EVENT BASED VISION SENSING AND PROCESSING

J. A. Pérez-Carrasco[1, 2], C. Serrano[2], B. Acha[2], T. Serrano-Gotarredona[1], B. Linares-Barranco[1]

[1]Instituto de Microelectrónica de Sevilla (IMSE-CNM-CSIC)
[2]Dpto. Teoría de la Señal, ETSIT, Universidad de Sevilla

ABSTRACT

In this paper we briefly summarize the fundamental properties of spike events processing applied to artificial vision systems. This sensing and processing technology is capable of very high speed throughput, because it does not rely on sensing and processing sequences of frames, and because it allows for complex hierarchically structured cortical-like layers for sophisticated processing. The paper describes briefly cortex-like spike event vision processing principles, and the AER (Address Event Representation) techique used in hardware spiking systems. Then a texture-based image retrieval using the AER technique is proposed. Realistic behavioral simulations based on existing hardware characteristics, reveal that the application, although processing large kernel convolutions, is capable of performing recognition in less than 10ms.

Index Terms— address event representation (AER), image processing, texture retrieval.

1. INTRODUCTION

Machine vision systems usually operate by capturing and processing sequences of frames, 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, specially if several convolutions need to be computed in sequence for each input image frame. Biological brains do 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. Very active pixels will send more spikes than less active pixels. 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. Therefore, in biological brains [1], strong features are propagated and processed from layer to layer as soon as they are produced, without waiting to finish collecting and processing data of whole image frames. One big problem encountered by engineers when it comes to implement bio-inspired (vision) processing systems is to overcome the massive feedforward and feedback interconnections among the neural layers existing in the human vision processing system. The Address Event Representation (AER) [2],[3] approach is a possible solution. Fig. 1 illustrates the communication in a traditional point-to-point AER link [4]. The continuos-time state of the emitter neurons in one chip is transformed into a sequence of fast digital pulses (spikes) of minimum width (in the order of ns) but with inter-spike interval in the order of ms (neurons in the brain also emit fast spikes

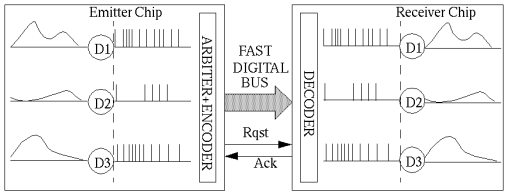

Fig. 1. Point-to-point AER link

separated times in the order of ms). This high inter-spike interval allows powerful time multiplexing, and the pulses generated by all the emitter neurons can be time-multiplexed in a common output digital bus. Each neuron is identified with an address. Each time a neuron emits a pulse or spike, that neuron address appears in the output digital bus, together with standar four-phase handshake signals for request (Rqst) and acknowledge (Ack). This is called an "Address Event". The receiver chip reads and decodes the addresses of the incoming events and sends pulses to the corresponding receiving neurons. The receiving neurons integrate those pulses and are able to reproduce the state of the corresponding emitter neurons. This is the simplest AER-based inter-chip communication. However, this point-to-point communication can be extended to a multi-receiver or multi-emitter scheme [5]. Moreover, translations, rotations or complex processing such us convolutions can be performed when the spikes travelling through the system come into one of the system´s post-processing chips. As an illustration, consider the setup in Fig. 2 (a). An image, with only three active pixels is received in a classical frame-based scheme and the convolution with a 3x3 convolution mask is performed pixel by pixel using (1):

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

In a AER-based approach (Fig.2 (b)), the same visual stimulus would be received by an AER-based retina, which will produce almost instantaneoulsy a sequence of pulses in a fast digital bus, where the address-events corresponding to those active pixels with higher value will appear now with a higher frequency than the address-events corresponding to pixels with lower activity (the value '2' in the image will produce a frequency two times higher than the value '1'). In the Figure, this implies that the address (3,3) will appear in the bus two times and the rest of active addresses only once in each period. If we wanted to add some kind of processing, such us convolutions, we could include convolution modules like the one reported in [6]. In this way, each time an event is received by the convolution module, its address coordinate is decoded and, a convolution mask (or also called here *projection-field* due to its similarity with projection-field processing in biological neuro-cortical layers), stored in the convolution module is added or subtracted to a pixel array (also stored in the

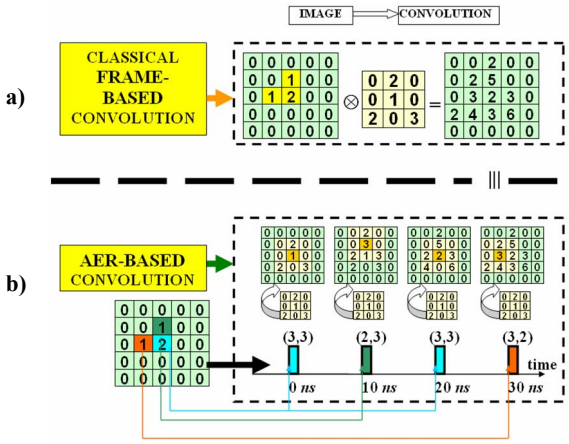

Fig. 2. Comparation between classical frame-based (a) and aer-based processing techniques (b).

convolution module) around the coordinate specified by the input event (each pixel in the matrix of pixels belonging to the convolution module contains one neuron and this neuron integrates the corresponding convolution mask value). This is shown at Fig. 2 (b). In this way, applying the mask as a projection field in the array of pixels in the convolution module, each pixel (or neuron), in the end, will have an activity that responds to (1), demonstrating that this is equivalent to the convolution between the input visual stimulus, source to the system, and the convolution mask stored in the convolution module, in the same way as it would have been done in a classical frame-based approach. The novelty of this aer-based scheme is that the visual stimulus has been sent to the convolution module and processed almost instantaneously, without waiting for an artificial-frame time. Also, we can introduce threshold levels in each of the pixels or neurons in the pixel array in the convolution module so that if one of the neurons reaches its threshold, indicating that it is receiving a higher activity, it will fire a spike and it will reset itself, in the same way as neurons in human brains do. Therefore, strong features are propagated and processed from layer to layer as soon as they are produced, without waiting to finish collecting and processing data of whole image frames. As we can see, this aer-based projection-field processing approach is structurally much faster than a conventional frame-based processing approach.

The goal of this paper is to initiate a link between the wide work being done in the world of bio-inspired electronic aer-based applications (but using simple image processing techniques) [3],[5],[7] and the new and complex advances in classical frame-based image processing. In this way, this article shows an image processing application of texture based image retrieval based on the method proposed by Manjunath [8], pretending to emulate the neural layers of the brain and based on AER.

2. TEXTURE-BASED RETRIEVAL OF IMAGES

To illustrate the potential of AER modules based spiking systems, and how AER chips and modules can be used in a multi-chip multi-layer AER structure, we will show in this Section a system for texture-based retrieval of images based on Gabor filters. Texture analysis algorithms range from using random field models to multiresolution filtering techniques such as the wavelet transform. The use of Gabor filters in extracting textured image

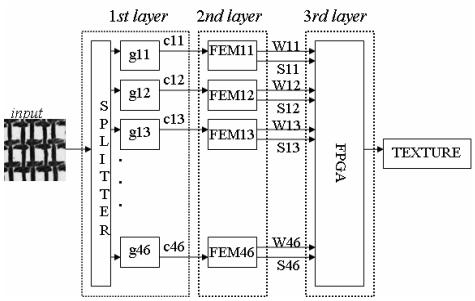

Fig. 3. Scheme of the system implemented based on AER for texture based retrieval of images.

features is motivated by various factors. The Gabor representation has been shown to be optimal in the sense of minimizing the joint two-dimentional uncertainty in space and frequency [9] and it has been demonstrated that using the Brodatz texture database, the Gabor features provide a very good pattern retrieval accuracy. Furthermore, on the theoretical side, an important insight has been advanced by Marcelja [10] and Daugman, [11] that simple cells in the visual cortex can be modeled by Gabor functions. For all these reasons, if we pretend to emulate the behavior of some layers of the brain, it make sense using the Gabor wavelets better than others for texture based retrieval of image data. The focus of this application is on the image processing aspects of the texture based retrieval processes. We have used in our system Manjunath's Gabor wavelet features for texture analysis [8] and provide a comprehensive experimental evaluation. The features are built by filtering the input image with a bank of orientation and scale sensitive filters and computing the mean and standard deviation of the output in the frequency domain. Gabor wavelets together with Manjunath´s Gabor features are still today being used in a lot of image analysis applications including image recognition, object recognition, image registration, medical applications and motion tracking. By performing texture analysis using Gabor filters at different scales and orientations, these patterns can be efficiently: 1) described in the frequency domain and 2)localized in the spatial domain.

The system implemented in this article is a sligthly modified aer-based version of the originally system proposed by Manjunath for texture retrieval. The system proposed is described in Fig. 3. This system has three layers. The first layer (layer '1' in Fig. 3) is composed by a splitter module and 24 aer-based convolution chips working in parallel [5]. It implements a Gabor filter bank with 4 scales and 6 orientations, that is, a total of 24 convolution chips. In the figure, a texture image coded by events separated each other 50ns, comes to a splitter module that replicates the input events in each of the 24 output channels. Each output channel is connected with a processing convolution module *Gmn* that uses as kernel the real part of a gabor wavelet with a determined scale *m* and orientation *n*. It has been well-demonstrated that the real part of the gabor wavelet performs as good as the complete gabor wavelet when it is used for texture retrieval applications. When a pixel in the array of pixels belonging to one of the convolution modules reaches its threshold value, it will reset itself and generate an output event, which will be sent off convolution module. Our processing chips are configured so that only positive events will be sent off the chips (positive and negative pixels will be reset when they reaches their threshold), so that the output obtained from the chip will be the events which number *Wmn* represents the convolution absolute value between the gabor wavelet with scale

m and orientation *n* and the input events belonging to the texture image under analysis. Each event obtained from one chip belonging to layer '1' comes to a processing module into layer '2'. Layer '2' consists of 24 feature extraction modules (FEM in Fig. 3) placed and working in parallel. One of the modules is described in Fig. 4. The first block in this layer is a splitter module with three output channels. The events travelling through the first one represent the absolute value of the convolution result *Wmn* obtained in the respective chip of layer '1'. The events coming out from splitter outputs 2 and 3 are used to calculate an estimation of the dispersion from the main value parameter using the following formula:

$$S_{mn} = \sum_x \sum_y \left(\left| W_{mn}(x,y) \right| - \mu_{mn} \right|) \quad (2)$$

This formula is computed by the system as follows:

The events in channel 3, come to a mapper module. This module produces output events that represent the mean value of the convolution result obtained in the previous layer.

Otherwise, events in channel 4 and events in channel 5 come to the two respective ports of an adder module that computes a standar deviation estimation of the visual flow entering layer '2'. This module is simply a 2-input merger block in sequence with a convolution chip with a 1x1 kernel of amplitude 1.5. Events coming to port one of the merger will be fed to the convolution chip with negative sign. In contrast, events coming to port two will be fed to the convolution chip with positive sign. In this way, the adder module computes the absolute value of the substraction between the filtered image represented by the events in channel 5 and the mean value represented by the events in channel 4, and this result is our estimation of the parameter *Smn* computed as indicated in (2).

Finally, Layer '3' consists of a FPGA that scans during a specified and programmed time the events from the 48 input channels, creating in this way a feature vector:

$$f = [W_{11}S_{11}W_{12}S_{12}...W_{46}S_{46}] \quad (3)$$

Afterwards, it computes, as described by Manjunath, the distance between the new-created feature vector and the feature vectors in the feature space corresponding to the patterns in the database. The texture under analysis will have texture *k*, if the distance to the feature vector corresponding to the pattern *k* is the minimal one.

There is a lot of research made in the area of texture based image retrieval. Some of them use different filters, different features distance measures or different number of scales or orientations [12],[13],[14]. In all of them our system is still valid due to the fact that we can adapt our system, changing the kernels programmed in the convolution chips, changing the distance measurement formulas implemented in the FPGA or adding new convolution chips in parallel so that new scales or orientations can be analysed, increasing the performance of our system and maintining the time previously computed for the retrieval process.

3. EXPERIMENTAL RESULTS

At present we don´t need the hardware infrastructure to illustrate the behavior of any kind of multi-layer multi-processing system experimentally, because we have designed a Matlab toolbox [15],[6] to emulate behaviorally the operation and timing of any aer-based system and particularlly, the system described in the previous section.

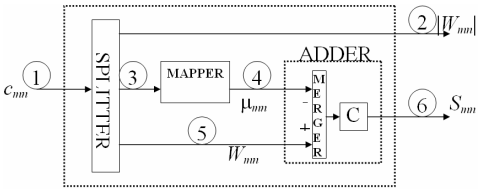

Fig. 4. Scheme of a generic Feature Extraction Module (FEM).

AVERAGE RETRIEVAL RATE (%)

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	78,45	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

Fig. 5 Retrieval performance obtained for each of the 112 Brodatz images.

The system previously described has been tested using 112 images from de Brodatz database [16]. Each image has been divided into 16 90x90 nonoverlapping subimages, thus creating a database of 1792 texture images. These images have been coded into events separated each other 50ns and creating an stimulus flow of 30ms on average using our Matlab tool, and they have been sent to the system. The 48 channels from layer '3' obtained from each of the image into the database have been scanned and its events have been collected during 30ms for creating the features vector database. Finally, the retrieval performance is computed as described by Manjunath in [8]. Fig. 5 provides a summary of the experimental results. It shows the retrieval accuracy of the different texture features for each of the 112 texture classes in the database when compared with the results obtained by Manjunath and that we have computed using Matlab. As it can be seen, the results are aproximately equal, but we need a time extremely lower (less than 10ms) to complete the retrieval process.

To demonstrate that we need only less than 10ms to detect the texture under analysis, the performance of the system has been evaluated varying the scanning time for collecting the events from the 48 channels in layer '3'. We have used a maximum scanning time of 30ms (because it is the duration of the input stimulus) and we have evaluated the average retrieval rate collecting events in intervals of 30/200 ms. In Fig. 6 the retrieval rate obtained for the images D1-D2-D3-D8-D9-D10 of the Brodatz database are shown for each one of these intervals. As the Fig. 6 indicates, we need only to collect events during 10ms for getting the final accuracy value indicated in Fig. 5.

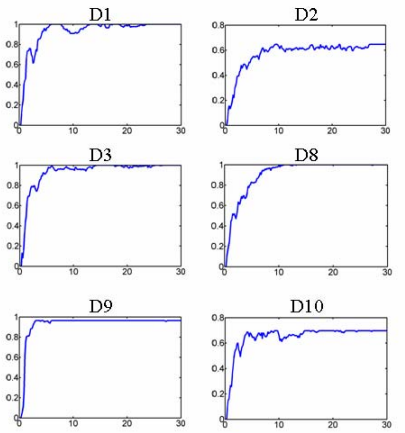

Fig. 6. Texture retrieval accuracy obtained for images D1-D2-D3-D8-D9-D10 for each of the intervals 30/200ms in which the duration of the total input stimulus has been divided.

4. CONCLUSIONS AND FUTURE WORK

The results obtained in the previous section indicate some of the clear advantages of using AER rather than the classical image processing methods based on frames:

1.- We do not need to wait for the complete image frame for starting processing in each of the modules belonging to one of the layers. In contrast, and from the beginning, we have output events in the implemented system travelling through layers '1' and '2' which are being processed and produce new output events to the following layers.

2.- We do not need to collect all the output events in layer '3' for completing the retrieval process. As the results indicate, with only the first few output events in layer '3', we are able to detect the input texture in the system.

3.- We are working always in parallel, so that it is possible to add new modules in parallel in layers '1' and '2' in the system. This fact will allow us to analyze more scales and orientations and always performing the results in the texture based retrieval process without computational cost.

4.- The system has 48 convolution modules. If we used the classical frame based methods for processing, we would have to wait at least for 48*tframe seconds (note that this value is computed without considering the post-processing time due to the convolution modules). In the classical frame methods, times higher than 0.38s approximmately have been published [12],[13] which suppose times extremely longer than the time needed when we work with our scheme based on AER (0.01s).

5. ACKNOWLEDGMENT

This work was supported in part by spanish grant TEC-2006-11730-C03-01 (Samanta2) and EU grant IST-2001-34124 (Caviar). JAPC was supported by the andalusian government grant P06-TIC-01417 (Brain System).

6. REFERENCES

[1] Gordon M. Shepherd, "The Synaptic Organization of the Brain," Oxford University Press, 3rd Edition, 1990.

[2] M. Sivilotti, "Wiring Considerations in analog VLSI Systems with Application to Field-Programmable Networks", *Ph D.Thesis*, California Institute of Techonology, Pasadena CA, 1991

[3] T. Serrano-Gotarredona, et al."AER image filtering architecture for vision processing Systems," *IEEE Trans. Circuits and Systems Part-II*, vol. 46, No. 9, pp. 1064- 1071, September 1999.

[4] K. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Trans. on Circuits and Systems Part-II*, vol. 47, No. 5, pp. 416-434, May 2000.

[5] R. Serrano-Gotarredona, et al. "AER Building Blocks for Multi-Layers Multi-Chips Neuromorphic Vision Systems" *Advances in Neural Information Processing Systems*, vol. 18, Y. Weiss and B. Schölkopf and J. Platt (Eds.), (NIPS'06), MIT Press, Cambridge, MA, pp. 1217-1224, 2006.

[6] R. Serrano-Gotarredona, José A. Pérez-Carrasco, B. Linares-Barranco, A. Linares-Barranco, G. Jiménez-Moreno, and A. Civit-Ballcells. "On Real-Time AER 2d Convolutions Hardware for Neuromorphic Spike Based Cortical Processing," *IEEE Trans. Neural Netw*.in Press. June 2008.

[7] A. Cohen, et al. *Report on the 2003 Workshop on Neuromorphic Engineering*, Telluride, CO, June 29 to July 19, 2003. {www.ini.unizh.ch/telluride}.

[8] 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.

[9] J.G. Daugman, "Complete Discrete 2D Gabor Transforms by Neural Networks for Image Analysis and Compression," *IEEE Trans. ASSP,* vol. *36,* pp. 1,169-1,179, July 1988.

[10] *S.* Marcelja, "Mathematical Description of the Responses of Simple Cortical Cells," 1. *Optical Soc. Am.,* vol. 70, pp. 1,297 1,300, 1980..

[11] J.G. Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters," *J. Optical* Soc. *Amer.,* vol. *2,* no. 7, pp. 1,160-1,169,1985.

[12] M. Kokare, P. K. Biswas, and B. N. Chatterji, "Texture Image Retrieval Using New Rotated Complex Wavelet Filters" *IEEE Trans.on Systems Man Cybernet*,2005,35(6):1168-1178.

[13] M. Kokare, et al. "Rotation-Invariant Texture Image Retrieval Using Rotated Complex Wavelet Filters," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 36, Issue 6, pp.1273-1282, Dec. 2006.

[14] L. Chen, G. Lu, D. Zhang, "Effects of Different Gabor Filter Parameters on Image Retrieval by Texture" *Proceedings of the 10th International Multimedia Modelling Conference*,pp. 273-278,2004.

[15] 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. *XXII Conference on Design of Circuits and Integrated Systems (DCIS)*. Sevilla, Spain, 21-23 November 2007.

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