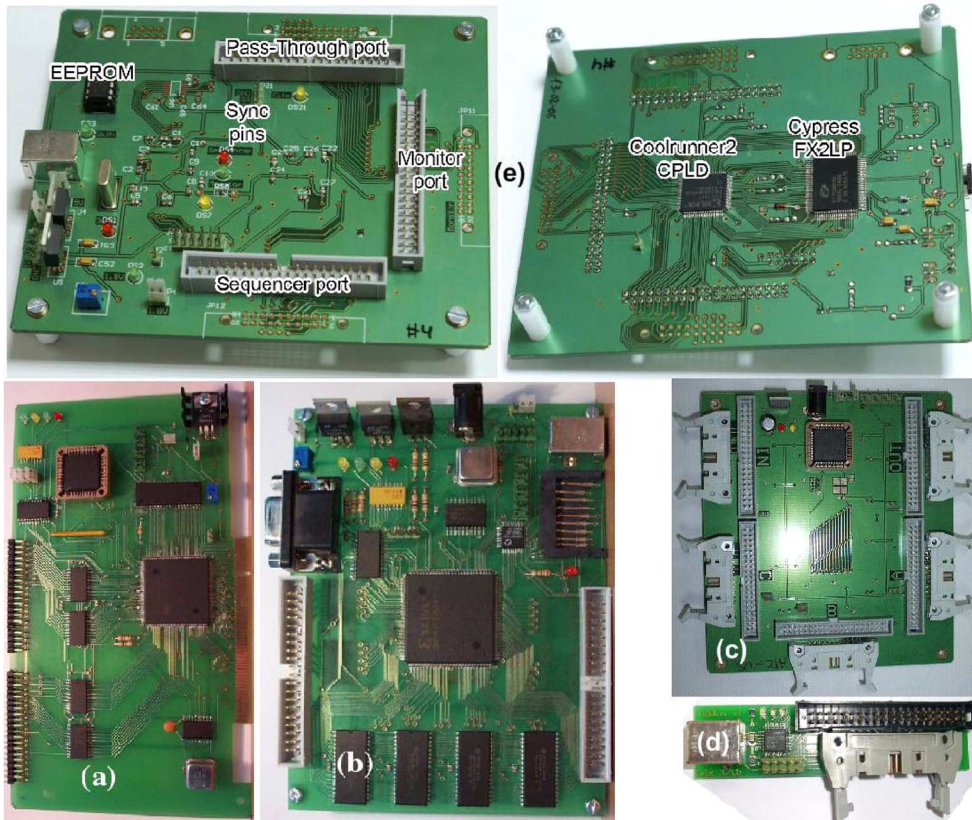
Fig. 10.   CAVIAR interfaces: (a) PCI-AER. (b) USB-AER. (c) Switch-AER. (d) Mini-USB-AER. (e) USB2AER (left: top face, right: bottom face).
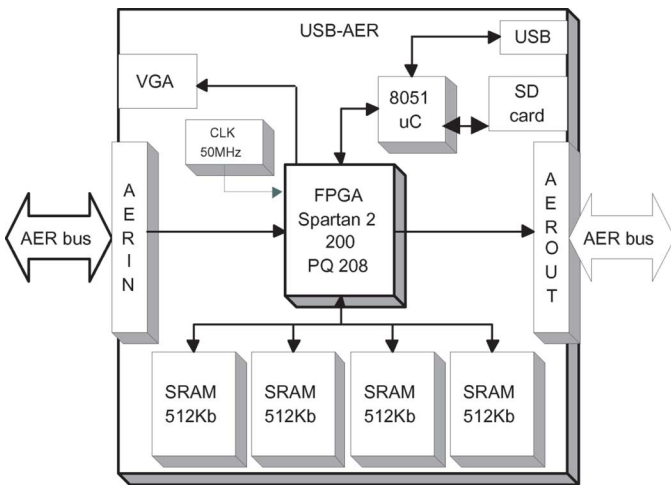

Fig. 11.   USB-AER codesign PCB block diagram. A Silicon Labs C8051F320 USB microcontroller with SD/MMC support, connected with a Xilinx Spartan 2 FPGA. The FPGA firmware is able to perform complex operations with the two AER buses, the SRAM memory, and the VGA interface.
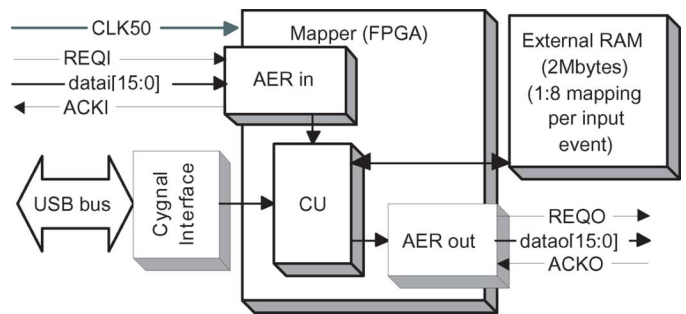

Fig. 12.   Mapper block diagram: through the USB, the mapping table is stored in the external SRAM memory and per each input event the control unit (CU) generates the sequence of up to eight probabilistic output events.

D. Mini-USB-AER

Fig. 10(d) is a lower performance but more compact bus-powered USB interface based on a C8051F320 microcontroller. It captures and time-stamps events to a computer at peak rates of up to 100 kilo events per second (keps) and is particularly useful for portable demonstrations, and the components cost is less than $20 [22].

E. USB2AER

The last board, shown in Fig. 10(e), is a bus-powered USB version of the PCI-AER board. This board supports USB 2.0 high speed, allowing event rates of 6 Meps between the computer and the AER chip both for sequencing and for monitoring purposes [64].

The next section describes the operation of the CAVIAR demonstrator system, shown in Fig. 13 where we made extensive use of the AER interfacing PCBs. The USB-AER board remapped AEs in real time. The USB2AER board is used as an AER monitor (to visualize reconstructed/histogrammed images in real time) and for storing time-stamped events in computer memory for offline analysis.
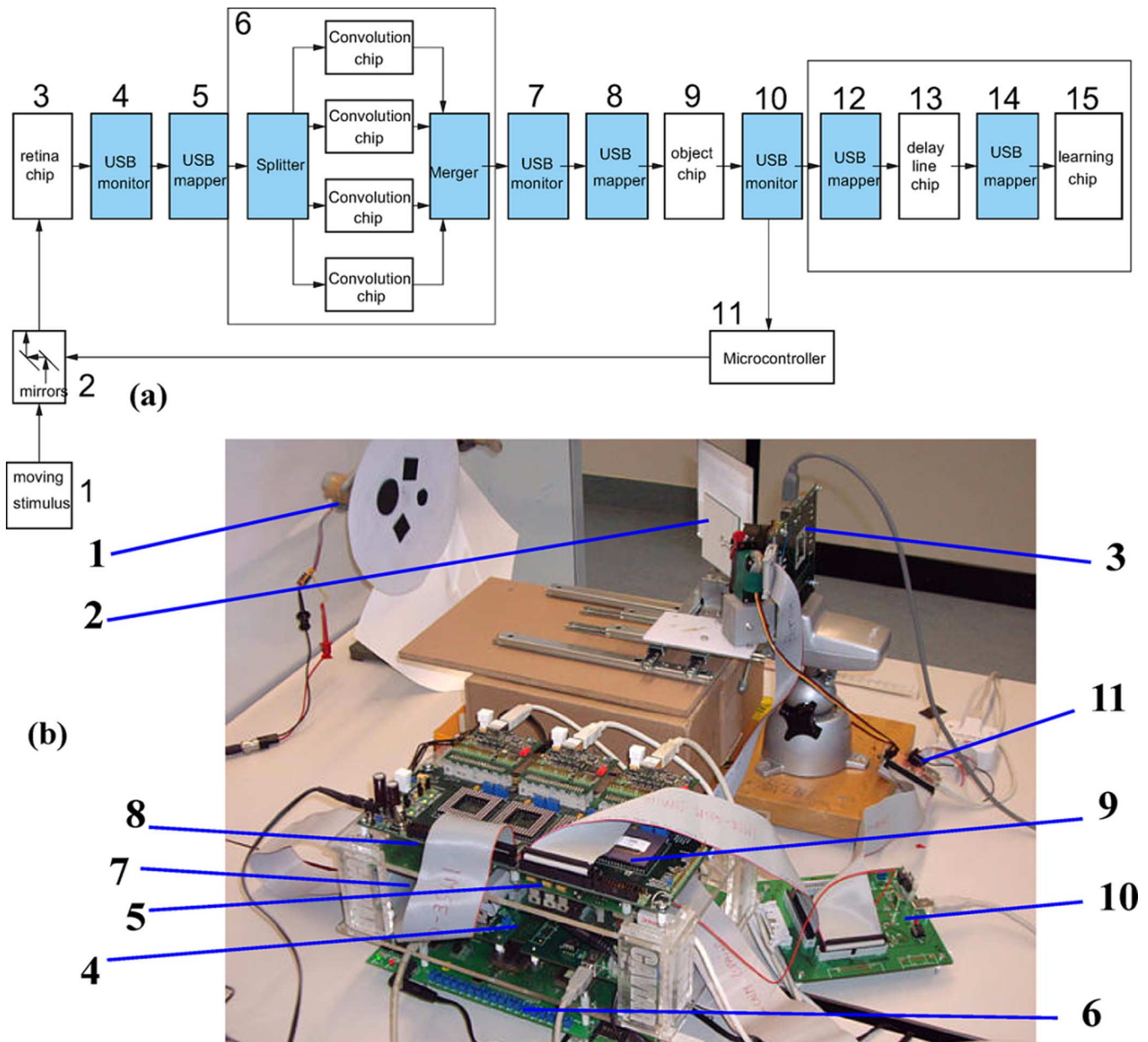
Fig. 13. Experimental setup of multilayered AER vision system for ball tracking (final demonstrator of CAVIAR FET project). (a) Block diagram (white boxes include custom designed chips and gray boxes are interfacing PCBs). (b) Photograph of a setup.

The reconfigurability of the boards was actively used during the CAVIAR project developments to test, debug, and adjust each layer of the processing separately, or two by two. For example, a repetitive input sequence of events is useful for proper tuning of biases and kernels for testing the convolution chips. For this, one can use the USBAER board (with AER data logger firmware) to capture a time-stamped sequence of events from the retina to a file. One can later reconfigure the board to act as an input to the convolution chip by repeatedly reproducing these stored events in real time (by loading the AER data-player firmware for this functionality). Often, during the CAVIAR project development, events were recorded in one lab and reproduced at a different lab. This way one partner could use the output of a previous block as their input without having this block physically present in their lab [62], [80].

VII. THE CAVIAR VISION SYSTEM

Using all these building blocks, we assembled the frame-free spike-based sensing–processing–actuating–learning AER vision system shown in Fig. 13. A mechanical rotor (1) holds a rotating white piece of paper with two circles of different radius and some distracting geometric figures. The vision system follows the two circles only, and discriminates between the two. A pair of servomotor driven mirrors changes the point of view of the AER retina (3), which sends outputs to a monitor PCB (4), and a mapper PCB (5) before reaching the convolution PCB with four convolution chips (6). The latter PCB output is sent through another monitor PCB (7) and mapper PCB (8) to the 2-D WTA "object" chip (9). This output is received by a monitor PCB (10) which sends a copy to a microcontroller (11) that controls the mirror motors (2) to center the detected circle. Another

copy of the WTA output is sent to the learning system which consists of a mapper (12), a delay line chip (13), another mapper (14), and a learning classifier chip (15), and which learns to classify trajectories into different classes.

The temporal contrast retina provides an output event space of size 128×128 pixels. The four-convolution-chip PCB can process the complete 128×128 retina space, although it would compute the convolution output for only the central 64×64 pixels. In our setup, we introduced a mapper between the retina PCB and convolution PCB (block 5 in Fig. 13) to downsample from 128×128 pixels to 64×64 pixels. This way, the convolution PCB will provide outputs for the complete retina visual range. The mapper also eliminates the retina sign bit, since the convolution stage needs to detect full circumferences. In the following experiments, we used circular kernels such as the ones in Fig. 3(c) and (d), which emphasize circumferences of particular sizes by doing template matching. Using these kernels continuously is somehow similar to performing circular Hough transforms [65], which detect the locations of circular features in the input, except that we use analog graded values and include negative kernel values for penalizing the absence of features.

The 64×64 convolution output is fed through a mapper that subsamples to a 32×32 space for the WTA (object) chip. This WTA output provides cleaned up coordinates for the center of the target-size detected circle. This signal is split at this point into two separate paths: (a) the motor control subsystem and (b) the learning subsystem.

The motor control subsystem is built using a commercial microcontroller which acts on two servomotors, each holding one mirror. One of the mirrors is tilted horizontally while controlled by the y-coordinate of the "object" chip output, and the other mirror is tilted vertically while controlled by the x-coordinate of the "object" chip output. This way, the coordinate provided by the "object" chip represents the deviation $(\Delta x, \Delta y)$ of the detected circle from center of the field of view. A proportional controller (11 in Fig. 13) uses the motor system to zero this error, thus keeping the target-size circle centered on the field of view of the WTA.

The object centering can be seen in Fig. 14. The three subfigures show the outputs captured by the three monitor PCBs in Fig. 13, when the feedback servo control is enabled to center the detected object. Monitor PCBs connect to a host computer through a high-speed USB2.0 connection, sending AEs at a speed of up to a peak rate of 6 Meps. The jAER software [22] reads those events, and reconstructs and renders 2-D histograms by collecting events belonging to a time slice which is programmable. If this time slice is similar to the monitor refresh rate, one can visualize the reconstructed images (video) in real time. For high-speed phenomena, one can configure a time slice of very short duration (down to a few microseconds) and visualize a slow-motion recorded sequence of events offline.

Fig. 14(a) shows a histogram reconstructed from the 128×128 retina output captured by monitor PCB (4) in Fig. 13. White dots represent positive sign events (dark-to-light transitions) and black dots negative sign events (light-to-dark transitions), allowing identification of the direction of motion
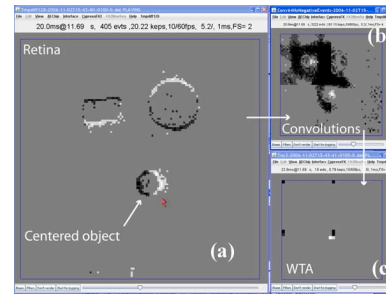

Fig. 14. The 20-ms snapshot of outputs of vision tracking system. The retina central point of view is changed dynamically to follow the small circle, which appears always centered in the field of view.
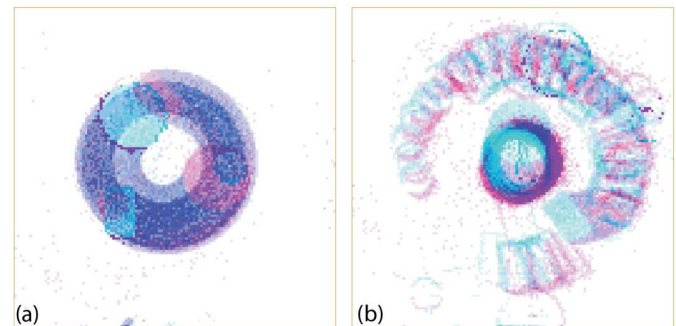

Fig. 15. Electronic centering. (a) Raw retina output over 1.3 s without centering. (b) With electronic centering using WTA to steer retina output.

of the geometric figures, which is clockwise in this case. Fig. 14(b) also shows a histogram image reconstructed from the 64×64 convolution PCB output captured by monitor PCB (7) in Fig. 13. In this case, the kernel was programmed to detect the small circle [see Fig. 3(c)]. Positive sign events (white) show where the small circle is centered, while the negative events (dark) show where it is not. The convolution output includes some noise, which is filtered out by the WTA operation. The convolution output pixels are transformed from size 64×64 to 32×32 (by grouping 2×2 pixels into one) by the mapper (8) in Fig. 13. Fig. 14(c) also shows the output of the WTA computing stage, where all noise has been filtered out. The dark pixels are the local and global inhibitory units for each quadrant.

We also implemented a fully electronic (without mirrors, mechanical parts, or motors) servo system for changing the central view point. For this system, an extra splitter was inserted at the retina output. The extra retina output together with the WTA output were fed to an extra mapper, subtracting. The result is the same as when using the mirrors, except that now the maximum rotating speed of the stimulus is not limited by the timing constraints of the mechanical components and the object must remain in the field of view of the fixed retina. Fig. 15(a) shows the accumulated retina output over 1.3 s without centering and all three objects rotate around the center of view. Fig. 15(b) shows the electronically centered retina output. Now the distractor objects rotate around the centered large circle, which was programmed into the convolution chip kernels.
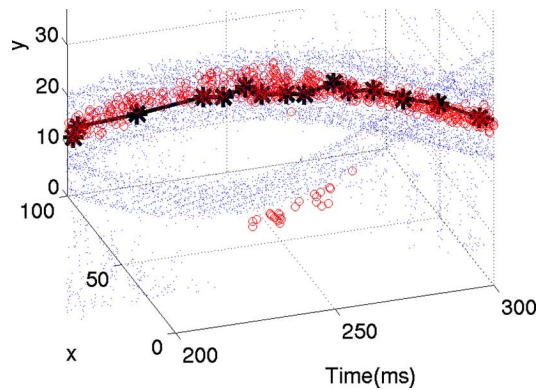
Fig. 16.   A 3-D x-y time reconstruction of events captured during a 100-ms interval with a rotating stimulus of two circles of different radii, rotating at four revolutions per second. Events are captured from the retina output (small dots), from the convolution PCB output (circles), and from the WTA output (big stars).
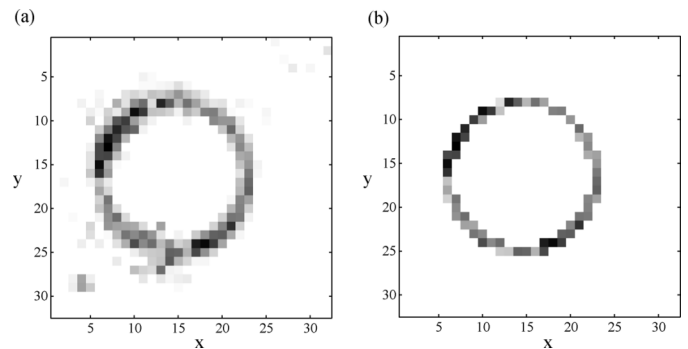

Fig. 17.   Spatial trajectory of the stimulus center. The stimulus is a disc that rotates with constant velocity in front of the retina. (a) The convolution stage contains a matched-filter kernel; its output is a smoothed version of the center of the object. The gray level gives the spike count for one revolution of the stimulus. (b) For the analysis, we consider only pixels that fall onto the trajectory of the stimulus. We masked these pixels with a manually defined region of interest.

A. The 3-D Reconstruction of Captured Events

For precise timing measurements, we need to visualize captured events as function of time. To illustrate this, we performed the same experiment as discussed in Figs. 13 and 14, but this time using a stimulus with two circles rotating at four revolutions per second and disconnecting the feedback servo mechanism to center one of the circles. Fig. 16 shows the events captured during a 100-ms time interval, from the retina, convolution chip, and object chip. This data shows how the retina output is filtered down by the convolution to a cloud mostly around the center of the matching input circle, and how the WTA events further reduce the data to just events centered on the programmed object.

B. Spike Output Statistics

From the recorded time-stamped spike information, we can perform various analysis, for example, measuring the exact statistical distribution of the outputs from the different chips. This measured distribution is useful when setting up the parameters of a chip to process its inputs. For example, since the task of WTA chip in the CAVIAR system during the visual tracking task is to localize the optimal location of the desired stimulus, the known statistical distribution of its inputs helps in the proper configuration of the parameters of the WTA chip [67]. The statistics of these spikes are important because the WTA is the first nonlinear decision making module in the CAVIAR chain. To determine the actual statistics of the convolution chip, we analyzed its output spike data in response to a disc rotating with constant velocity in front of the retina (Fig. 16). The convolution chip contains a matched-filter kernel of one size of a circle. The output spikes from the convolution chip in this figure indicate the location of the center of the detected kernel in the field of view. To simplify the subsequent analysis, we transformed the 2-D convolution chip output into 1-D by considering only neurons along the stimulus trajectory [Fig. 17(b)]. This transformation is possible since we know the trajectory in this simple problem. The transformation discards activity from neurons outside the trajectory of the stimulus center. These outliers receive less input than neurons on the trajectory and do not evoke output
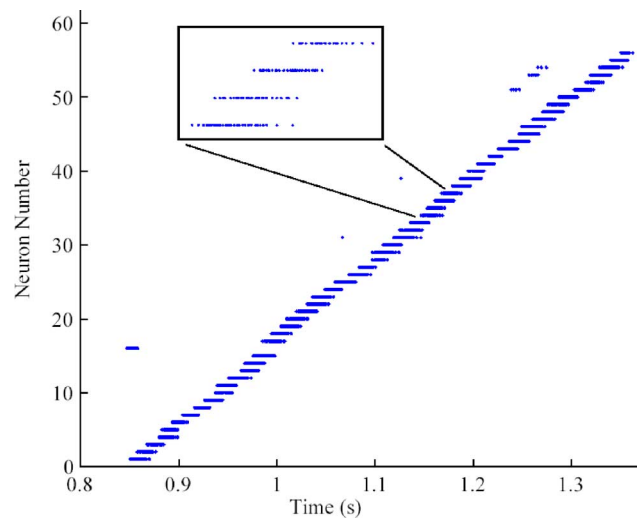

Fig. 18.   Raster plot of the input spike trains from the convolution chips to the WTA chip. Input addresses along the trajectory are sorted in the order of the stimulus movement. Each point marks one spike (every spike train contains about 20–40 spikes). Data from one revolution of the stimulus disc are shown.

spikes from the WTA. Our analysis focuses on the spatio–temporal estimation of the stimulus position, for which only the neurons with a significant spike input are relevant.

Fig. 18 shows the input spike trains to the WTA neurons along the stimulus trajectory before alignment, but sorted by their mean spike time. From this representation, the average travelling time d from one neuron to the next can be calculated, by averaging the difference in the mean time between each pair of neighboring input channels. By using a time-rescaling method to convert the nonstationary output of the convolution chip to a stationary one, subsequent analysis of the homogeneous distribution shows that the outputs are well modeled by a travelling Gaussian wave with Poisson statistics [67], [68], [90].

1) Reconstruction of Object Position From WTA: We consider how well the WTA network can reconstruct the stimulus position by requiring each output spike of the WTA to indicate the actual object position. The object position is discretized in value to the neuron address and in time to the occurrence of the output spikes as quantized by the recorded time stamp.
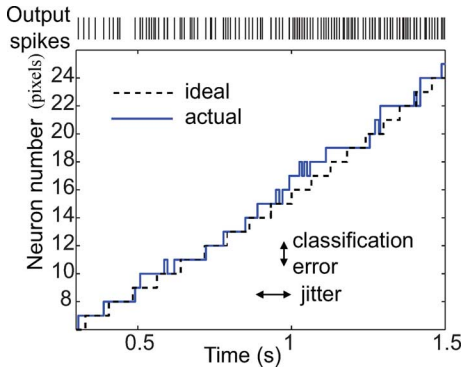
Fig. 19. Reconstruction of object position from the WTA output spikes. Horizontal axis is time (units in seconds) and vertical axis is neuron number (with units in pixels).
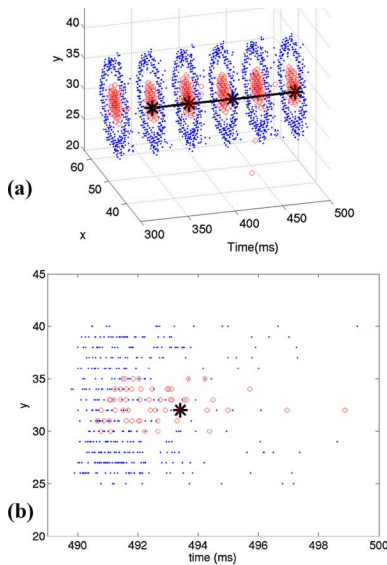

Fig. 20. Events obtained with a circle of flashing LEDs. (a) The 3-D x-y time reconstruction of events from the retina, convolution PCB output, and WTA output, captured during 200 ms. LEDs were flashing with an 80-ms period. (b) y time projection of one of the transients.

Fig. 19 shows how we can reconstruct the stimulus position from this asynchronous representation. The ideal output of the network is an update of the object position as soon as the object is aligned to a new neuron (dashed staircase). Since the object in our experiment moves with constant speed, the transition time from each neuron to the next is constant. In the sparsest representation, the network would elicit one spike at each vertical line of the dashed staircase function. In the CAVIAR data (continuous staircase), the WTA network elicits more spikes than one spike per position, as illustrated by the output spike train of the network at the top of Fig. 19. This leads to switching in the predicted object position, for example, between neurons 17 and 18. In addition, the output sometimes indicates an incorrect position, for example, at neurons 18 and 19, or the spike times are jittered, for example, at neurons 13 and 14.

We determine how much the object position reconstructed from the output spikes of the WTA network deviates from the ideal case in both position and time. We call an error in the object position a "classification error" if a neuron other than the
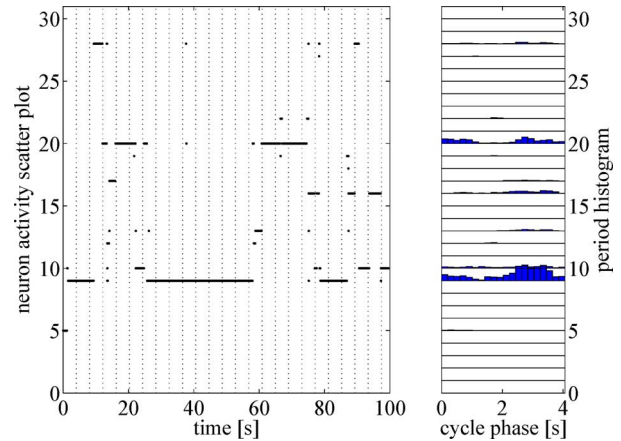

Fig. 21. Raster plot and cycle histogram of the activity of all 32 neurons after random initialization of the weights with learning turned off. Cycle times are indicated by the vertical dotted lines. Two neurons (9 and 20) are very dominant and mostly active for more than one cycle. This is a bad representation of the stimulus that will not allow to determine the stimulus' position within a cycle.

one aligned to the object position makes an output spike. This error is derived from a probability distribution. The spike time of the neuron before or after the ideal point in time is considered as jitter. This error is normalized by the average time T_n taken by the ball to proceed between one neuron to the next. For example, in Fig. 19, $T_n \approx 1.2$ s/17. Both errors are induced by the Poisson statistics of the input and the variations in its parameters. Since both jitter and classification error are interdependent, we use an error measure e, which quantifies the pixel error, and is computed as the area difference A_n between the reconstructed object location (dashed staircase) and the ideal case (continuous staircase), normalized to T_n times the number of neurons (pixels) N_n. The error from this experimental set obtained for one ball velocity is

$$e \approx \frac{A_n}{T_n N_n} \approx 0.64 \text{ pixels} \qquad (2)$$

which indicates an average error position of slightly better than a single pixel if one looks at the decision of the network at any one moment in time. This accuracy is achieved because the average WTA spike location, combined with its precision timing, overcomes the quantization and other noise sources.

C. Latency Measurements

To measure the chain processing latency, we stimulated the system with a circle of flashing light-emitting diodes (LEDs). The size of the flashing circumference is equal to one of the kernel sizes in the convolution chips. The LEDs are turned on and off with a period of 80 ms, and events are captured from the same three nodes as in Section VII-A. Fig. 20(a) shows the events captured from these three nodes during a time slice of 200 ms. We see event bursts every 40 ms, which corresponds to either a transient OFF–ON or ON–OFF of the diodes (here we are ignoring the sign bit of the retina). Fig. 20(b) shows the y-time projection of one such burst. We can see that the transient output of the retina lasts around 2 ms (the most dense part). The convolution output events start to appear between 0.5 and 1.0 ms after the initial wavefront of the retina (it needs enough events

to recognize the circle). The WTA stage fires an event with a delay of around 3 ms with respect to the onset of the convolution output. Not every convolution output burst produces a WTA output event [Fig. 20(a)]. In this configuration, the WTA was biased to produce a reduced number of events (without spurious or noisy events), so that in the experiment of Figs. 13 and 14, the mechanical stimulus centering section would receive a clean control signal with as little noise as possible.

D. Learning in the CAVIAR Demonstrator

In this section, we demonstrate the performance of the CAVIAR learning chips in achieving unsupervised classification of phases in a cyclic trajectory [73]. Thus, a compressed/quantized representation of that trajectory is learned, minimizing information loss.

The spatio–temporal learning classifier was connected to the WTA stage while the processing chain was configured to track the larger of two rotating circles. A sequence of the input to the learning system is depicted in Fig. 16 as a 3-D scatter plot. This input is projected (downsampled) to 2×2 pixels by combining all events from one quadrant into a single event stream. This is a low resolution and without the temporal aspect one could only hope to classify the WTA output into four categories, since there are now only four positions in space. However, these 2×2 spike trains are inserted into the delay line chip that is configured correspondingly as four separate delay lines (56 320 delay elements each). Those delay lines are tapped at three different delays (approximately 0 s, 200 ms, and 400 ms) and the resulting $2 \times 2 \times 3$ spike trains are passed on to the learning chip. There is now significantly more information that can be categorized by the learning chip: for example, different speeds and directions and in our case a higher spatial resolution, as the stimulus will cause several different trajectory patterns as it crosses between spatial pixels. The learning chip has been tuned to express rate-based learning behavior with an approximate time constant of 200 ms. The task of the learning classifier chip is now to provide a good representation of at most 32 categories (since there are 32 neurons) from the repeated spatio–temporal pattern.

After random initialization of the weights, the learning classifier performs poorly, as expected (Fig. 21). The cyclic motion (the vertical dotted lines represent the cycle frequency) cannot easily be deduced from the active neuron outputs: individual neurons remain exclusively active over several stimulus cycles and the activity is dominated by two to three neurons. These were often the same neurons in different experiments, since they tend to have generally stronger synapses due to transistor mismatch.

After learning for a few cycles, the situation is improved (Fig. 22). More neurons (approximately seven to nine) show significant activity and they are clearly phase locked with the stimulus. They have become specialized on a spatio–temporal input pattern that occurs during the cyclic object trajectory. They provide a richer encoding of the cyclic pattern, representing several different phases of the motion. Effects of device mismatch that put some neurons at an initial disadvantage are, thus, partially countered by learning, although still only 9 of 32 total neurons show significant activity with these parameter settings. Considering that there are still only four positions to distinguish (after
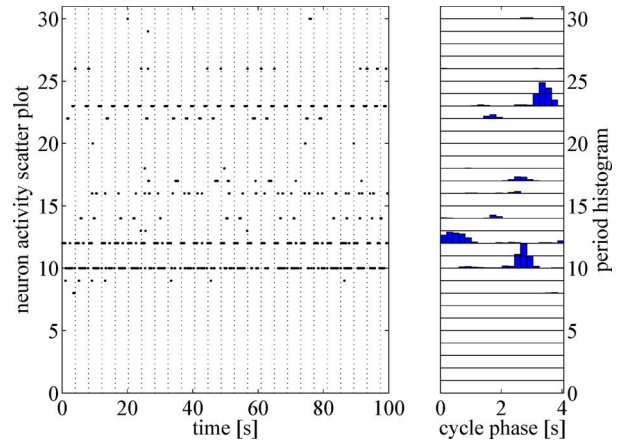

Fig. 22. Neuron activity after learning. The activities of several neurons are reliably phase locked at different phase delays and allow stimulus reconstruction with a better resolution than the cycle time.

the input space projection) and that the object only moves in one direction at constant speed, this is in fact a rather good performance. One could have expected that there are still only four different input patterns and that maximally four neurons could specialize on exactly those four patterns, but since this real-world input is changing its state in a continuous fashion rather than just assuming four discrete states, some of the neurons have become selective for transitory states "between" the four positions.

A more quantitative analysis [73] with three test sets of different well-controlled simulated inputs instead of real sensor data has shown that the chip implementation of the learning algorithm is able to improve information content of the network's output by 16%–20%, always close to the theoretical optimum for these particular test cases.

VIII. CONCLUSION AND FUTURE OUTLOOK

This paper demonstrates the high potential of modular AER hardware systems. To illustrate this, it demonstrates a multichip multilayer vision sensing processing and actuating architecture, configured for fast recognition and tracking of moving circles.

What is unique about the example system presented here is that it is composed of multistages of AER processing modules interconnected through multiple independent AER links, mimicking the multilayer structure of neural cortex. This approach allows to feasibly scale up systems by either adding more parallel modules in a layer, adding more layers, or even including recurrent feedback connections.

The CAVIAR system consists of about 45k spiking neurons and 5M synapses; and it can perform up to 12G connections/operations per second.[2] These metrics are computed as follows. For processing one 31×31 kernel the convolution chips need 330 ns per input event. Since there are four chips in parallel, it yields ($4 \times 31 \times 31/330$ n $= 12e9$) operations/s. The

---

[2]Here "connections/s" represents the traditional neural network (hardware and software) computational performance figure describing the number of synaptic connections computed per unit time. Note that this is not equal to the number of physical events communicated through an AER channel. Our convolution chips are very efficient in this sense, because for each input event, they can process up to $31 \times 31 = 961$ synaptic connections in 330 ns $= 2.9e9$ connections/s/chip.

synaptic strengths are 5-b values. For communicating analog values, one needs several events (assuming rate coding) for the same synapse between neurons A and B. Therefore, this figure could also be regarded as representing 12 giga events per second (Geps). Regarding the 5M synapses, what matters are the possible connections the system can be configured to implement. Of course, since this is implemented through convolutions, the projection fields have the restriction that they are position independent. The possible synaptic connections are as follows. From the retina to the convolution chip, there are up to $4 \times 32 \times 32 \times 31 \times 31 = 3.94$M possible connections and from the convolution chip to the WTA there are $32 \times 32 \times 32 \times 32 = 1.05$M connections, resulting in a total of approximately 5.2M synapses, but using 6.1k unique valued synapses. All these figures (number of neurons, synapses, and eps or connections/s) would scale up linearly by adding more modules to the system. This is because our up-scaling approach consists of adding AER modules as well as independent AER channels [91]. The ultimate bottleneck is the throughput of an individual AER channel. Splitting AER channels replicates their traffic while reaching more modules. On the other hand, merging AER channels increases throughput on the resulting channel. To avoid channel saturation, this merging should be done with some care (for example, instead of merging 100 channels and perform a convolution, one can merge them in groups of ten, do the same convolution with ten independent modules, and merge the ten convolution outputs).

The CAVIAR system in its present form has drawbacks. The major ones are the complexity of the hardware setup and the lack of rapidly configured flexibility. The present components share a common communication infrastructure and standardized cabling and logic protocols, but not a common configuration infrastructure. Each partner had developed their own tools and hardware and software interfaces for configuring chip biases, address mappings, and calibration, leading to the requirement for at least two experienced operators for the setup of the system and the use of typically two computers to impose the desired configuration over USB interfaces. The learning capabilities of CAVIAR are limited by the small numbers of plastic synapses. In addition, a major unsolved problem in computer science is the lack of general understanding of how to impose a desired behavior on interconnected networks of simple computing elements or how learning should be incorporated efficiently into the system. Thus, it is difficult to encode known mathematical signal processing methods or branch-like behavior which can readily be serially programmed but not configured into the system's connectivity.

Spinoff activities from this CAVIAR project are the ongoing development of fully digital convolution chips [86], application in real-time robotics [87] and commercialization [88] of the silicon retina, development of tools for behavioral simulation of future AER systems [89], and the active open-source software project jAER for real-time event-based procedural processing of AER sensor data [22], [90].

Still, this real-time hardware implementation of a neurophysiological model is far from attaining the same complexity as that of any brain: it consists of about six orders of magnitude fewer neurons and about eight orders fewer synapses than a human

brain. It requires slightly more space than a shoe box. Thus, if we would hypothetically scale it up to 10^{15} synapses, the 100 million shoe boxes would easily fill a few good sized warehouses.

Nevertheless, this AER system is a significant step towards the construction of more complex real-time, and real-world interactive artificial neural systems. We plan to miniaturize it by about 3–4 orders of magnitude within the next few years, by increasing the numbers of synapses and neurons per AER-module and by integrating more modules into a smaller physical volume. With present day miniature surface mount PCB technology and the latest deep submicron complementary metal–oxide–semiconductor (CMOS) technology, it is quite realistic to fit about 100 chips of 256×256 neurons each, with 32×32 synapses per neuron. One such PCB would host 6.5×10^6 neurons and 6.7×10^9 synapses. Assuming timing delays similar to those reported in this paper, preliminary results [91] suggest that these systems could perform sophisticated object recognition with delays around 100 μs. This would be equivalent to a computing power of 100 chips$\times 6.7 \times 10^9$ synapses/chip/100 μs $= 6.7 \times 10^{13}$ MAC/s (multiply and accumulates per second), which is 3–4 orders of magnitude above present state-of-the-art dedicated hardware for high-speed vision processing (see Appendix I).

With such developments, we will be able to provide a modular and scalable platform for real-time implementations of neural models of a really challenging complexity. Coupling this massive preprocessing power with flexible back-ends of conventional procedural computation will enable solutions to a host of practical applications.

APPENDIX I
COMPARISON TO CONVENTIONAL FRAME-BASED IMAGE PROCESSING APPROACHES

Other approaches to image processing are based on a frame-by-frame acquisition and consequent processing. Let us distinguish between the *sensing step* (acquiring an image frame) and the *processing steps* (processing the sequences of frames).

A. Sensing Step

Frames are acquired at a given frame rate $1/T$, and each pixel integrates light during a time period T_p. Usually both times are similar, although the latter has to be at least slightly smaller. For high-speed images, one can increase frame rate (which renders an explosion in the number of frames to process afterwards), or reduce T_p to avoid blurring of fast moving objects (which reduces either exposure times or image quality, or results in more expensive imagers). If one chooses to reduce T_p while keeping frame rate $1/T$ constant ($T_p \ll T$), then there will be information missing for a time $T - T_p$. This can be a severe limitation for recognition/tracking of moving objects. On the other hand, if one keeps $T_p \approx T$, either T is too small yielding an excessive number of frames for processing or T_p is too large, yielding blurred images. In any case, a compromise has to be reached, and, in either case, one always suffers the physical restriction of integrating light during a finite time period T_p, thus averaging light over this time.

For the sensors in this paper and related work, things are different. Each pixel has an internal light-dependent photocurrent which is continuous in time. In a luminance retina, it is directly coded into event frequency [38]. In a spatial [23] or temporal contrast [20] retina, its derivative is computed continuously and coded as pixel events. Consequently, as soon as a contrast is detected, events are sent out with very short delays: pixels do not wait for their frame rate refresh instant. This way, the precise timing of what is happening in the real world is preserved with AER frameless vision sensors. This allows to sense visual reality with timings equivalent to conventional cameras acquiring 10 000 frames/s or faster. The retinas developed in the context of the CAVIAR project are low power ($\sim$10 mW), small area ($\sim$5 mm^2), standard CMOS low-cost chips, as opposed to their high-end counterpart of imagers capable of operating at 10K frames/s.

B. Processing Steps

There are a number of solutions and products in the market for performing sophisticated visual object recognition tasks. All of them are based on sequential frame acquisition, and consequently suffer from the limitations mentioned above. Assuming images are acquired with satisfactory quality for later processing, let us now compare the CAVIAR approach to other conventional approaches, from the processing point of view. In the conventional approach, the usual solution is to perform the high-computation demanding operations (such as 2-D convolutions) on high-speed DSPs (for example, TMS320DM6446 DaVinci from Texas Instruments, special for video applications) or using computers with fast central processing units (CPUs) and special multimedia-oriented instructions. DSP TMS320DM6446 [92] can optimistically process up to 2.4×10^9 MAC/s (multiply accumulates per second). This will allow to perform a generic (no symmetries) 16×16 kernel convolution on a 256×256 image in 7 ms ($\sim$143 convolutions/s). Other more vision-oriented DSP chips, such as the DaVinci TMS320DM355 [93], are however designed for standard consumer video cameras operations like MPEG compression/decompression, histogramming, resizing, and autofocus. They do not implement 2-D convolutions directly, and are not really meant for recognition tasks.

Perhaps the most impressive chip available presently is the MathStar[3] field-programmable object array (FPOA): for example, it can process large kernel (16×16, although they must have symmetry constraints) 2-D convolutions on 256×256 images at a rate of $\sim$ 100 convolutions/frame at 25 frames/s ($\sim$2500 convolutions/s). This is equivalent to 3.8×10^{10} MAC/s. However (and ignoring the restriction on kernel symmetry), this approach is not scalable to multiple chips while maintaining speed.

C. Software Solutions

Commercial computers rely on video graphic cards for performing efficiently vision recognition convolution-based-like algorithms. Such cards ultimately use special purpose chips like those discussed above. At present, graphics processing unit

[3]www.mathstar.com

(GPU) chips are very popular among these cards. Cope *et al.* [95] perform a comparison of software using GPU versus CPU for 2-D convolutions. Performance changes with kernel size. The equivalent MAC per second can be obtained from [95] by multiplying the pixel throughput (number of pixels processed per second) by the number of MAC per pixel (which varies from 4 for a 2×2 kernel convolution to 121 for an 11×11 one): a state-of-the-art GPU yields 4×10^9 MAC/s for 2×2 kernels and 7.3×10^9 MAC/s for 11×11 kernels, while a Pentium 4 at 3 GHz ranges from 6×10^7 MAC/s for 2×2 kernels to 1.2×10^8 MAC/s for 11×11 kernels.

APPENDIX II
COMPETITIVE HEBBIAN LEARNING FOR CLASSIFICATION

Weight vector normalizing competitive Hebbian learning has been used to achieve principal component analysis (PCA) [76]. One particular implementation of this is Sanger's rule [77]. It differs somewhat from the implementation presented here: since Sanger's rule implements the inhibitory term into the learning rule, whereas here the neurons' activities inhibit each other and thus, indirectly the learning term. Furthermore Sanger's rule achieves ordered PCA because it only introduces partial cross inhibition, where "higher order neurons" inhibit "lower order neurons" but not *vice versa*. Here full cross inhibition, which has also been suggested, for example, in [78], would lead to unordered extraction of principle components.

The experiments here, however, are set up such that not all conditions for computing a PCA are met. The input distribution is not of zero mean, weights and signals can only be positive, and the cross-inhibition is so strong as to induce "hard" WTA, where all neurons but the winner are suppressed completely and the number of neurons is bigger than the dimensionality of the input space. So how can the resulting behavior be described?

Integrate-and-fire neurons described in terms of signal rates behave like linear threshold elements [73]

$$y = \begin{cases} \vec{w}^T \vec{x} - \dfrac{1}{\tau} = \|\vec{w}\|\|\vec{x}\|(\cos\alpha) - \dfrac{1}{\tau}, & \text{if } \vec{w}^T\vec{x} > \dfrac{1}{\tau} \\ 0, & \text{if } \vec{w}^T\vec{x} \le \dfrac{1}{\tau} \end{cases}$$
$$(3)$$

where $\vec{w}$ is the neuron's weight vector, $\vec{x}$ is the input vector, y is the output, τ is the time constant of the leakage, and α is the angular distance between x and w. This shows that with the weight vector normalized to the same length for all neurons, the neuron with the smallest α will always win the WTA competition and thus the input space is carved up into different classes along hyperplanes of equal angular distance between the weight vectors.

Hebbian learning will reduce the angular distance between the input vector $\vec{x}$ and $\vec{w}$. Thus, the winning neuron will adapt its $\vec{w}$ closer to $\vec{x}$. With no competitors and the inputs limited to the first hyperoctant (only positive input vector elements) a neuron will tend to move towards the direction of the most dense distribution of long input vectors. When competing with others, it will move towards dense regions of inputs that are not yet claimed by another neuron. This behavior is much akin the well-known learning vector quantization (LVQ) algorithm [94], with the difference that the weight vectors move according to their angular distance to the inputs, not their Euclidian distance, and that the
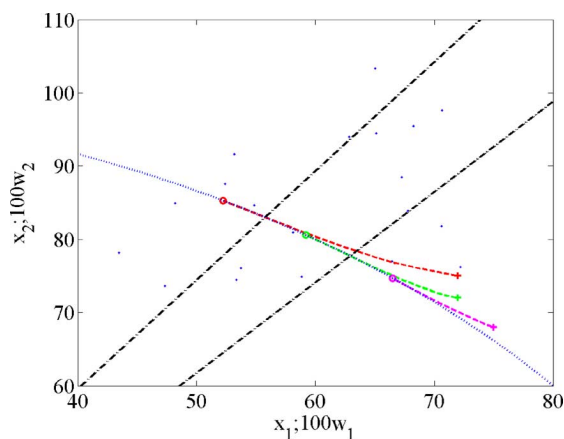
Fig. 23. Example for learning classification with weight vector normalizing competitive Hebbian learning.

input space is separated along hyperplanes of equal angular distance between weight vectors, and not equal Euclidian distance.

An illustration is given in Fig. 23 with a 2-D example. Weights $\vec{w}$ (multiplied by 100) and inputs $\vec{x}$ are plotted in the same coordinate system. Twenty different input patterns/vectors are randomly distributed in a limited area (marked with dots, x_1-coordinates from 40 to 80 and x_2-coordinates from 70 to 110). During learning, these 20 input vectors are presented repeatedly to the neurons in random order. The initial weight values (marked with "+") lie to the right corner of this input cluster. With these initial weight values only one neuron (the one with the weight vector that is closest to the input cluster) will win the WTA competition, a quite poor classification of the inputs. Learning is performed according to a rate-based description of (1) that has been derived in [73]. It is equivalent to (1) under the assumption that all signals are independent Poisson spike trains

$$\frac{d\vec{w}}{dt} = \mu \left(\frac{y\tau}{1+y\tau} x - \frac{1}{\nu^2} y\vec{w} \right). \tag{4}$$

μ is the learning rate (set to 0.0001) and ν is another parameter that sets the length for the weight vector normalization (set to 1). Learning moves all three weight vectors towards the cluster of inputs (dashed lines) and normalizes the weight vectors (the dotted line is the unity circle around the origin multiplied with 100). After 100 iterations, they reach the normalized states indicated by the small circles. The final classification areas are indicated by the dashed–dotted separation lines (that go through the coordinate system's origin). Now the classification/quantization of the inputs is much more evenly distributed among the three classes. From an information preservation point of view, this is a much improved classification. Quantization always leads to information loss but if the classes occur with equal probability, the information content of the quantized signal is maximized.

To summarize, the weight vector normalizing competitive Hebbian learning employed on the CAVIAR learning chip attempts to assign all input patterns it experiences evenly to its WTA neurons to obtain a classification/quantization of the inputs trying to preserve as much information as possible. In the general case, one can expect an improvement of the classification, but an optimal outcome is not guaranteed. It depends on the initial condition of the weights and the particulars of the input distribution, as well as on the nonidealities of the physical chip implementation, such as mismatch and quantized weight storage.

REFERENCES

[1] T. Serre, "Learning a dictionary of shape-components in visual cortex: Comparison with neurons, humans and machines," MIT. Comput. Sci. & AI Lab, Cambridge, MA, Tech. Rep. MIT-CSAIL-TR-2006-028 CBCL-260, 2006.

[2] S. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system," *Nature*, vol. 381, pp. 520–522, Jun. 1996.

[3] S. Lawrence, C. L. Giles, A. Tsoi, and A. Back, "Face recognition: A convolutional neural network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.

[4] Y. Le Cun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. Cambridge, MA: MIT Press, 1995, pp. 255–258.

[5] M. Matsugu, K. Mori, M. Ishi, and Y. Mitarai, "Convolutional spiking neural network model for robust face detection," in *Proc. 9th Int. Conf. Neural Inf. Process.*, 2002, vol. 2, pp. 660–664.

[6] B. Fasel, "Robust face analysis using convolutional neural networks," in *Proc. Int. Conf. Pattern Recognit.*, 2002, pp. 40–43.

[7] M. Browne and S. S. Ghidary, "Convolutional neural networks for image processing: An application in robot vision," in *Advances in Artificial Intelligence*. Cambridge, MA: MIT Press, 2003, pp. 641–652.

[8] C. Neubauer, "Evaluation of convolution neural networks for visual recognition," *IEEE Trans. Neural Netw.*, vol. 9, no. 4, pp. 685–696, Jul. 1998.

[9] K. Fukushima, "Analysis of the process of visual pattern recognition by the neocognitron," *Neural Netw.*, vol. 2, pp. 413–420, 1989.

[10] L. Itti, "Quantitative modeling of perceptual salience at human eye position," *Vis. Cogn.*, vol. 14, no. 4, pp. 959–984, 2006.

[11] L. Itti, "Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes," *Vis. Cogn.*, vol. 12, no. 6, pp. 1093–1123, Aug. 2005.

[12] H. Fujii, H. Ito, K. Aihara, N. Ichinose, and M. Tsukada, "Dynamical cell assembly hypothesis – Theoretical possibility of spatio-temporal coding in the cortex," *Neural Netw.*, vol. 9, pp. 1303–1350, 1996.

[13] G. M. Shepherd, *The Synaptic Organization of the Brain*, 3rd ed. Oxford, U.K.: Oxford Univ. Press, 1990.

[14] T. Crimmins, "Geometric filter for speckle reduction," *Appl. Opt.*, vol. 24, pp. 1438–1443, 1985.

[15] S. Grossberg, E. Mingolla, and J. Williamson, "Synthetic aperture radar processing by a multiple scale neural system for boundary and surface representation," *Neural Netw.*, vol. 8, no. 7/8, pp. 1005–1028, 1995.

[16] S. Grossberg, E. Mingolla, and W. D. Ross, "Visual brain and visual perception: How does the cortex do perceptual grouping?," *Trends Neurosci.*, vol. 20, pp. 106–111, 1997.

[17] E. Mingolla, W. Ross, and S. Grossberg, "A neural network for enhancing boundaries and surfaces in synthetic aperture radar images," *Neural Netw.*, vol. 12, no. 3, pp. 499–511, 1999.

[18] R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jimenez, and B. Linares-Barranco, "A neuromorphic cortical-layer microchip for spike- based event processing vision systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 12, pp. 2548–2566, Dec. 2006.

[19] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 × 128 120 dB 30 mW asynchronous vision sensor that responds to relative intensity change," in *Dig. Tech. Papers IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, 2006, pp. 508–509.

[20] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 × 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2007.

[21] T. Delbruck and P. Lichtsteiner, "Fully programmable bias current generator with 24 bit resolution per bias," in *Proc. IEEE Int. Symp. Circuits Syst.*, Athens, Greece, 2006, pp. 2849–2852.

[22] T. Delbrück, jAER Open Source Project, [Online]. Available: http://jaer.wiki.sourceforge.net

[23] J. Costas-Santos, T. Serrano-Gotarredona, R. Serrano-Gotarredona, and B. Linares-Barranco, "A contrast retina with on-chip calibration for neuromorphic spike-based AER vision systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 7, pp. 1444–1458, Jul. 2007.

[24] M. Sivilotti, "Wiring considerations in analog VLSI systems with application to field-programmable networks," Ph.D. dissertation, Comput. Sci. Div., California Inst. Technol., Pasadena, CA, 1991.

[25] M. Mahowald, "VLSI analogs of neural visual processing: A synthesis of form and function," Ph.D. dissertation, Comput. Sci. Div., California Inst. Technol., Pasadena, CA, 1992.

[26] M. Mahowald, *An Analog VLSI Stereoscopic Vision System*. Norwell, MA: Kluwer, 1994.

[27] J. Lazzaro, J. Wawrzynek, M. Mahowald, M. Silvilotti, and D. Gillespie, "Silicon auditory processors as computer peripherals," *IEEE Trans. Neural Netw.*, vol. 4, no. 3, pp. 523–528, May 1993.

[28] J. P. Lazzaro and J. Wawrzynek, "A multi-sender asynchronous extension to the address-event protocol," in *Proc. 16th Conf. Adv. Res. VLSI*, W. J. Dally, J. W. Poulton, and A. T. Ishii, Eds., 1995, pp. 158–169.

[29] A. Mortara and E. A. Vittoz, "A communication architecture tailored for analog VLSI artificial neural networks: Intrinsic performance and limitations," *IEEE Trans. Neural Netw.*, vol. 5, no. 3, pp. 459–466, May 1994.

[30] A. Mortara, E. A. Vittoz, and P. Venier, "A communication scheme for analog VLSI perceptive systems," *IEEE J. Solid-State Circuits*, vol. 30, no. 6, pp. 660–669, Jun. 1995.

[31] Z. Kalayjian and A. G. Andreou, "Asynchronous communication of 2D motion information using winner-takes-all arbitration," *Int. J. Anal. Integr. Circuits Signal Process.*, vol. 13, no. 1-2, pp. 103–109, Mar./Apr. 1997.

[32] K. A. Boahen, "Communicating neuronal ensembles between Neuromorphic Chips," in *Neuromorphic Systems Engineering: Neural Networks in Silicon*, T. S. Lande, Ed. Norwell, MA: Kluwer, 1998, ch. 11.

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

[34] K. A. Boahen, "A burst-mode word-serial address-event link-I: Transmitter design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 7, pp. 1269–1280, Jul. 2004.

[35] K. A. Boahen, "A burst-mode word-serial address-event Link-II: Receiver design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 7, pp. 1281–1291, Jul. 2004.

[36] K. A. Boahen, "A burst-mode word-serial address-event Link-III: Analysis and test results," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 7, pp. 1292–1300, Jul. 2004.

[37] K. Boahen, ChipGen Silicon Compiler, [Online]. Available: http://www.stanford.edu/group/brainsinsilicon/Downloads.htm

[38] E. Culurciello, R. Etienne-Cummings, and K. A. Boahen, "A biomorphic digital image sensor," *IEEE J. Solid-State Circuits*, vol. 38, no. 2, pp. 281–294, Feb. 2003.

[39] U. Mallik, M. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings, "Temporal change threshold detection imager," in *Dig. Tech. Papers IEEE ISSCC*, San Francisco, CA, 2005, pp. 362–363.

[40] P. F. Ruedi, P. Heim, F. Kaess, E. Grenet, F. Heitger, P. Y. Burgi, S. Gyver, and P. Nussbaum, "A 128 × 128, pixel 120-dB dynamic-range vision-sensor chip for image contrast and orientation extraction," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2325–2333, Dec. 2003.

[41] C. Shoushun and A. Bermak, "A low power CMOS imager based on time-to-first-spike encoding and fair AER," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, pp. 5306–5309.

[42] X. G. Qi and J. Harris, "A time-to-first-spike CMOS imager," in *Proc. IEEE Int. Symp. Circuits Syst.*, Vancouver, BC, Canada, 2004, pp. 824–827.

[43] M. Azadmehr, J. Abrahamsen, and P. Häfliger, "A foveated AER imager chip," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, May 2005, vol. 3, pp. 2751–2754.

[44] R. J. Vogelstein, U. Mallik, E. Culurciello, R. Etienne-Cummings, and G. Cauwenberghs, "Spatial acuity modulation of an address-event imager," in *Proc. 11th IEEE Int. Conf. Electron. Circuits Syst.*, Dec. 2004, pp. 207–210.

[45] K. A. Zaghloul and K. Boahen, "Optic nerve signals in a neuromorphic chip: Part I and II," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 4, pp. 657–675, Apr. 2004.

[46] M. Arias-Estrada, D. Poussart, and M. Tremblay, "Motion vision sensor architecture with asynchronous self-signaling pixels," in *Proc. 7th Int. Workshop Comput. Architecture Mach. Perception*, 1997, pp. 75–83.

[47] C. M. Higgins and S. A. Shams, "A biologically inspired modular VLSI system for visual measurement of self-motion," *IEEE Sensors J.*, vol. 2, no. 6, pp. 508–528, Dec. 2002.

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

[49] G. Indiveri, A. M. Whatley, and J. Kramer, "A reconfigurable neuromorphic VLSI multi-chip system applied to visual motion computation," in *Proc. Int. Conf. Microelectron. Neural Fuzzy Bio-Inspired Syst.*, Granada, Spain, 1999, pp. 37–44.

[50] K. Boahen, "Retinomorphic chips that see quadruple images," in *Proc. Int. Conf. Microelectron. Neural Fuzzy Bio-Inspired Syst.*, Granada, Spain, 1999, pp. 12–20.

[51] R. Z. Shi and T. K. Horiuchi, "A VLSI model of the bat dorsal nucleus of the Lateral Lemniscus for azimuthal echolocation," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, 2005, pp. 4217–4220.

[52] V. Chan, S.-C. Liu, and A. van Schaik, "AER EAR: A matched silicon cochlea pair with address event representation interface," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 1, pp. 48–59, Jan. 2007.

[53] G. Cauwenberghs, N. Kumar, W. Himmelbauer, and A. G. Andreou, "An analog VLSI chip with asynchronous interface for auditory feature extraction," *IEEE Trans. Circuits Syst. II, Anal. Digit. Signal Process.*, vol. 45, no. 5, pp. 600–606, May 1998.

[54] M. Oster and S.-C. Liu, "Spiking inputs to a spiking winner-take-all circuit," in *Advances in Neural Information Processing Systems*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, 2006, vol. 18, pp. 1051–1058.

[55] J. Abrahamsen, P. Häfliger, and T. S. Lande, "A time domain winner-take-all network of integrate-and-fire neurons," in *Proc. IEEE Int. Symp. Circuits Syst.*, Vancouver, BC, Canada, May 2004, vol. V, pp. 361–364.

[56] E. Chicca, G. Indiveri, and R. J. Douglas, "An event-based VLSI network of integrate-and-fire neurons," in *Proc. IEEE Int. Symp. Circuits Syst.*, Vancouver, BC, Canada, 2004, vol. V, pp. 357–360.

[57] T. Teixeira, A. G. Andreou, and E. Culurciello, "Event-based imaging with active illumination in sensor networks," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, 2005, pp. 644–647.

[58] P. Vernier, A. Mortara, X. Arreguit, and E. A. Vittoz, "An integrated cortical layer for orientation enhancement," *IEEE J. Solid-State Circuits*, vol. 32, no. 2, pp. 177–186, Feb. 1997.

[59] T. Y. W. Choi, B. E. Shi, and K. Boahen, "An ON-OFF orientation selective address event representation image transceiver chip," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 2, pp. 342–353, Feb. 2004.

[60] T. Serrano-Gotarredona, A. G. Andreou, and B. Linares-Barranco, "AER image filtering architecture for vision processing systems," *IEEE Trans. Circuits Syst. II, Anal. Digit. Signal Process.*, vol. 46, no. 9, pp. 1064–1071, Sep. 1999.

[61] A. Jiménez-Fernández, C. D. Luján, A. Linares-Barranco, F. G.-. Rodríguez, M. Rivas, G. Jiménez, and A. Civit, "Address-event based platform for bio-inspired spiking systems," *Proc. SPIE*, vol. 6592, 2007, DOI: 10.1117/12.724156, 659206.

[62] R. Paz, A. Linares-Barranco, M. Rivas, L. Miró, S. Vicente, G. Jiménez, and A. Civit, "AER tools for communications and debugging," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 3253–3256.

[63] E. Chicca, A. M. Whatley, V. Dante, P. Lichtsteiner, T. Delbrück, P. D. Giudice, R. J. Douglas, and G. Indiveri, "A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 5, pp. 981–993, May 2007.

[64] R. Berner, T. Delbruck, A. Civit-Balcells, and A. Linares-Barranco, "A 5 Meps $100 USB2.0 address-event monitor-sequencer interface," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2007, pp. 2451–2454.

[65] C. Kimme, D. H. Ballard, and J. Slansky, "Finding circles by an array of accumulators," *Commun. Assoc. Comput. Mach.*, vol. 18, pp. 120–122, 1975.

[66] M. Oster, Y. Wang, R. Douglas, and S.-C. Liu, "Quantification of a spike-based winner-take-all VLSI network," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 10, pp. 3160–3169, Nov. 2008.

[67] M. Oster and S.-C. Liu, "Quantifying input and output statistics of a winner-take-all network in a vision system," in *Proc. IEEE Int. Symp. Circuits Syst.*, New Orleans, LA, May 2007, pp. 853–856.

[68] S.-C. Liu and M. Oster, "Feature competition in a spike-based winner-take-all VLSI network," in *Proc. IEEE Int. Symp. Circuits Syst.*, Island of Kos, Greece, May 2006, pp. 3634–3637.

[69] M. Oster and S. Liu, "Spiking inputs to a winner-take-all network," in *Advances in Neural Information Processing Systems (NIPS)*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA: MIT Press, 2005, vol. 18, pp. 1051–1058.

[70] M. Oster, R. Douglas, and S.-C. Liu, "Computation with spikes in a winner-take-all network," in *Neural Computation*. Cambridge, MA: MIT Press, to be published.

[71] M. Boegerhausen, P. Suter, and S.-C. Liu, "Modeling short-term synaptic depression in silicon," *Neural Comput.*, vol. 15, pp. 331–348, 2003.

[72] C. Mead and T. Delbrück, "Scanners for visualizing activity of analog VLSI circuitry," *Int. J. Anal. VLSI Signal Process.*, vol. 1, pp. 93–106, 1991.

[73] P. Häfliger, "Adaptive WTA with an analog VLSI neuromorphic learning chip," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 551–572, Mar. 2007.

[74] R. Kempter, W. Gerstner, and J. L. van Hemmen, "Spike-based compared to rate-based Hebbian learning," *Neural Inf. Process. Syst.*, vol. 11, pp. 125–131, 1999.

[75] P. Häfliger and H. K. Riis, "A multi-level static memory cell," in *Proc. IEEE Int. Symp. Circuits Syst.*, Bangkok, Thailand, May 2003, vol. 1, pp. 22–25.

[76] E. Oja, "A simplified neuron model as principal component analyzer," *J. Math. Biol.*, vol. 15, pp. 267–273, 1982.

[77] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Netw.*, vol. 2, pp. 459–473, 1989.

[78] L. A. Akers, D. K. Ferry, and R. O. Grondin, "Synthetic neural systems in the 1990s," in *An Introduction to Neural and Electronic Networks*, S. F. Zornetzer, J. L. Davis, C. Lau, and McKenna, Eds. New York: Academic, 1995, pp. 359–387.

[79] E. Brown, R. Barbieri, V. Ventura, R. Kass, and L. Frank, "The time-rescaling theorem and its application to neural spike train data analysis," *Neural Comput.*, vol. 14, no. 2, pp. 325–346, 2002.

[80] A. Linares-Barranco, G. Jimenez-Moreno, B. Linares-Barranco, and A. Civit-Ballcels, "On algorithmic rate-coded AER generation," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 771–788, May 2006.

[81] A. Delorme, J. Gautrais, and S. Thorpe, "SpikeNET: A simulator for modeling large networks of integrate-and-fire neurons," *Neurocomputing*, vol. 26–27, pp. 989–96, 1999.

[82] A. Delorme, L. Perrinet, and S. Thorpe, "Networks of integrate-and-fire neurons using rank order Coding B: Spike timing dependent plasticity and emergence of orientation selectivity," *Neurocomputing*, vol. 38–40, pp. 539–45, 2001.

[83] D. George and F. T. Sommer, "Computing with inter-spike interval codes in networks of integrate and fire neurons," *Neurocomputing*, vol. 65–66, pp. 415–420, 2005.

[84] F. Gomez-Rodriguez, R. Paz, L. Miró, A. Linares-Barranco, G. Jiménez, and A. Civit, "Two hardware implementations of the exhaustive synthetic AER generation method," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2005, vol. 3512, pp. 534–540.

[85] A. Linares-Barranco, M. Oster, D. Cascado, G. Jiménez, A. Civit, and B. Linares-Barranco, "Inter-spikes-intervals analysis of AER Poisson-like generator hardware," *Neurocomputing*, vol. 70, no. 16-18, pp. 2692–2700, 2007.

[86] L. Camuñas-Mesa, A. Acosta-Jiménez, T. Serrano-Gotarredona, and B. Linares-Barranco, "Fully digital AER convolution chip for vision processing," in *Proc. IEEE Int. Symp. Circuits Syst.*, Seattle, WA, May 2008, pp. 652–655.

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

[88] H. Garn and M. Krenn, Smart Eye TDS-Traffic Data Sensor, [Online]. Available: http://www.smart-systems.at/products/ products_smart_optical_sensors_t_en.html

[89] J. A. Perez-Carrasco, T. Serrano-Gotarredona, C. Serrano-Gotarredona, B. Acha, and B. Linares-Barranco, "High-speed character recognition system based on a complex hierarchical AER architecture," in *Proc. IEEE Int. Symp. Circuits Syst.*, Seattle, WA, May 2008, pp. 2150–2153.

[90] T. Delbruck, "Frame-free dynamic digital vision," in *Proc. Int. Symp. Secure-Life Electron. Adv. Electron. Quality Life Soc.*, Tokyo, Japan, 2008, pp. 21–26.

[91] R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jimenez, C. Serrano-Gotarredona, J. A. Perez-Carrasco, A. Linares-Barranco, G. Jimenez-Moreno, A. Civit-Ballcels, and B. Linares-Barranco, "On real-time AER 2D convolutions hardware for neuromorphic spike based cortical processing," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1196–1219, Jul. 2008.

[92] TMS320DM6446 Digital Media System-on-Chip (DMSoC) Datasheet, Texas Instruments,, Houston, TX, SPRS283E–December 2005–Revised March 2007.

[93] TMS320DM355 Digital Media System-on-Chip (DMSoC) Datasheet, Texas Instruments, Houston, TX, SPRS463–September 2007.

[94] T. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. New York: Springer-Verlag, 1989.

[95] B. Cope, P. Y. K. Cheung, W. Luk, and S. Witt, "Have GPUs made FPGAs redundant in the field of Video Processing?," in *Proc. Int. Conf. Field Programmable Technol.*, 2005, pp. 111–118.

Rafael Serrano-Gotarredona received the B.S. degree in telecommunications engineering and the Ph.D. degree [after completing all his research at the Seville Microelectronics Institute (IMSE), which is one of the institutes of the National Microelectronics Center (CNM) of the "Spanish Research Council (CSIC) of Spain] from the University of Seville, Seville, Spain, in 2002 and 2007, respectively. During his doctoral studies, he held a scholarship from the Spanish Ministry of Education and Science.

During June–July 2006, he was with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, as a visiting scholar. He also visited the Institute of Neuroinformatics (INI), ETH Zürich/University of Zürich, Zürich, Switzerland, under the European funded research project CAVIAR. From February 2007 to October 2008, he was Analog Design Engineer at the Design Center, NXP, Graz, Austria. Currently, he is a Staff Design Engineer at Austriamicrosystems, Valencia, Spain. His research interests include analog and mixed-signal very large scale integration (VLSI) circuit design applied to vision processing systems, high-speed LVDS chip communications, and inductive coupling circuits for automotive applications.

Matthias Oster studied electrical engineering and information technologies at the Technical University of Munich (TUM), Munich, Germany, from 1997 to 2002 and received the Ph.D. degree from the Institute of Neuroinformatics (INI), ETH Zürich/University of Zürich, Zürich, Switzerland, in 2006, under the supervision of S.-C. Liu and R. Douglas.

During his doctoral studies, he explored spike-based winner-take-all computation integrated in the European Union funded project CAVIAR building a biologically inspired high-speed image-processing system. In a follow-up postdoctoral project that lasted until March 2007, he explored the embedding of syntactic constraints in hardware neuronal networks in the scope of the DAISY project. His research interests include biologically inspired computation algorithms and their implementation in hardware very large scale integration (VLSI) systems, asynchronous communications and information theory, and the field of vision processing. Since May 2007, he has been with Varian Medical Systems, Baden, Switzerland.

Dr. Oster was a member of the German National Academic Foundation and the "Deutsche Schülerakademie" and received travel grants from the International Conference on Artificial Neural Networks in 2005, the Institute of Neuromorphic Engineering, Maryland, in 2004, and the University of Seville, Seville, Spain, in 2005.

Patrick Lichtsteiner received the Diploma (equivalent to M.S.) degree in physics and the Ph.D. degree from the Swiss Federal Institute of Technology, Zürich, Switzerland, in 2002 and 2006, respectively.

He recently joined the chip design group of Espros Photonics Corporation, Baar, Switzerland. His research interests include complementary metal–oxide–semiconductor (CMOS) imaging, neuromorphic vision sensors, and high-speed vision.

Dr. Lichtsteiner and his colleagues have received four prizes for IEEE conference papers, including the 2006 International Solid-State Circuits Conference (ISSCC) Jan Van Vessem Outstanding Paper Award.

Alejandro Linares-Barranco received the B.S. degree in computer engineering, the M.S. degree in industrial computer science, and the Ph.D. degree in computer science (specializing in computer interfaces for bioinspired systems) from the University of Seville, Seville, Spain, in 1998, 2002, and 2003, respectively.

From January 1998 to June 1998, he was Second Lieutenant in the Spanish Airforce working as System Administrator and Software Developer. During 1998, he also worked at the Colors Digital Communications S. L. Company, Seville, Spain. From November 1998 to February 2000, he was a Member of the Technical Staff at the Seville Microelectronics Institute (IMSE), an institute of the National Microelectronics Center (CNM), Spanish Research Council (CSIC), Seville, Spain. From March 2000 to February 2001, he was a Development Engineer at the Research and Development Department, SAINCO company, Seville, Spain, working on VHDL-based field-programmable gate array (FPGA) systems for the INSONET European project on power line communications. In September 2001, he became an Assistant Professor of Computer Architecture and Technology at the University of Seville. In October 2006, he was promoted to Associate Professor. His research interests include very large scale integration (VLSI) and FPGA digital design, vision processing systems, bus emulation, computer architectures, motor control, and robotics.

Rafael Paz-Vicente was born in Seville, Spain, in 1976. He received the B.S. degree in computer engineering, the M.S. degree in industrial computer science, and the Ph.D. degree in computer science (specializing in bioinspired systems for image processing and digital interfaces) from the University of Seville, Seville, Spain, in 2001, 2003, and 2009, respectively.

Since September 2001, he has been an Assistant Professor of Computer Architecture and Technology at the University of Seville. His research interests include very large scale integration (VLSI) and field-programmable gate array (FPGA) digital design, vision processing systems, bus emulation, computer architectures, motor control, and robotics.

Francisco Gómez-Rodríguez received the B.S. degree in computer engineering from the University of Seville, Seville, Spain, in 1999. His Ph.D. work is devoted to exploring the artificial neuroinspired vision systems possibilities.

From November 1999 to December 2002, he was Member of the Artificial Vision, Automatic Control and Robotics Research Group at the University of Seville, where participated in European Union funded projects devoted to automatic fire detection, monitoring, and modeling (INFLAME and SPREAD projects). Since January 2003, he has been Member of the Robotics and Computer Technologies Research Group at the University of Seville. He is also an Assistant Professor at the Computer Architecture and Technology Department, University of Seville. In 2005, he was an Invited Researcher at the Institute of Neuroinformatics (INI), ETH Zürich/University of Zürich, Zürich, Switzerland. His research interests are artificial vision systems and bioinspired systems.

Luis Camuñas-Mesa was born in Córdoba, Spain, in 1979. He received an Engineer degree in telecommunications from the University of Seville, Seville, Spain, in 2003 and the M.Sc. degree in microelectronics from the Institute of Microelectronics of Seville (IMSE-CNM), Seville, Spain, in 2005, where he is currently working towards the Ph.D. degree.

His research interests include analog design of floating-gate-based circuits and very large scale integration (VLSI) implementations of real-time vision processing systems.

Raphael Berner received the M.Sc. degree in electrical engineering and information technology from the Swiss Federal Institute of Technology, Zürich, Switzerland, in 2006. He conducted his M.S. thesis at the Department of Computer Architecture and Technology, University of Seville, Seville, Spain, on high speed USB2 AER computer interfaces. Currently, he is working towards the Ph.D. degree at the Institute of Neuroinformatics (INI), ETH Zürich/University of Zürich, Zürich, Switzerland, and Swiss Federal Institute of Technology, Zürich, Switzerland.

His research interests include complementary metal–oxide–semiconductor (CMOS) neuromorphic vision sensors and event-based processing.

Manuel Rivas-Pérez received the B.S. degree in computer engineering from the University of Seville, Seville, Spain, in 2004, where he is currently working towards the Ph.D. conducting research in cellular-automata-based artificial vision systems.

Since November 2004, he worked on research projects related to AER-based systems. Since October 2005, he has also been an Assistant Professor at the Computer Architecture and Technology Department, University of Seville. His research interests include bioinspired vision systems, field-programmable gate array (FPGA) designs, and computer architectures.

Tobi Delbrück studied physics and applied mathematics as an undergraduate and received the Ph.D. in computation and neural systems with carver mead from California Institute of Technology (Caltech), Pasadena, in 1993.

Currently, he is a titular Professor of Physics and Group Leader at the Institute of Neuroinformatics (INI), ETH Zürich/University of Zürich, Zürich, Switzerland and Visiting Scientist at Caltech. His recent interests are in spike-based silicon retina and vision sensor design, event-based digital vision, and micropower visual and auditory sensors. He coinvented the standard neuromorphic adaptive photoreceptor circuit and bump circuit. Subsequently, he worked for several years on electronic imaging at Arithmos, Synaptics, National Semiconductor, and Foveon, where he was one of the founding employees. In 1998, he moved to Switzerland to join INI. In 2002, he was Lead Developer of the tactile luminous floor used in INI's exhibit "Ada: Playful Intelligent Space." He has been awarded eight patents, and has over 30 refereed papers in journals and conferences, four book chapters, and one book.

Dr. Delbrück has been awarded six IEEE awards, including the 2006 ISSCC Jan Van Vessem Outstanding European Paper Award. He coorganized the demonstration sessions at the Neural Information Processing Systems (NIPS) meeting in 2004–2005 and the ISCAS demonstration session in 2007–2008 and is an officer of the IEEE CAS Sensory Systems Technical Committee and Associate Editor of the IEEE TRANSACTIONS OF BIOMEDICAL CIRCUITS AND SYSTEMS.

Shih-Chii Liu received the B.S. degree in electrical engineering from Massachusetts Institute of Technology, Cambridge, in 1983, the M.S. degree in electrical engineering from University of California, Los Angeles, in 1988, and the Ph.D. degree from the Computation and Neural Systems Program, California Institute of Technology (Caltech), Pasadena, in 1997.

Currently, she is an Assistant Professor at the Institute of Neuroinformatics (INI), ETH Zürich/University of Zürich, Zürich, Switzerland. She was with

Gould American Microsystems from 1983 to 1985, and with LSI Logic from 1985 to 1988. She worked at Rockwell International Research Labs from 1988 to 1997. Her research interests include neuromorphic neuronal modeling of vision and cortical processing, networks for behavior generation, and hybrid analog/digital signal processing.

Dr. Liu has been Chair of the IEEE CAS Sensory Systems Technical Committee from May 2007 until May 2009.

Rodney Douglas graduated in science and medicine and received the Ph.D. degree in neuroscience from the University of Cape Town, Rondebosch, South Africa, in 1985.

He joined the Anatomical Neuropharmacology Unit in Oxford, U.K., where he continued his research on the anatomy and biophysics of neocortical circuits together with K. Martin. Currently, he is Professor of Neuroinformatics, and Co-Director at the Institute of Neuroinformatics (INI), ETH Zürich/University of Zürich, Zürich, Switzerland. As a Visiting Associate, and then Visiting Professor at California Institute of Technology (Caltech), Pasadena, he extended his research interests in neuronal computation to the modeling of cortical circuits using digital methods (together with C. Koch), and also by the fabrication of analog very large scale integration (VLSI) circuits (together with M. Mahowald). In 1996 he and K. Martin moved to Zürich, Switzerland, to establish the Institute of Neuroinformatics.

Dr. Douglas was awarded the Körber Foundation prize for European Science in 2000.

Philipp Häfliger received the M.Sc. degree in computer science with astronomy as a second subject from the Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, in 1995 and the Ph.D. degree from the Institute of Neuroinformatics (INI), ETH Zürich/University of Zürich, Zurich, Switzerland, in 2000.

At INI, where biologists, psychologists, neurologists, engineers, mathematicians, physicists and computer scientists join forces to explore the working principles of the brain, he started in earnest to design bioinspired analog and hybrid integrated circuits. Many of these circuits operate on subthreshold currents (i.e., currents that traditional digital electronics considers entirely negligible) trying to approach the energy efficacy of biological organisms. His special focus was on on-chip learning algorithms. He moved for a postdoctoral position to the Institute of Informatics, University of Oslo, Oslo, Norway, where he now is an Associate Professor. Between 2002 and 2006, he has participated in the European Union funded project CAVIAR and since fall 2006 he has worked on the GlucoSense project sponsored by the Norwegian research council, developing ultralow power electronics for a pill sized wireless implant.

Dr. Häfliger is Secretary of the IEEE Circuits and Systems (CAS) Society's Biomedical CAS Technical Committee, Track Chair for the IEEE International Symposium on Circuits and Systems live demonstrations track 2009, and Technical Program Committee Member of the IEEE Conference on Biomedical CAS (2007 and 2008). He serves as reviewer for a number of journals (recently the IEEE TRANSACTIONS ON NEURAL NETWORKS and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: REGULAR PAPERS) and conferences (recently the IEEE International Symposium on Circuits and Systems and Neural Information Processing Systems).
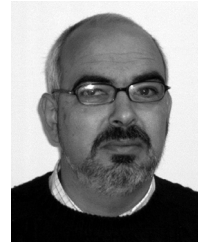
Gabriel Jiménez-Moreno received the M.S. degree in physics (electronics) and the Ph.D. degree from the University of Seville, Seville, Spain, in 1990 and 1992, respectively.

After working with Alcatel, he was granted a fellowship from the Spanish Science and Technology Commission (CICYT). Currently, he is Professor Titular of Computer Architecture at the University of Seville. From 1993 until 1995, he was Assistant Editor of the Journal of Computer and Software Engineering, Ablex Publishing Corporation. From 1996 until 1998, he was Vice-Dean of the E.T.S. Ingenieria Informatica, University of Seville. He participated in the creation of the Department of Computer Architecture, University of Seville, and since 2004, he has been its Secretary. He is author of various papers and research reports on robotics, rehabilitation technology, and computer architecture. He has directed two national research projects on neuromorphic systems. His research interests include neuronal networks, vision processing systems, embedded systems, computer interfaces, and computer architectures.

Dr. Jiménez-Moreno has been the Counselor of the IEEE Student Branch at the University of Seville since 2003.

Antón Civit Ballcels received the M.S. degree in physics (electronics) and the Ph.D. degree in hierarchical multiprocessor design from the University of Seville, Seville, Spain, in 1984 and 1987, respectively.

He worked for several months with Hewlett-Packard. In the late 1980s, he participated in the creation of two SMEs related to eLearning and environment monitoring networks. Since 1990, he has been Professor Titular of Computer Architecture at the University of Seville. He participated in the creation of the Department of Computer Architecture, University of Seville, where he is currently the Director. Initially, he worked in research projects related to multiprocessor multiple robot control architectures. He published several papers and directed four Ph.D. theses on these topics. As Director of the "Robotics and Computer Technology" research group he has lead projects related to advanced wheelchair navigation and intelligent environment support for wheelchair users. These topics have also produced publications and two Ph.D. dissertations. Since 1992, he has also worked on web and computer accessibility issues. He participates in several EU and national level research projects in the areas of neuromorphic systems, accessible telecoms and ambient intelligence.

Dr. Civit is a member of the European Commission eAccessibility expert group. He has been a member of the COST219bis research action management committee and, currently, he participates in the COST 219ter (accessibility to next generation networks) management committee. He has participated in several European Commission evaluation activities in the Telematics Applications and IST programs.

Teresa Serrano-Gotarredona (M'07) received the B.S. degree in electronic physics from the University of Seville, Seville, Spain, in June 1992. She received the Ph.D. degree in very large scale integration (VLSI) neural categorizers from the University of Seville, in December 1996, after completing all her research at the Seville Microelectronics Institute (IMSE), which is one of the institutes of the National Microelectronics Center (CNM) of the Spanish Research Council (CSIC) of Spain. She received the M.S. degree from the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD, in 1997, where she was sponsored by a Fulbright Fellowship.

She was on a sabbatical stay at the Electrical Engineering Department, Texas A&M University, College Station, during Spring 2002. She was Assistant Professor at the University of Seville from 1998 until 2000. Since June 2000, she has been a Tenured Scientist at the Seville Microelectronics Institute (IMSE), Seville, Spain, and in July 2008, she was promoted to Tenured Researcher. She is coauthor of the book *Adaptive Resonance Theory Microchips* (Norwell, MA: Kluwer, 1998). Her research interests include analog circuit design of linear and nonlinear circuits, VLSI neural-based pattern recognition systems, VLSI implementations of neural computing and sensory systems, transistor parameters mismatch characterization, address–event-representation VLSI, radio-frequency (RF) circuit design, and real-time vision processing chips.

Dr. Serrano-Gotarredona was corecipient of the 1997 IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS Best Paper Award for the paper "A real-time clustering microchip neural engine" and of the IEEE CAS Darlington Award for the paper "A general translinear principle for subthreshold MOS transistors." She is an officer of the IEEE CAS Sensory Systems Technical Committee.

Antonio J. Acosta-Jiménez received the "Licenciado en Física" degree and the Ph.D. degree in electronic physics from the University of Seville, Seville, Spain, in 1989 and 1995, respectively.

Currently, he is with the Seville Microelectronics Institute, Seville, Spain, and also with the Department of Electronics and Electromagnetism, University of Seville, where he has been an Associate Professor since 1998. His current research interests are in the areas of complementary metal–oxide–semiconductor (CMOS) digital and mixed-signal very large scale integration (VLSI) design, low-power and low-noise CMOS, description of timing phenomena in VLSI digital system, and asynchronous and self-timed circuits. He authored or coauthored more than 70 international scientific publications and has been involved in different national and European R&D projects.

Dr. Acosta-Jiménez was General Chair of the 2002 PATMOS International Workshop.

Bernabé Linares-Barranco received the B. S. degree in electronic physics, the M. S. degree in microelectronics, and the Ph.D. degree in high-frequency OTA-C oscillator design from the University of Seville, Seville, Spain, in 1986, 1987, and 1990, respectively, and the Ph.D. degree in analog neural network design from Texas A&M University, College Station, in 1991.

Since September 1991, he has been a Tenured Scientist at the Seville Microelectronics Institute (IMSE), which is one of the institutes of the National Microelectronics Center (CNM) of the Spanish Research Council (CSIC) of Spain. In January 2003, he was promoted to Tenured Researcher and in January 2004, to full Professor of Research. Since March 2004, he has been also a part-time Professor at the University of Seville. From September 1996 to August 1997, he was on sabbatical stay at the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD, as a Postdoctoral Fellow. During Spring 2002, he was Visiting Associate Professor at the Electrical Engineering Department, Texas A&M University. He is coauthor of the book *Adaptive Resonance Theory Microchips* (Norwell, MA: Kluwer, 1998). He was the coordinator of the European Union funded CAVIAR project. He has been involved with circuit design for telecommunication circuits, very large scale integration (VLSI) emulators of biological neurons, VLSI neural-based pattern recognition systems, hearing aids, precision circuit design for instrumentation equipment, bioinspired VLSI vision processing systems, transistor parameters mismatch characterization, address–event-representation VLSI, radio-frequency (RF) circuit design, and real-time vision processing chips.

Dr. Linares-Barranco was corecipient of the 1997 IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS Best Paper Award for the paper "A real-time clustering microchip neural engine," and of the 2000 IEEE CAS Darlington Award for the paper "A general translinear principle for subthreshold MOS transistors." He organized the 1994 NIPS Postconference Workshop "Neural Hardware Engineering." From July 1997 until July 1999, he was Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING, and since January 1998, he has also been the Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS. He was Chief Guest Editor of the 2003 IEEE TRANSACTIONS ON NEURAL NETWORKS Special Issue on Neural Hardware Implementations. From June 2009 until May 2011, he is Chair of the Sensory Systems Technical Committee of the IEEE CAS Society.