Fast Vision Through Frameless Event-Based Sensing and Convolutional Processing: Application to Texture Recognition

José Antonio Pérez-Carrasco, Begoña Acha, Carmen Serrano, Luis Camuñas-Mesa, Teresa Serrano-Gotarredona, *Member, IEEE*, and Bernabé Linares-Barranco, *Fellow, IEEE*

***Abstract*—Address–event representation (AER) is an emergent hardware technology which shows a high potential for providing in the near future a solid technological substrate for emulating brain-like processing structures. When used for vision, AER sensors and processors are not restricted to capturing and processing still image frames, as in commercial frame-based video technology, but sense and process visual information in a pixel-level event-based frameless manner. As a result, vision processing is practically simultaneous to vision sensing, since there is no need to wait for sensing full frames. Also, only meaningful information is sensed, communicated, and processed. Of special interest for brain-like vision processing are some already reported AER convolutional chips, which have revealed a very high computational throughput as well as the possibility of assembling large convolutional neural networks in a modular fashion. It is expected that in a near future we may witness the appearance of large scale convolutional neural networks with hundreds or thousands of individual modules. In the meantime, some research is needed to investigate how to assemble and configure such large scale convolutional networks for specific applications. In this paper, we analyze AER spiking convolutional neural networks for texture recognition hardware applications. Based on the performance figures of already available individual AER convolution chips, we emulate large scale networks using a custom made event-based behavioral simulator. We have developed a new event-based processing architecture that emulates with AER hardware Manjunath's frame-based feature recognition software algorithm, and have analyzed its performance using our behavioral simulator. Recognition rate performance is not degraded. However, regarding speed, we show that recognition can be achieved before an equivalent frame is fully sensed and transmitted.**

***Index Terms*—Address–event representation (AER), AER chips, convolutional neural networks, event coding and processing, real-time vision hardware processing, spike signal processing, texture retrieval, vision chips.**

Manuscript received November 14, 2008; accepted December 20, 2009. Date of publication February 22, 2010; date of current version April 02, 2010. This work was supported in part by the Spanish Ministry of Education and Science under Grant TEC-2006-11730-C03-01 (SAMANTA2), be the Andalusian regional government under Grant P06-TIC-01417 (Brain System), and by the European Union (EU) Grants IST-2001-34124 (CAVIAR) and 216777 (NABAB). The work of J. A. Pérez-Carrasco was supported by a doctoral scholarship as part of research project Brain System.

J. A. Pérez-Carrasco is with the Dpto. Teoría de la Señal, ETSIT, Universidad de Sevilla, Sevilla, Spain, and also with the Instituto de Microelectrónica de Sevilla (IMSE-CNM-CSIC), Sevilla 41092, Spain.

B. Acha and C. Serrano are with the Departamento de Teoría de la Señal, ETSIT, Universidad de Sevilla, Sevilla, Spain.

L. Camuñas-Mesa, T. Serrano-Gotarredona, and B. Linares-Barranco are with the Instituto de Microelectrónica de Sevilla (IMSE-CNM-CSIC), Sevilla 41092, Spain (e-mail: bernabe@imse-cnm.csic.es).

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.2009.2039943

I. Introduction

ARTIFICIAL man-made machine vision systems operate in a quite different way from biological brains. Machine vision systems usually capture and process sequences of frames. For example, a video camera captures images at about 25–30 frames per second, which are then processed frame by frame, pixel by pixel, usually with convolution operations, to extract, enhance, and combine features, and perform operations in feature spaces, until a desired recognition is achieved. This frame convolution processing is slow, especially if many convolutions need to be computed in sequence for each input image or frame.

Biological brains seem to not operate on a frame by frame basis. In the retina, each pixel sends spikes (also called events) to the cortex when its activity level reaches a threshold. Pixels are not read by an external scanner. Pixels decide when to send an event. All these spikes are transmitted as they are being produced, and do not wait for an artificial "frame time" before sending them to the next processing layer.[1] Besides this frameless nature, brains are structured hierarchically in cortical layers [1]. Neurons (pixels) in one layer connect to a *projection field* of neurons (pixels) in the next layer. This processing based on projection fields is similar to convolution-based processing [2], at least for the earlier cortical layers. For example, it is widely accepted that the first layer of visual cortex V1 performs an operation similar to a bank of 2-D Gabor-like filters at different scales and orientations [3] whose actual parameters have been measured [4]–[6]. This fact has been exploited by many researchers to propose powerful convolution-based image processing algorithms [3], [7]–[12]. However, convolutions are computationally expensive. It seems unlikely that the high number of convolutions that might be performed by the brain could be emulated fast enough by software programs running on the fastest of today's computers. Many researchers believe that a new hardware technology is required for approaching the processing capability of biological brains.

Address–event representation (AER) is a promising emergent hardware technology that shows potential for providing the computing requirements of large frameless projection-field-based multilayer systems. AER was first proposed in 1991 in one of the California Institute of Technology (Caltech) research

[1]Strictly speaking, this argument is still under debate as some researchers suggest there exists a saccadic induced refresh. In any case, it is widely accepted that retina pixels are not scanned sequentially.

labs [13], and has been used since then by a wide community of neuromorphic hardware engineers. AER has been used fundamentally in image sensors, for simple light intensity to frequency transformations [15], time-to-first-spike coding [16], [17], foveated sensors [18], contrast [19], [20], more elaborate transient detectors [21], and motion sensing and computation systems [22]. But AER has also been used for auditory systems [14], [23], competition and winner-takes-all networks [24], [25], and even for systems distributed over wireless networks [26]. However, the high potential of AER has become even more apparent since the availability of AER convolution chips [27], [28]. These chips, which can perform large arbitrary kernel convolutions (32×32 in [27]) at speeds of about 3×10^9 connections/s/chip, can be used as building blocks for larger cortical-like multilayer hierarchical structures, because of the modular and scalable nature of AER-based systems. Currently, only a small number of such chips have been used simultaneously[2] [29], but it is expected that hundreds of such modular AER convolution units could be integrated in a compact volume, such as a miniature printed circuit board (PCB) or into chips of the type known as networks-on-chip (NoC) [30]. This would eventually allow the assembly of large cortical-like convolutional neural networks and event-based frameless vision processing systems operating at very high speeds.

II. Frame-Based Versus Event-Based Sensing and Processing

Fig. 1 illustrates the conceptual difference between a frame- and an event-based sensing and processing system. Each use a camera sensor to capture reality. In the top row, a frame-based camera captures a sequence of frames, each of which is transmitted to the computing system. Each frame is processed by sophisticated image processing algorithms for achieving some recognition. The computing system needs to have all pixel values of a frame before starting any computation. In the bottom row, an event-based vision sensor operates without frames. Each pixel sends an event (usually its own x, y coordinate) when it senses something (change in intensity [21], contrast with respect to neighboring pixels [20], etc.). Events are sent out to the computing system as they are produced, without waiting for a frame time. The computing system updates its state after each event. Fig. 2 illustrates the inherent difference in timings between both concepts. In the top (frame-based), reality is binned into compartments of duration T_{frame}. During the first frame T_1, an event happens (such as a flashing shape), but the information produced by this event does not reach the computing system until the full frame is captured (at T_1) and transmitted (with an additional delay Δ). Then, the computing system has to process the full frame, handling large amount of data and requiring a long frame computation time T_{FC} before the "recognition" information is available. In the bottom of Fig. 2, pixels "see" directly the event in reality and send out their own events with a delay Δ to the computing system. Events are processed as they flow with an event computation delay T_{ev} (some nanoseconds [27]). For performing recognition not all events are necessary. Actually, more relevant events usually come out first or with higher frequency. Consequently, recognition time T_{rcg} can be smaller than the total time of the events produced. Note that recognition is possible before frame time T_1, resulting in a negative T'_{FC} when compared to the recognition delay of a frame-based system.

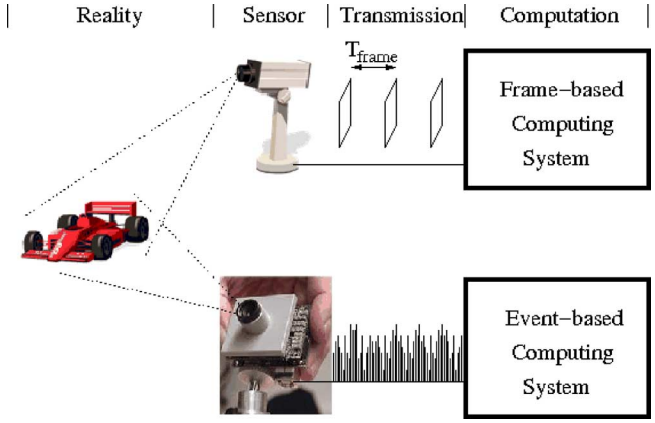

Fig. 1. Conceptual illustration of frame-based (top) versus event-based (bottom) vision sensing and processing system.

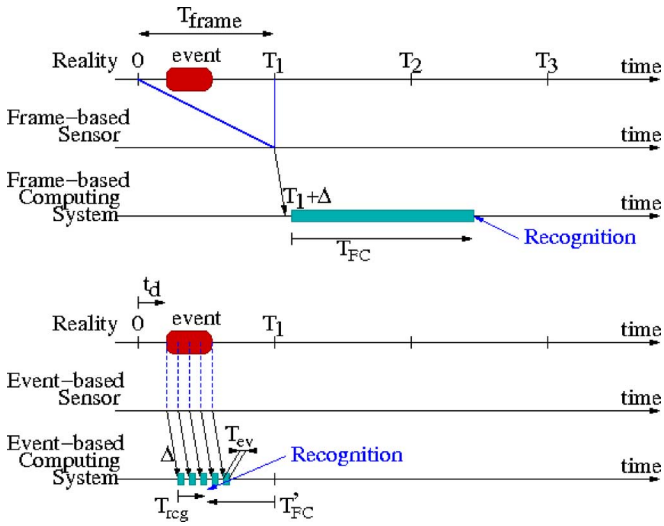

Fig. 2. Comparison of timing issues between (top) a frame- and (bottom) an event-based sensing and processing system.

Fig. 3 provides an illustration of a typical operation of an AER-based hardware [31]. In this case, the hardware is composed of one temporal contrast (motion) sensing retina of 128×128 pixels [21] that is sending its output events to a 2-D convolution chip programmed with a 7×7 pixel vertical Gabor filter. A pixel in the retina sends out an event (which usually consists of its x, y coordinate) every time its incident light intensity changes a relative amount of at least 2.5%. Fig. 3(a) shows the 1500 events generated by the retina during about 80 ms when observing two persons walking. The receiver convolution chip processes each event as it comes in with a delay of about $T_{\text{ev}} = 90$ ns [27]. Pixels in the 2-D array of integrators of the convolution chip will generate their own output events. Fig. 3(b) shows the 300 output events produced by the convolution chip during the same 80 ms. This 7×7

[2]This is because currently only noncommercial experimental prototyping chips are available, which are provided in reduced number of samples by microchip foundries.

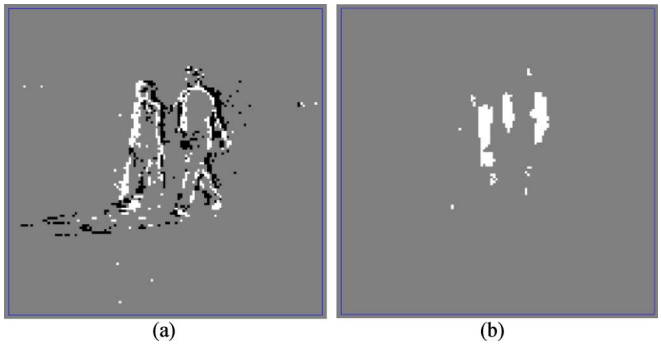

Fig. 3. Illustration about the hardware implementation of the method. (a) Two persons walking captured with a 128 × 128 temporal contrast (motion) retina [21]. Pixels sensing a positive time derivative in light intensity send a positive event (white), while those sensing a negative time derivative send a negative event (black). Gray pixels are silent. The figure shows the events captured during an interval of about 80 ms with a total of about 1500 events. (b) As these pixel events are generated asynchronously by the motion retina, they are received and processed one by one by a receiver convolution chip programmed with a 7×7 vertical Gabor 2-D spatial filter. The computation delay in the convolution chip is 90 ns per event [27]. The figures shows about 300 output events produced during the same 80 ms by the convolution chip.

kernel typically requires between 5 and 20 spatio–temporal correlated input events to produce an output event. As soon as these events are fed to the convolution chip, the corresponding output event appears with a delay of 90 ns. Consequently, in practice, input and output event flows are simultaneous.

Interestingly, AER hardware sensing or processing modules can be assembled into large hierarchical structures, as if one assembles bricks [29]. This is because of the robustness and asynchrony of the AER communication links between the modules, and the availability of "glue" modules such as AER splitters, mergers, and mappers [29], [32].

While the AER hardware technology takes its time to mature for allowing the availability of such large scale modular systems, the AER research community also needs to provide a more theoretical substrate for knowing how to assemble, configure, program, and train such systems. What is the optimum hierarchical structure for a desired application? What kernels are best? Can they be learned through a training process? What other parameters should be set? In this paper, our goal is to perform a step towards this more theoretical direction. We will concentrate on one potential application for AER convolution-based visual processing: texture recognition. Based on performance results of individual AER convolution chips already tested, our goal is to emulate through behavioral simulations, a relatively large multimodule AER convolutional neural network for texture recognition, and estimate its eventual performance, especially in terms of speed response. We will use an AER behavioral simulator developed in Visual C++ [33], which allows to behaviorally describe any AER module (including timing and nonideal characteristics), and assemble large netlists of many different modules. This allows obtaining a realistic estimate of the processing delays of the simulated systems. As we will see, recognition retrieval performance is similar to state-of-the-art frame-based algorithms, while recognition time delays are such that the results are available before the equivalent frame sensing and transmission time.

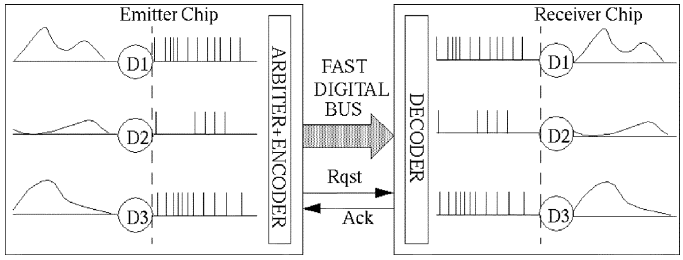

Fig. 4. Concept of point-to-point interchip AER communication.

III. AER for Convolution Processing

Fig. 4 illustrates event communication in a point-to-point AER link [36], where pixel intensity is coded directly as pixel event frequency.[3] The continuous-time states of pixels D_i in an emitter chip are transformed into sequences of fast digital pulses (spikes or events) of minimal width (in the order of nanoseconds) but with much longer inter-spike intervals (typically in the order of milliseconds). Each time a pixel generates a spike, its x, y address is written on the interchip digital bus, after proper arbitration [13]. This is called an "address event." The receiver chip reads and decodes the addresses of the incoming events and sends spikes to the corresponding receiving pixels for reconstruction or further processing. This point-to-point communication in Fig. 4 can be extended to a multireceiver scheme [14]. Also, multiple emitters can merge their outputs into a smaller set of receiver chips [29]. Moreover, AER visual information can easily be translated or rotated by remapping the addresses during interchip transmission [37], [38]. Complex processing such as convolutions has also been demonstrated [27]–[29].

To illustrate how AER convolution is performed event by event (without frames) consider the example in Fig. 5. Fig. 5(a) corresponds to a conventional frame-based convolution, where a 5×5 input image $f(i,j)$ is convolved with a 3×3 kernel $h(m,n)$, producing a 5×5 output image $g(i,j)$. Mathematically, this corresponds to sweeping kernel $h()$ over the full pixel array $f()$

$$g(i,j) = \sum_m \sum_n h(m,n) f(i-m, j-n). \qquad (1)$$

In an AER system, shown in Fig. 5(b), an intensity retina sensing the same visual stimulus would produce events for some pixels only (those sensing a nonzero light intensity). Every time an event from the retina chip is received by the convolution chip, the kernel is added to the array of pixels (which operate as adders and accumulators) around the pixel having the same event coordinate. Note that this is actually a *projection-field* operation. This way, after the four retina events have been received and processed, the result accumulated in the array of pixels in Fig. 5(b) is equal to that in Fig. 5(a). In a more realistic situation, the retina pixel values are higher and more events are sent per pixel. However, note that more intense pixels have higher frequencies, and consequently, their events will start to come out earlier, and will

[3]This is known as rate coding and is used in AER luminance retinas [15] and spatial contrast retinas [20]. Coding the time derivative results in temporal contrast (motion) retinas [21]. Other coding schemes have also been proposed [16], [17], [34].

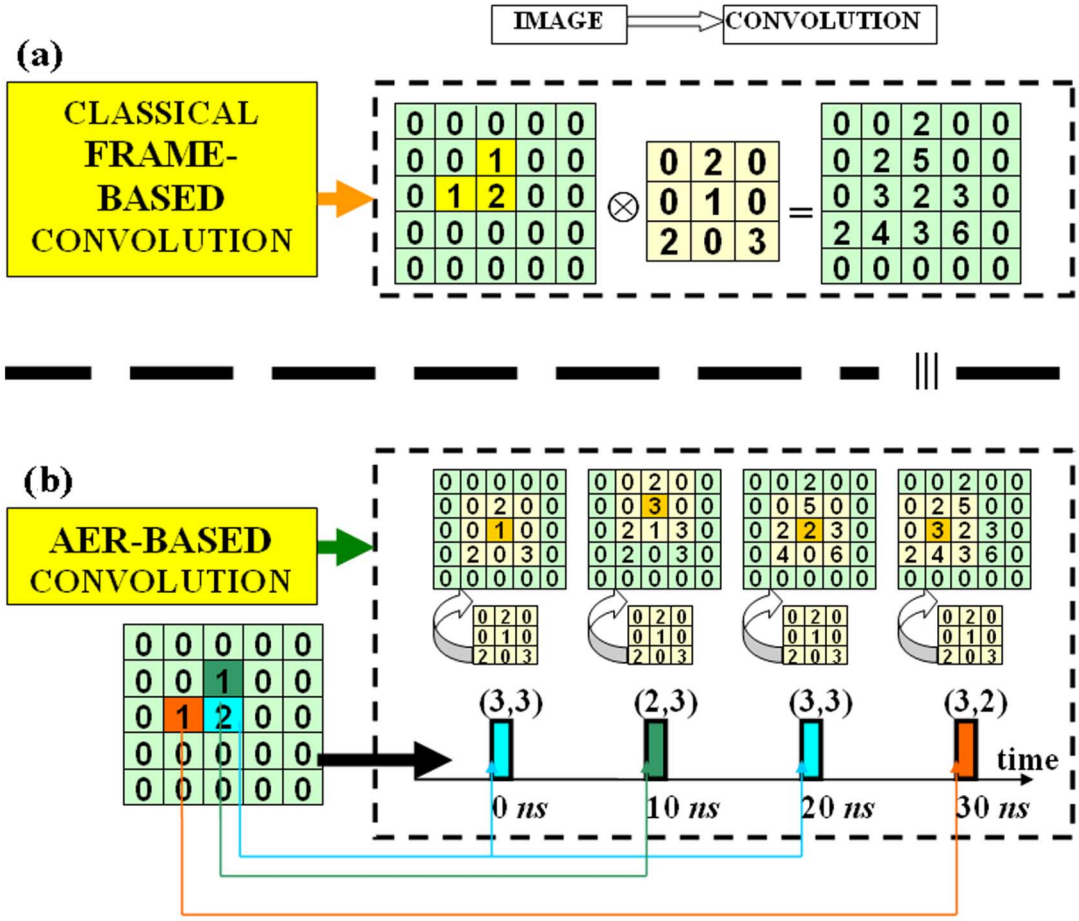

Fig. 5. Comparison between (a) classical frame-based and (b) AER event-based convolution processing.

be processed first. The first "wave front" of events is therefore more relevant for object recognition. AER visual sensors become significantly more efficient if they include on-chip some extra preprocessing, such as temporal [21] or spatial [20] contrast. In this case, only pixels with a minimum contrast level generate events. These pixels are the most meaningful for object/texture recognition. Using such sensors also increases dramatically the efficiency of the posterior cortical processing, as the number of events is reduced at least one order of magnitude while keeping the meaningful information content. In AER systems, since events are processed by a multilayer cortical-like structure as they are produced by the sensor, it is possible to achieve successful recognition after a fraction of the total number of events are processed [39].

IV. Texture-Based AER Retrieval

We have developed an AER system for computing Manjunath's Gabor wavelet features for texture analysis [35]. By performing texture analysis using Gabor filters (2-D convolutions) at different scales and orientations, these patterns can be efficiently described in the frequency domain and localized in the spatial domain. Texture is analyzed by applying a bank of scale and orientation Gabor filters to an image. Next we summarize the sequence of computations performed in Manjunath's method [35], and indicate how we have adapted them for an AER hardware system.

A. Manjunath's Frame-Based Method

A 2-D Gabor function $g(x,y)$ can be written as

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+j2\pi Wx\right] \quad (2)$$

where σ_x, σ_y, and W are its geometric parameters. Let $g(x,y)$ be the mother wavelet. A Gabor filter bank can be obtained by appropriate dilations and rotations of $g(x,y)$ through the generating function

$$g_{s,k}(x,y) = a^{-s}g(x',y')\begin{cases} x' = a^{-s}(x\cos\theta + y\sin\theta) \\ y' = a^{-s}(-x\sin\theta + y\cos\theta) \end{cases} \quad (3)$$

where θ represents orientation and s the scale. The filter bank parameters $\{\sigma_x, \sigma_y, a, \theta, W\}$ are computed by Manjunath's method [35]. Given an image $I(x,y)$, its Gabor wavelet transform is then defined as

$$W_{mn}(x,y) = \int I(x_1,y_1)g^*_{mn}(x-x_1, y-y_1)dx_1dy_1. \quad (4)$$

The mean μ_{mn} and the standard deviation σ_{mn} of the magnitude of the transform coefficients

$$\mu_{mn} = \iint |W_{mn}(xy)|dxdy$$
$$\sigma_{mn} = \sqrt{\iint (|W_{mn}(x,y)| - \mu_{mn})^2 dxdy} \quad (5)$$

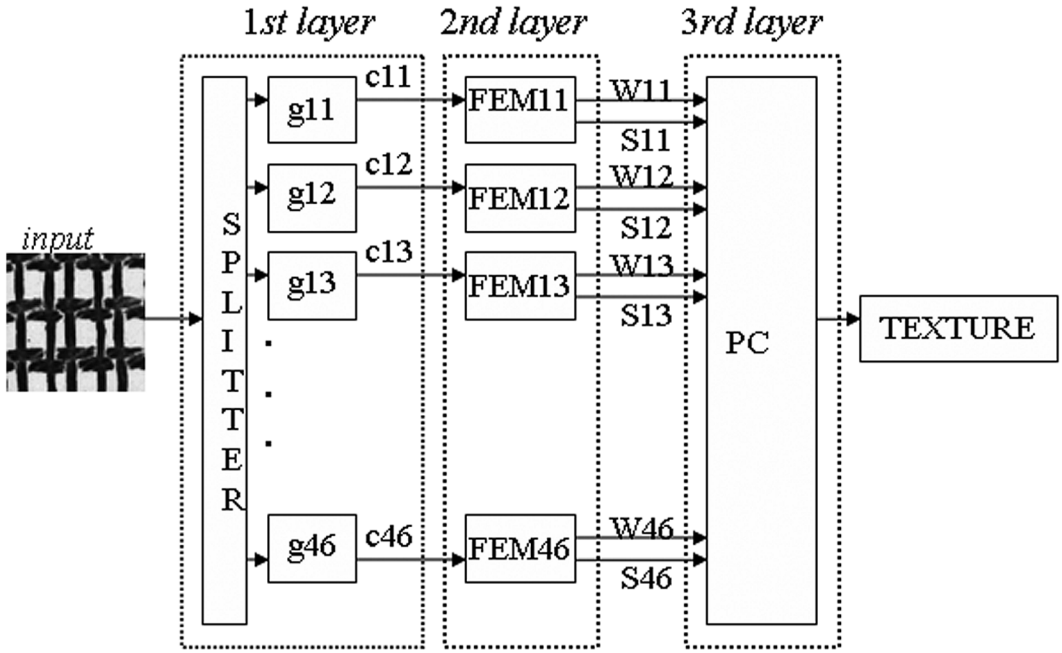

Fig. 6. Scheme of the AER-based system implemented for texture-based retrieval of images.

are used to represent the region for classification and retrieval purposes. In our AER implementation, we will not compute σ_{mn} as given in (5), but as

$$\sigma_{mn} = \int\int ||W_{mn}(x,y)| - \mu_{mn}|\, dxdy \tag{6}$$

without any degradation in performance. A feature vector is now constructed using μ_{mn} and σ_{mn} as feature components. In the experiments, we use four scales and six orientations, resulting in a 48 component feature vector

$$f = [\mu_{oo}\sigma_{00}\mu_{01}\sigma_{01}\cdots\mu_{35}\sigma_{35}] = [\mu_{mn}\sigma_{mn}]_{\substack{m=0,\ldots,3\\ n=0,\ldots,5}}. \tag{7}$$

Consider two image patterns i and j, and let $\bar{f}^{(i)} = [\mu_{mn}^{(i)}\sigma_{mn}^{(i)}]$ and $\bar{f}^{(j)} = [\mu_{mn}^{(j)}\sigma_{mn}^{(j)}]$ represent the corresponding feature vectors. The distance between the two patterns in feature space is then defined as

$$d(i,j) = \sum_m \sum_n d_{mn}(i,j)$$
$$d_{mn}(i,j) = \left|\frac{\mu_{mn}^{(i)} - \mu_{mn}^{(j)}}{\alpha(\mu_{mn})}\right| + \left|\frac{\sigma_{mn}^{(i)} - \sigma_{mn}^{(j)}}{\alpha(\sigma_{mn})}\right| \tag{8}$$

where $\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$ are the standard deviations of the respective features over the entire database, and are used to normalize the individual feature components. For database texture retrieval, the feature vector $\bar{f}^{(i)}$ of a new input image is compared with a precomputed database of feature vectors $\bar{f}^{(j)}$. Computation of $d(i,j)$ is fast. However, computing the feature vector is a slow process in conventional computers.

B. Adaptation of Manjunath's Method to AER Convolutional Event-Based Hardware

Our AER system implements a slightly modified version of the algorithm originally proposed by Manjunath for texture retrieval. The AER system is shown in Fig. 6. It has three layers. The first one is composed of a splitter module and 24 AER convolution modules in parallel. It implements a Gabor filter bank with four scales and six orientations. In [40], this configuration of filters was demonstrated to provide the best results. An input texture image is coded by events at intervals of 50 ns. These events are fed to a splitter module that replicates them on the 24 output channels. Each output channel is connected to a convolution module g_{mn} that uses as kernel the real part of a Gabor wavelet with scale m and orientation n. In the system of Fig. 6, each convolution module in the first layer is configured to change the sign bit of negative output events to positive (this is a full-wave rectification). This way, the output at each convolution module is $|W_{mn}(x,y)|$. Note that adding more modules to layer "1" increases the number of scales and orientations in the bank of Gabor filters. This improves classification performance. However, note that adding more modules to a layer will not increase the processing delay of the hardware.

Layer "2" consists of 24 feature extraction modules (FEM in Fig. 6). A FEM module is shown in Fig. 7. The first block is a splitter with three output channels. The top channel (labeled "2" in Fig. 7) goes directly to layer 3, thus providing an AER representation for $|W_{mn}(x,y)|$. The bottom channel (labeled "5") goes to an internal merger module with a hardwired positive sign. The central channel (labeled "3") goes to an internal mapper. This mapper ignores the address of the incoming event, and generates a new address by sequentially sweeping all addresses. Consequently, at the mapper output, a uniform AER image is represented with the same number of events as $|W_{mn}(x,y)|$. Thus, this represents the mean μ_{mn} of (5). This mean is fed to the internal merger with a hardwired negative sign. Consequently, at the merger output, we have all $|W_{mn}(x,y)|$ events with a positive sign and all μ_{mn} events with a negative sign. After convolving them with a unitary kernel C and changing the negative output event signs to positive, the

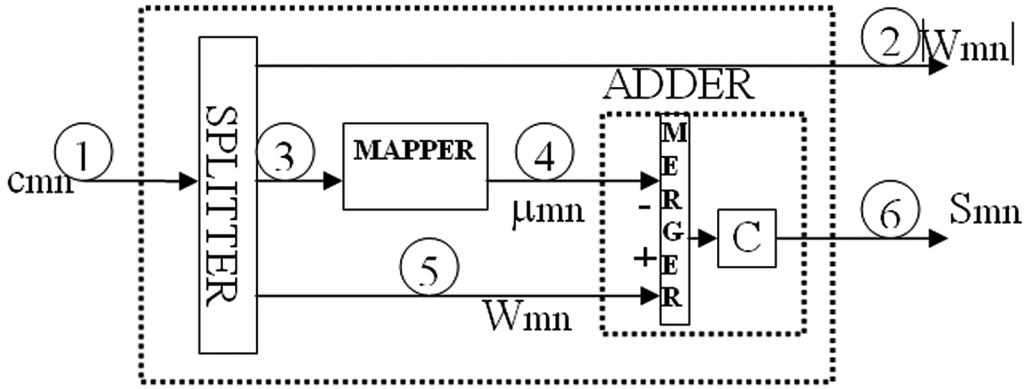

Fig. 7. Scheme of a generic FEM used in Fig. 6.

output will represent

$$S_{mn}(x,y) = \left||W_{mn}(x,y)| - \mu_{mn}\right|. \tag{9}$$

Finally, for each of its input channels $|W_{mn}(x,y)|$ and $S_{mn}(x,y)$, layer 3 will count the total number of events (regardless of their addresses) per unit time. We will use these numbers to create our feature vector described as

$$f = [W_{11}S_{11}W_{12}S_{12}\ldots W_{46}S_{46}]. \tag{10}$$

Numbers W_{mn} and S_{mn} will be a representation of μ_{mn} and $|\sigma_{mn}|$ and are the extracted characteristic feature vector for the input texture. Although slightly different from Manjunath's vector in (7), retrieval performance will not degrade, as shown in Section V and in the Appendix.

V. Results

In this section, we provide a performance evaluation of an eventual hardware implementation. For this we use a behavioral simulator developed in Visual C++ [29], [33] which allows to test large modular AER systems. The performance characteristics of the AER modules employed (convolution chips, mergers, splitters, and mappers) are obtained from already manufactured, tested, and reported AER modules [27], [29], [32], [39]. Unfortunately, those AER chips are presently experimental prototypes and only a small number of them are available. At this moment, it is therefore not possible to assemble large AER systems like the ones discussed in this paper. However, using the module performance characteristics together with the AER behavioral simulator, we can obtain a good estimate of the overall system performance.

We have used the Brodatz database [41], which consists of 112 images and each image has been divided into 16 90×90 nonoverlapping subimages, thus creating a database of 1792 texture images. These images have been rate-coded into events separated by 50 ns, creating stimulus bursts of 30 ms on average.[4] We used our C++ behavioral simulation tool to estimate the performance of an eventual hardware implementation. The 48 channel outputs of layer 2 (see Fig. 6) obtained for each of the images in the database were collected during the 30 ms (duration of the input burst) to create the feature vector database. In what follows, a query pattern is any one of the 1792 patterns in the database. This pattern is then processed to compute the feature vector as in (10). The distances $d(i,j)$, where i is the query pattern index and j is the index of a pattern from the database (with $i \neq j$), are computed and sorted in increasing order. Only the closest set of patterns are retrieved. Ideally, all top 15 retrievals are from the same large image. The performance is measured in terms of the average retrieval rate which is defined as the average percent number of patterns belonging to the same image as the query pattern in the top 15 matches. Table I summarizes the results. It shows the retrieval accuracy of the different texture features for each of the 112 texture classes in the database when we compare our AER-based method with the original Manjunath results. As can be seen, the retrieval accuracies are approximately equal.

To estimate the minimum time for correct texture retrieval, we proceeded as follows. Input stimuli lasted for about 30 ms. Layer 3 counts events coming from the 48 layer 2 output channels during a time T_{count}. This time was increased in steps of 15 μs from 0 to 30 ms. We found that for T_{count} approximately equal to 10 ms the results shown in Table I were similar. Consequently, an AER hardware implementation would be able to achieve correct texture retrieval in about $T_{\text{rcg}} = 10$ ms. As an illustration, Fig. 8 shows the retrieval accuracy as a function of T_{count} for six of the texture images in [41]. As can be seen, after 10 ms, the retrieval accuracy has stabilized; this is 20 ms before the input stimulus is finished.

In the Appendix, retrieval performance is compared against other state-of-the-art texture retrieval algorithms. The conclusion is that retrieval rate is not degraded in an AER implementation, but speed response is dramatically improved since recognition is achieved before the equivalent frame becomes fully available (see Table III in Appendix).

VI. Discussion

AER is an emerging hardware technology with great potential for providing complex cortical-like sensory-processing systems. Of special interest is its potential for providing very fast spike-processing convolutional neural networks with complex hierarchical structures, similar to those found in biological cortex. Recent work on individual AER convolutional chips reveals the outstanding capabilities of such components as "bricks" for larger highly sophisticated and hierarchically

[4]This burst time is conceptually comparable to the frame time in a frame-based system.

TABLE I
RETRIEVAL PERFORMANCE FOR EACH OF THE 112 BRODATZ IMAGES. COMPARISON BETWEEN MANJUNATH'S FRAME-BASED METHOD AND THE PROPOSED AER EVENT-BASED METHOD

IMAGE	FRAME-BASED	AER-BASED	IMAGE	FRAME-BASED	AER-BASED	IMAGE	FRAME-BASED	AER-BASED
D1	100	100	D39	47.67	40.49	D77	100	100
D2	67.39	64.58	D40	25.75	38.44	D78	88.21	87.64
D3	100	100	D41	79.45	41	D79	100	100
D4	100	100	D42	19.72	22.04	D80	100	100
D5	39.45	65.09	D43	37.81	42.54	D81	100	100
D6	100	100	D44	40.55	37.41	D82	100	100
D7	19.18	22.55	D45	10.41	13.32	D83	100	100
D8	88.21	100	D46	86.02	66.62	D84	100	100
D9	93.69	96.35	D47	100	100	D85	100	100
D10	72.87	69.7	D48	75.06	57.91	D86	46.03	64.06
D11	100	100	D49	100	100	D87	100	100
D12	100	100	D50	82.74	89.69	D88	24.11	26.65
D13	19.18	23.06	D51	100	99.43	D89	19.18	33.31
D14	27.4	29.21	D52	89.31	58.94	D90	52.6	37.41
D15	78.9	99.43	D53	100	100	D91	15.89	16.4
D16	100	100	D54	80	87.64	D92	100	99.42
D17	100	100	D55	100	100	D93	100	98.91
D18	52.6	66.62	D56	100	100	D94	100	100
D19	100	90.71	D57	100	100	D95	100	97.37
D20	100	100	D58	14.25	15.37	D96	66.85	72.26
D21	100	100	D59	37.81	45.1	D97	36.16	59.96
D22	100	100	D60	39.45	51.25	D98	33.97	43.56
D23	21.92	33.31	D61	33.42	44.07	D99	19.72	24.09
D24	100	100	D62	32.33	33.83	D100	45.48	49.2
D25	56.98	55.86	D63	38.35	43.56	D101	100	99.43
D26	96.43	71.24	D64	100	100	D102	100	100
D27	31.23	38.44	D65	100	100	D103	98.08	100
D28	64.65	75.85	D66	76.71	87.12	D104	77.26	87.64
D29	100	100	D67	49.86	48.18	D105	100	94.3
D30	24.66	36.9	D68	100	100	D106	100	100
D31	21.37	15.89	D69	80	83.54	D107	32.87	11.79
D32	100	100	D70	95.89	97.89	D108	13.15	13.84
D33	94.79	100	D71	98.08	100	D109	72.32	66.11
D34	100	100	D72	35.07	33.82	D110	100	86.1
D35	100	100	D73	20.27	27.68	D111	71.78	78.41
D36	95.89	84.05	D74	56.44	37.93	D112	52.6	57.4
D37	100	100	D75	84.38	92.76			
D38	100	93.79	D76	100	100	AVERAGE	73.21	73.89

Average Retrieval Rate (in %) for all 112 images of the Brodatz Database. The table compares two cases: Manjunath's frame-based method, and the AER event-based method analyzed in this paper. Each of the 112 database images is divided into 16 non-overlapping images, creating a database of 1792 texture sub-images. A query sub-image is processed by the network to obtain its feature vector, and its distance to all feature vectors of the other 1791 sub-images. The smallest 15 distances are retrieved. If these 15 cases are from the same correct texture image, retrieval rate is 100%. If not, the Average Retrieval Rate is computed as the percent number of patterns belonging to the correct target image.

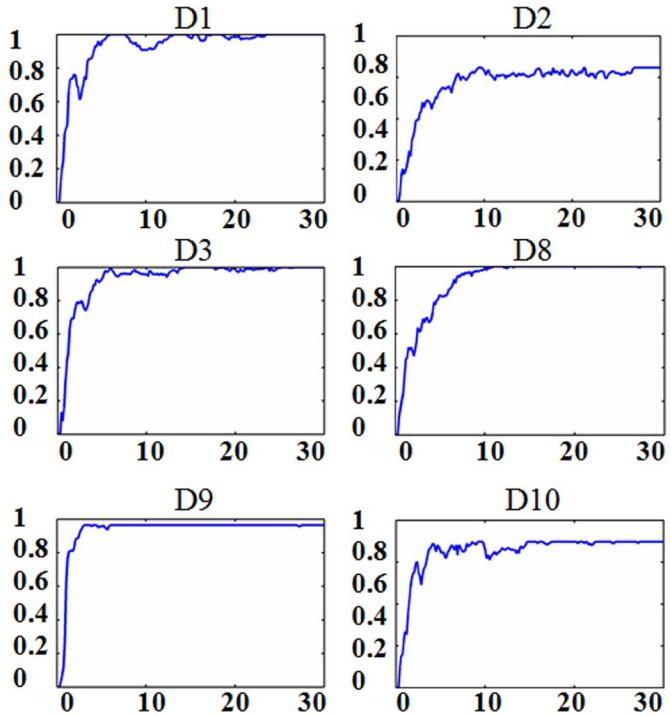

Fig. 8. Texture retrieval accuracy obtained for images D1-D2-D3-D8-D9-D10 as function of T_{count} (in milliseconds).

TABLE II
COMPARISON OF ARR USING THE BRODATZ DATABASE

METHOD	NUMBER OF CLASSES CONSIDERED (%)	ARR(%)
MDFB [64]	100%	73.00%
FAST MDFB1 [64]	100%	73.00%
FAST MDFB2 [64]	100%	73.00%
CONTOURLET TRANSFORM [64]	100%	71.00%
LOCAL AFFINE REGIONS [116]	100%	76.26%
LOCALLY INVARIANT DESCRIPTORS [66]	100%	78.50%
STANDARD REAL DWT (discrete wavelet transform) [67]	100%	64.17%
DT-CWT (Dual-Tree Complex Wavelet Transform) [67]	100%	76.83%
COMBINATION OF DT-CWT AND DT-RCWF (Dual-Tree Rotated Complex Wavelet filters) [67]	100%	78.93%
ROTATION INVARIANT GABOR FEATURE [54]	100%	59.00%
SCALE INVARIANT GABOR FEATURE [54]	100%	57.00%
MDFB [68]	100%	72.10%
STEERABLE PYRAMID [68]	100%	69.60%
FRACTAL-CODE SIGNATURES [69]	36%	53.2% - 85.3%
TPLP (three-pass layer probability) signature [70]	36%	82.10%
MANJUNATH APPROACH [35]	100%	73.2%
AER-BASED APPROACH	100%	73.89%

structured cortical-like sensory processing systems. To date, the largest AER multimodule system reported uses only four processing stages, one of which is a convolution [29]. We believe that we are not far from seeing systems made out of several hundreds (or thousands) of AER convolutional modules in the near future. NoC technology could host around 100 individual convolutional modules on a single chip, and about 100 such chips could be put on one single PCB. Consequently, a small physical volume like a desktop computer could easily hold 20–40 such PCBs, providing a total of almost half million convolution modules.

However, currently, it is not obvious what architectural structures should be used to assemble these AER convolutional "bricks" and how to set their parameters for a desired (recognition) application. In this paper, we have concentrated on one such possible application, texture recognition, emulated it with a behavioral AER simulator, and used it as an exercise to see how to set up such a system, its parameters, and estimate the performance of multilayer AER convolutional systems. There are starting to appear some software computational works in the literature that use massive convolutions for vision processing. For example, in texture recognition, experiments in the last years have demonstrated that filter-based schemes provide excellent results [42], [61]–[63]. However, massive convolutions on conventional computers result in excessive computational times, making such approaches nonpractical for real-world

TABLE III
COMPUTATION TIMES USING THE BRODATZ DATABASE

METHOD	Feature Extraction (FE) time (s)	Searching and Sorting time (s)	Total time (s) FRAME-based T_{FC}	Total time (s) EVENT-based T_{rcg}	SOFTWARE	HARDWARE
MDFB [64]	NOT AVAILABLE	NOT AVAILABLE	2.59		Matlab 6.5	CPU of Intel Pentium 4 2.4 GHz
FAST MDFB1 [64]	NOT AVAILABLE	NOT AVAILABLE	1.69		Matlab 6.5	"CPU of Intel Pentium 4 2.4 GHz
FAST MDFB2 [64]	NOT AVAILABLE	NOT AVAILABLE	1.62		Matlab 6.5	"CPU of Intel Pentium 4 2.4 GHz
CONTOURLET TRANSFORM [64]	NOT AVAILABLE	NOT AVAILABLE	1.38		Matlab 6.5	"CPU of Intel Pentium 4 2.4 GHz
STANDARD REAL DWT (discrete wavelet transform) [67]	0.47	0.06	0.53		MATLAB 5.3	CPU of Intel Pentium III 866 MHz
DT-CWT (Dual-Tree Complex Wavelet Transform) [67]	0.56	0.06	0.62		MATLAB 5.3	CPU of Intel Pentium III 866 MHz
COMBINATION OF DT-CWT AND DT-RCWF (Dual-Tree Rotated Complex Wavelet filters) [67]	1.05	0.09	1.14		MATLAB 5.3	CPU of Intel Pentium III 866 MHz
FRACTAL-CODE SIGNATURES [69]	NOT AVAILABLE	NOT AVAILABLE	0.42 - 18		NOT AVAILABLE	CPU of Intel Pentium 4 (2 GHz)
TPLP (three-pass layer probability) signature [70]	3.3	4.78	NOT AVAILA-BLE		Visual C++ 6.0	CPU of Intel Pentium 4 (2 GHz)
MANJUNATH APPROACH [35]	9.3	1.02	10.32		MATLAB 5	CPU of SUN Sparc20
AER-BASED APPROACH	0.01	0.01		0.02	---	AER-BASED DEVICES

applications. In general, vision processing researchers tend to avoid the use of convolutional processing because of its excessive computational load. For example, quoting Serre *et al.* [3] who use a first stage with 64 Gabor filters (for an input image of 128×128 pixels), the main limitation of their powerful recognition system is the delay of this first stage, which requires several tens of seconds. An AER-based spiking hardware could perform this processing with delays of a few milliseconds, or fractions of milliseconds, while the visual input is being sensed.

In all reported approaches for texture recognition, there is a relationship between the length of the feature vector and the computational time. The longer the feature vector, the longer the feature extraction time. In AER convolutional hardware, this is not the case, because all the elements of the feature vector are computed in parallel. Consequently, it is possible to increase the feature vector length or elements [54] to improve retrieval rate, without increasing feature extraction time, although at the cost of using more hardware "bricks." Actually, novel approaches for texture retrieval are based on the use of filters that take into account more frequencies or scales [64], [67] and produce less redundant features as compared to other wavelets (Gabor wavelet in our case).

This property is not specific for the texture retrieval application, but is generic for AER convolutional hardware: increasing the number of convolutional filters in a layer does not degrade speed response of the overall system. This is because the filters receive the same input events simultaneously and process them in parallel. There will be some delay the hardware will add to distribute the events to a larger number of receivers, but this extra delay will be in the order of nanoseconds, and consequently not perceived by the overall system. We have observed that the main potential for introducing delays in a multimodules AER system comes from the finite bandwidth of individual AER links. For present day reported AER links, a typical bandwidth is in the order of 10–30 Meps (mega events per second). Retina sensors output event rate is usually below 1 Meps. However, when merging several AER module outputs into one single AER channel, especially if we are thinking of several hundreds for the near future, it is realistic to expect that the limited AER link bandwidth could easily end up being the main delay bottleneck for such systems. Solutions for this problem could be to do a hierarchical merging of outputs combined with replicating the number of AER links to increase bandwidth. Also, we have observed that event traffic is higher for the first stages and is gradually reduced as convolutional processing compresses and extracts relevant information.

Perhaps the most interesting observation is that in AER sensory processing hardware, processing is performed as events are communicated between modules. As a retina is sending out its events they are sent directly to the processing structure and are processed as they flow in. In the same way, each "brick" processes its input events as they flow in and generates new ones. This way the whole system operates as if a wave of (visual) information (in the form of flow of events) travels through the convolutional structure while it is processed. Since processing is on a per event basis, stages do not wait for transmitting full "images" before processing them, thus reducing drastically the latency between input and output information flow.

What we have found with the specific example we have analyzed in this paper is that when mapping a know convolutional processing (frame-based) algorithm to AER hardware: 1) the recognition performance remains similar and also comparable to state-of-the-art computational methods not based on convolutions (or filters), and 2) if some day we are able to build physically this hardware, it will be capable of providing output recognition while the input stimulus is being produced by the sensors.

VII. Conclusion and Future Work

This paper shows performance results for a relatively large multimodule multilayer convolutional neural network frameless AER processing system, estimated through behavioral simulations but using performance figures of real individual AER hardware modules already available. A texture classification system based on Manjunath's method has been analyzed. This scheme uses 48 AER convolutional modules plus a similar number of interfacing modules, such as splitters mergers and mappers. We have shown that the recognition performance of the AER system is equivalent to its original frame-based reference. However, if built with realistic AER hardware, recognition is achieved while the sensory stimulus is being generated. This would be equivalent to stating that an AER system has a negative processing delay when compared to a frame-based system, where each frame has to be fully available before starting any recognition computation.

Thus, AER systems reveal some interesting properties. First, they are not constrained to frames and the output is often available even before the input stimulus has finished. Processing delay is given mainly by the number of layers and the number of events needed to represent the input stimulus. The processing capability of such systems is increased by adding more modules per layer, but without increasing the number of layers. Consequently, processing capability can be increased without penalizing delays, although at the cost of adding hardware.

Currently, the available AER hardware modules are quite preliminary, although their performance figures provide very promising system level performance estimations. Future work is focused mainly on miniaturizing present AER modules so that a large number of them (several hundred) could fit on a single PCB or in a large NoC chip. Also, such multimodule elements should allow a large degree of reconfigurability and reprogrammability, so that many different applications can easily be set up. In parallel with the hardware developments, future work also has to focus on analyzing other system level applications, while developing new theoretical frameworks more specific to event-based frameless processing and learning techniques.

Appendix
Comparison to State-of-the-Art Texture Retrieval

The commonly used methods for texture characterization can be divided into three categories: statistical, model-based, and filtering approaches [42]. Statistical methods such as cooccurrence features [43], [44] describe the tonal distribution in textures. Model-based methods such as Markov random field (MRF) [45] and simultaneous autoregressive (SAR) models [46] provide a description of texture in terms of spatial interaction. Most of the statistical and model-based approaches for texture classification consider spatial interactions over relatively small neighborhoods. Therefore, these approaches are more apt only for microtextures [47], [48]. Filtering approaches including wavelet [49], [50], Gabor filters [47], [51], steerable pyramid [52], and directional filter bank (DFB) [53], [54] characterize textures in the frequency domain. Among the three categories, MPEG-7 has adopted Gabor-like filtering for texture description [55]. The rationale behind is that visual cortex is sensitive to localized frequency components [56]. It has been shown that the direction together with scale information is important for texture perception. In the last decade, researchers have been combining different methods in order to provide a better classification and retrieval of images. Fusion of different types of texture features can be found in the literature [57]–[60]. A comprehensive performance evaluation on filtering (i.e., spectral-based) methods for texture classification is presented in [42], which suggests that no single set of features derived from filtering approaches has consistent superior performances on all textures. Other comparative studies about all these methods can be found in [61]–[63].

In [64], two fast algorithms for multiscale directional filter banks (MDFB) are proposed. These two algorithms are compared with the previous algorithm for MDFB proposed in [68] and with the contourlet transform [71], [72] in terms of time of feature extraction (FE) and total computational time. In [65], a texture representation suitable for recognizing images of textured surfaces under a wide range of transformations, including viewpoint changes and nonrigid deformations is presented. At the feature extraction stage, a sparse set of affine Harris and Laplacian regions is found in the image. Each of these regions can be thought as a texture element having an elliptic-shape characteristic and a distinctive appearance pattern. The approach achieves a maximum average retrieval rate of 76.26% when combined Harris and Laplacian descriptor channels are used. In [66], a linear family of filters is introduced, which provides certain scale invariance, resulting in a texture description invariant to local changes in orientation, contrast and scale, and robust to local skew. Then, a texture discrimination method based on the χ^2 similarity measure is applied to the histograms derived from the filter responses. This approach achieves a maximum average retrieval rate of 78.5%. In [67], the authors propose an approach for rotation-invariant texture image retrieval by using a set of dual-tree rotated complex wavelet filter (DT-RCWF) and DT complex wavelet transform (DT-CWT) jointly. They make a comparison of average retrieval accuracy using standard real DWT, DT-CWT and a combination of DT-CWT and DT-RCWF. In [54], rotation-invariant and scale-invariant Gabor representations are proposed, where each representation only requires few summations on the conventional Gabor filter impulse responses. The results show that the new implementations behave better than the conventional Gabor-based scheme when rotated or scaled images are considered. However, a conventional Gabor-based scheme provides better results when no rotation or scaling is considered.

In [68], an MDFB is first proposed and it is compared with the Gabor filters in polar form [73] and steerable pyramid [74] in terms of retrieval accuracy. In [69], fractal-code signatures are proposed for texture-based retrieval of images. Fractal image coding is a block-based scheme that exploits the self-similarity hiding within an image. By combining fractal parameters and collage error, a set of statistical fractal signatures is proposed. In [70], image signatures constructed from the bit planes of wavelet sub-bands are presented [bit plane signature (BP) and three-pass layer probability (TPLP) signature]. As can be observed, the method that provides the highest ARR is filter based and is the combination of DT-CWT and DT-RCWF implemented by Kokare *et al.* [67].

In Table II, we compare our AER event-based method with those reported in [54] and [64]–[68] and with Manjunath approach [35] in terms of average retrieval rate (ARR) using the entire Brodatz database. In Table III, we compare our method with those published in [64], [67], [69], and [70] and also with Manjunath's method [35], in terms of computation times. We distinguish between a FE time [time required to obtain a feature vector of the type in (7)] and a searching and sorting time (additional time to classify texture: computation of terms d_{ij}, sorting them, and selecting the best match). The sum of both is the total computation time. Note that, because of the conceptual difference between a frame- and an event-based approach, total computation time for a frame-based system is T_{FC} (as defined in Fig. 2), while for an event-based system it is T_{rcg} (as defined in Fig. 2). Consequently, comparing the computational delay of the two approaches by simply comparing times T_{FC} and T_{rcg} is not a fair comparison. It is more realistic to either compare $T_{\mathrm{frame}} + T_{\mathrm{FC}}$ against T_{rcg}, or the time between a frame is fully available ($T_1 + \Delta$ in Fig. 2) and the computing system provides a recognition result: T_{FC} for a frame-based system against $T'_{\mathrm{FC}} = T_{\mathrm{rcg}} + t_d - T_{\mathrm{frame}}$ (see Fig. 2) for an event-based system. Note that the latter ends up being negative.

REFERENCES

[1] G. M. Shepherd, *The Synaptic Organization of the Brain*, 3rd ed. New York: Oxford Univ. Press, 1990.

[2] E. T. Rolls and G. Deco, *Computational Neuroscience of Vision*. New York: Oxford Univ. Press, 2002.

[3] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 411–426, Mar. 2007.

[4] R. DeValois, D. Albrecht, and L. Thorell, "Spatial frequency selectivity of cells in macaque visual cortex," *Vis. Res.*, vol. 22, pp. 545–559, 1982.

[5] R. DeValois, E. Yund, and N. Hepler, "The orientation and direction selectivity of cells in macaque visual cortex," *Vis. Res.*, vol. 22, pp. 531–544, 1982.

[6] P. H. Schiller, B. L. Finlay, and S. F. Volman, "Quantitative studies of single-cell properties in monkey striate cortex. Spatial frequency," *J. Neurophysiol.*, vol. 39, no. 6, pp. 1334–1351, 1976.

[7] 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.

[8] K. Fukushima and N. Wake, "Handwritten alphanumeric character recognition by the neocognitron," *IEEE Trans. Neural Netw.*, vol. 2, no. 3, pp. 355–365, May 1991.

[9] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Science and Neural Networks*, M. Arbib, Ed. Cambridge, MA: MIT Press, 1995, pp. 255–258.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] P. F. Rüedi, P. Heim, F. Kaess, E. Grenet, F. Heitger, P.-Y. Burgi, S. Gyger, 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 × 128 120 dB 30 mW asynchronous vision sensor that responds to relative intensity change," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb. 2008.

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

[23] 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, Analog Digit. Signal Process.*, vol. 45, no. 5, pp. 600–606, May 1998.

[24] E. Chicca, A. M. Whatley, P. Lichtsteiner, V. Dante, T. Delbruck, P. Del Giudice, R. J. Douglas, and G. Indiveri, "A multichip 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.

[25] 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.

[26] 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.

[27] R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jiménez, 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.

[28] 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.

[29] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, L. Camuñas-Mesa, R. Berner, M. Rivas, T. Delbrück, S. C. Liu, R. Douglas, P. Häfliger, G. Jiménez-Moreno, A. Civit, T. Serrano-Gotarredona, A. Acosta-Jiménez, and B. Linares-Barranco, "CAVIAR: A 45 k-Neuron, 5 M-Synapse, 12 G-connects/sec AER hardware sensory-processing-learning-actuating system for high speed visual object recognition and tracking," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1417–1438, Sep. 2009.

[30] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, P. Iyer, A. Singh, T. Jacob, S. Jain, S. Venkataraman, Y. Hoskote, and N. Borkar, "An 80-Tile 1.28 TFLOPS network-on-chip in 65 nm CMOS," in *Proc. IEEE Int. Solid-State Circ. Conf.*, Feb. 2007, pp. 98–99.

[31] L. Camuñas-Mesa, A. Acosta-Jiménez, T. Serrano-Gotarredona, and B. Linares-Barranco, "A fully digital event-based convolution processor chip for fast frame-less vision processing," *IEEE Trans. Circuits Systems*, submitted for publication.

[32] F. Gomez-Rodriguez, R. Paz, A. Linares-Barranco, M. Rivas, L. Miro, S. Vicente, G. Jimenez, and A. Civit, "AER tools for communications and debugging," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, DOI: 10.1109/ISCAS.2006.1693319.

[33] J. A. Pérez-Carrasco, T. Serrano-Gotarredona, C. Serrano-Gotarredona, B. Acha, and B. Linares-Barranco, "On the computational power of address-event representation (AER) vision processing hardware," in *Proc. XXII Int. Conf. Design Circuits Integrated Syst.*, Sevilla, Spain, Nov. 21–23, 2007.

[34] A. Delorme, L. Perrinet, and S. J. 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.

[35] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, Aug. 1996.

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

[37] T. Serrano-Gotarredona, A. G. Andreou, and B. Linares-Barranco, "AER image filtering architecture for vision-processing systems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 46, no. 9, pp. 1064–1071, Sep. 1999.

[38] D. H. Goldberg, G. Cauwenberghs, and A. G. Andreou, "Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons," *Neural Netw.*, vol. 14, pp. 781–793, 2001.

[39] 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.

[40] L. Chen, G. Lu, and D. Zhang, "Effects of different Gabor filter parameters on image retrieval by texture," in *Proc. 10th Int. Multimedia Model. Conf.*, 2004, pp. 273–278.

[41] P. Brodatz, *Textures: A Photographic Album for Artists & Designers*. New York: Dover, 1966.

[42] T. Randen and J. H. HusZy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 291–310, Apr. 1999.

[43] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, no. 5, pp. 786–804, May 1979.

[44] G. V. D. Wouwer, P. Scheunders, and D. Van Dyck, "Statistical texture characterization from discrete wavelet representations," *IEEE Trans. Image Process.*, vol. 8, no. 4, pp. 592–598, Apr. 1999.

[45] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, no. 1, pp. 25–39, Jan. 1983.

[46] R. L. Kashyap and R. Chellappa, "Estimation and choice of neighbors in spatial-interaction models of images," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 1, pp. 60–72, Jan. 1983.

[47] R. M. Haralick, K. Shanmugan, and I. Dinstein, "Texture features for image classification," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[48] A. Speis and G. Healey, "Feature extraction for texture discrimination via random field models with random spatial interaction," *IEEE Trans. Image Process.*, vol. 5, no. 4, pp. 635–645, Apr. 1996.

[49] T. Chang and C.-C. J. Kuo, "Texture analysis and classification with trees-tructured wavelet transform," *IEEE Trans. Image Process.*, vol. 2, no. 4, pp. 429–441, Apr. 1993.

[50] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Process.*, vol. 4, no. 11, pp. 1549–1560, Nov. 1995.

[51] G. M. Haley and B. S. Manjunath, "Rotation-invariant texture classification using a complete space-frequency model," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 255–269, Feb. 1999.

[52] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, Sep. 1991.

[53] J. G. Rosiles and M. J. T. Smith, "Texture classification with a biorthogonal directional filter bank," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2001, pp. 1549–1552.

[54] J. Han 1 and K.-K. Ma, "Rotation-invariant and scale-invariant Gabor features for texture image retrieval," *Image Vis. Comput.*, vol. 25, no. 9, pp. 1474–1481, Sep. 2007.

[55] T. Sikora, "The MPEG-7 visual standard for content description—An overview," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 696–702, Jun. 2001.

[56] J. J. Kulikowski and P. O. Bishop, "Fourier analysis and spatial representation in the visual cortex," *Experientia*, vol. 37, pp. 160–163, 1981.

[57] D. A. Clausi and H. Deng, "Design-based texture feature fusion using Gabor filters and co-occurrence probabilities," *IEEE Trans. Image Process.*, vol. 14, no. 7, pp. 925–936, Jul. 2005.

[58] D. A. Clausi, "Comparison and fusion of co-occurrence, Gabor and MRF texture features for classification of SAR sea-ice imagery," *Atmos. Oceans*, vol. 39, no. 4, pp. 183–194, 2001.

[59] S. Li and J. Shawe-Taylor, "Comparison and fusion of multiresolution features for texture classification," *Pattern Recognit. Lett.*, vol. 26, pp. 633–638, 2005.

[60] N. Qaiser, M. Hussain, A. Hussain, and N. Qaiser, "Texture recognition by fusion of optimized moment based and Gabor energy features," *Int. J. Comput. Sci. Network Security*, vol. 8, no. 2, pp. 264–270, Feb. 2008.

[61] C.-C. Chen and C.-C. Chen, "Filtering methods for texture discrimination," *Pattern Recognit. Lett.*, vol. 20, pp. 783–790, 1999.

[62] R. Picard, T. Kabir, and F. Liu, "Real-time recognition with the entire Brodatz texture database," in *Proc. Comput. Vis. Pattern Recognit.*, 1993, pp. 638–639.

[63] P. P. Ohanian and R. C. Dubes, "Performance evaluation for four classes of textural features," *Pattern Recognit.*, vol. 25, no. 8, pp. 819–833, 1992.

[64] K.-O. Cheng, N.-F. Law, and W.-C. Siu, "A novel fast and reduced redundancy structure for multiscale directional filter banks," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2058–68, Aug. 2007.

[65] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *Proc. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 319–324, 2003.

[66] M. Mellor, B.-W. Hong, and M. Brady, "Locally rotation, contrast, and scale invariant descriptors for texture analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 52–61, Jan. 2008.

[67] M. Kokare, P. K. Biswas, and B. N. Chatterji, "Rotation-invariant texture image retrieval using rotated complex wavelet filters," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1273–1282, Dec. 2006.

[68] K.-O. Cheng, N.-F. Law, and W.-C. Siu, "Multiscale directional filter bank with applications to structured and random texture retrieval," *Pattern Recognit.*, vol. 40, no. 4, pp. 1182–1194, 2007.

[69] M. Pi and H. Li, "Fractal indexing with the joint statistical properties and its application in texture image retrieval," *IET Image Process.*, vol. 2, no. 4, pp. 218–230, 2008.

[70] M. H. Pi, C. S. Tong, S. K. Choy, and H. Zhang, "A fast and effective model for wavelet subband histograms and its application in texture image retrieval," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3078–3088, Oct. 2006.

[71] M. N. Do and M. Vetterli, "Pyramidal directional filter banks and curvelets," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2001, vol. 3, pp. 158–161.

[72] M. N. Do and M. Vetterli, "The contourlet transform: An efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005.

[73] G. M. Haley and B. S. Manjunath, "Rotation-invariant texture classification using a complete space–frequency model," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 255–269, Feb. 1999.

[74] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *Proc. Int. Conf. Image Process.*, Oct. 1995, pp. 444–447.

José Antonio Pérez-Carrasco received the degree in telecommunication engineering from the University of Seville, Seville, Spain, in 2004. He is currently working towards the Ph.D. degree in event based vision processing at the Instituto de Microelectrónica de Sevilla, Sevilla, Spain.

His research interests include visual perception, real-time processing, pattern recognition, and VLSI circuit design applied to vision systems.

Begoña Acha received the Ph.D. degree in telecommunication engineering from the University of Seville, Seville, Spain, in July 2002.

Since 1996, she has been working at the Signal Processing and Communications Department, University of Seville, where she is currently Tenured Professor. Her current research activities include works in the field of color image processing and its medical applications. She is author of numerous papers both in journals and international conferences.

Carmen Serrano received the M.S. degree in telecommunication engineering and the Ph.D. degree in telecommunication engineering from the University of Seville, Seville, Spain, in 1996 and 2002, respectively.

In 1996, she joined the Signal Processing and Communication Department at the same university, where she is currently Tenured Professor. Her research interests concern image processing and, in particular, color image segmentation, classification and compression, mainly with biomedical applications. She is author of numerous papers both in journals and international conferences.

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.

Teresa Serrano-Gotarredona (M07) 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" 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.

Bernabé Linares-Barranco (M'94–F'10) 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." 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.