Image convolution using a probabilistic mapper on USB-AER board

Rafael Paz-Vicente, Angel Jiménez-Fernández, Alejandro Linares-Barranco, Gabriel Jimenez Moreno, Francisco Gomez-Rodriguez, Lourdes Miró-Amarante, Anton Civit-Ballcels.

Arquitectura y Tecnología de Computadores. Universidad de Sevilla. Av. Reina Mercedes s/n, 41012-Sevilla, SPAIN

rpaz@atc.us.es

Abstract— In this demo we propose a method for computing real time convolution on AER images. For that we use signed events. The AER events produced on an AER retina or an image/video to AER conversor, are processed using a probabilistic multi event mapper that produces more than one event for each incoming event according to an assigned probability. Kernel convolution size are limited by mapping tables size (on board RAM) and AER bus bandwidth. On reconstruction signed events needs to be simplified (subtracted) to get final convolved image. For that two different methods are proposed.

I. INTRODUCTION

Address-Event-Representation (AER) was proposed in 1991 by Sivilotti [1] for transferring the state of an array of neurons from one chip to another. It uses mixed analog and digital principles and exploits pulse density modulation for coding information. The state of the neurons is a continuous time varying analog signal.

It explains the principle behind the AER. The emitter chip contains an array of cells (like, e.g., an imager or artificial retina chip) where each pixel shows a state that changes with a slow time constant (in the order of milliseconds). Each pixel includes an oscillator that generates pulses of minimum width (a few nanoseconds). Each time a pixel generates a pulse (called "event"), it communicates with the periphery and its address is placed on the external digital bus (the AER bus). Handshaking lines (Acknowledge and Request) are used for completing the communication.

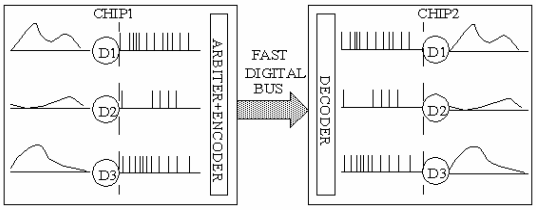

Figure 1. AER inter-chip communication scheme.

In the receiver chip the pulses are directed to the pixels or cells whose address was on the bus. This way, pixels with the same address in the emitter and receiver chips will "see" the same pulse stream. The receiver cell integrates the pulses and reconstructs the original low frequency continuous-time waveform.

Transmitting the pixel addresses allows performing extra operations on the images while they travel from one chip to another. For example, inserting memories (e.g. EEPROM) allows transformations of images.

There is a growing community of AER protocol users for bio-inspired applications in vision and audition systems, as demonstrated by the success in the last years of the AER group at the Neuromorphic Engineering Workshop series [2]. The goal of this community is to build large multi-chip hierarchically structured systems capable of performing complicated array data processing in real time. The CAVIAR EU project [5] had the objective to demonstrate this technology by targeting and following a moving ball. The planned AER system under CAVIAR uses the following AER chips: one Retina, four Convolutions, four Winner-Take-All (Object) and a Learning stage (a delay line chip and a learning chip).

In this paper we propose a method inspired in AER and neuronal alike events for computing convolution using an event mapper with probabilistic behaviour. For that a multi-event probabilistic mapper correctly configured is used for this purpose. Two sign events are used for negative coefficient and two methods for reconstructing final image merging positive and negative half-images (separate integration and event simplification) are presented and studied.

II. PROBABILISTIC MULTI-EVENT MAPPING

Basic transformation can be performed on images using event address mapping for each incoming event. Event address is used to index a event address table to obtain a new event address and send it out instead of original event. For 1:1 event mapping, only very basic transformation can be performed. One more complex mapping device can obtain more than a single event for each input event, so for each event received a variable number of different events are generated. Maximum number of outgoing events is only limited by mapping table memory size and hardware implementation.

Currently this limit is set on up to eight different events for each possible incoming event. A co-design USB AER interface based on Spartan II FPGA, Cygnal 8051 micro-controller and SRAM memory[10][11] board has 512Kx32 RAM memory. For 256x256 images, there are 64K different inputs address, so 512K/64K=8 slots.

AER mapping becomes more interesting if we associate a fixed probability for each different output event, and a multiplication factor. Each time a event arrives to USB-AER mapper device, input address is used to index in RAM memory and iterates on RAM memory to obtain associated output events, with probability. Probability value is compared with on-board random number generator, implemented using a free running LFSR register (Linear Feedback Shift Register). Only if probability value is greater than obtained from LFSR, output event is generated.

Probability relies on final pixel value. For an input event *i* of value In_i, desired output value is calculated multiplying probability by input value:

$$Out_i = P_i * In_i$$

This process is repeated for each incoming event as time as specified in multiplication factor field. So resulting output value is not limited by values below input one (probability is always between 0 and 1). Applying this repetition, resulting output value is:

$$Out_i = R_i * P_i * In_i$$

Reconstructed output value (number of events for a given address in a fixed period of time) correspond to theoretical value when image is integrated during enough time, so resulting image is the averaged value of several frames.

Also repetition factor is associated to each event, in the same way than probability so the event is going to be sent several times, extending the allowed values for the P_i coefficient in previous equation, from 0.1 to 0..M, being M the maximum repetition value allowed. Resulting coefficient for a given event is the result of multiply repetition factor by probability coefficient. For calculating repetition and probability, the following equation can be applied:

$$nrepeat = \lceil coefficient \rceil$$

$$probability = \frac{coefficient}{nrepeat}$$

For example, a desired kernel coefficient of *1.2*, repetition factor is set to *2* and probability to *0.6*.

III. Performing Convolutions

Complex filtering processing based on AER convolution chips already exists [12]. These chips are based on the integrate and fire neuron. Each time an event is received a kernel of convolution is copied in the neighborhood of the target pixel. Our solution is simpler since no analog implementation is required, nor adds calculations. We present another approach to this problem with reduced hardware using a programmable Xilinx Spartan II FPGA using advanced event mapping with multi-event and probability capabilities.

Bi-dimensional image convolution is defined mathematically by the following equation, being K the kernel convolution matrix and X the input image. Resulting convolved image is named Y.

$$Y(i,j) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} K(m,n) * X(i+m, j+n)$$

For a given output pixel Y(i,j), value is defined by X(i,j) and adjacent pixels, scaled by K coefficients. So a pixel X(i,j) defines the value of the output pixel Y(i,j) and their neighbors, multiplied by the same kernel coefficients. Using this reciprocity, it is possible to calculate for each input event, event outputs addresses and their associated probability. This computed values are loaded into probabilistic multi-event mapper.

We are going to calculate convolution for a sample image, with a sample 2x2 convolution kernel:

$$\begin{pmatrix} 0.75 & 0.1 \\ 0.1 & 0.05 \end{pmatrix}$$

In figure[2], original and convolved image are shown. Resulting image should be the initial one with soft-edges. Three USB AER boards are used in this demonstration. One for generating AER traffic for a static or motion image, one for performing convolution and last one for image reconstruction and PC based display.

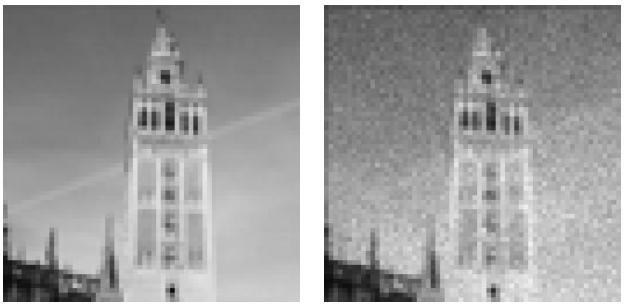

Figure 2. Original and convolved image

Recovered image is as expected, but even in simulations, it presents an added random noise. This noise is introduced in the image by random number generator, due to his unpredictable behavior when event emission is evaluated. This error is discussed later in this paper.

IV. Event Sign and Simplification

As negative probability is nonsense, convolution kernel elements are limited to positive values. To override this limitation, we can move sign from probability to event number, so when a negative coefficient is needed in convolution, the probability assigned for outgoing events is the absolute value of desired convolution coefficient, but output event address is replaced by a new event address that represents negative values. As a sample, edge detection kernel is going to be applied to an image. Kernel convolution is defined by following 2x2 matrix:

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

When input image is received, multi-event mapper generates two images, one for the positives coefficients of the kernel and other for the negatives ones. Negative image is right bottom shifted because convolution center is set in top-left element of the 2x2 matrix. In figure 3, is showed positive (left) and negative (right) images reconstructed separately.

Figure 3. Positive and negative image

For reconstructing a pixel value, it's needed to reconstruct two separate pixel values, negative and positive, and combine both subtracting them, or combine this reconstruction in a up-down counter model, so subtraction is calculated in the same time than integration. In both cases, result is the same (see fig. 4):

Figure 4. Positive and negative image merge

For a different approach, instead of simplifying positive and negative events on integration time, positive and negative events can be simplified delaying events on the bus, so when a positive event is received, emission is delayed until anther positive event for that address arrives. If a negative event is received, negative and delayed positive are removed. This way, the convolution is completely calculated "on the bus" so it can be applied to next AER neuronal layer, and reduces AER bus traffic.

This scheme can be extended to a n-bit counter so event re-emission is delayed more than one event. This can improve event elimination when positive and negative events are not well meshed and arrives as groups of positives or negatives events. In the ideal case, a pixel should never exist (be different to zero) on both positive and negative image. Existing on both images means that some event haven't been simplified.

Figure 5. Positive and negative images with events simplified

With this event simplification, resulting AER traffic represent convolution of original image so it can be applied to next layer.

V. ERROR INDUCED BY PROBABILITY

Some error is introduced due to non homogeneous event distribution. When probability is applied for a large stream of events for a given pixel, the average of events in the bus is the inputs events multiplied by probability. But due to probabilistic nature of this method, when a small amount of events are sampled, reconstructed value should not be exactly desired one, instead oscillate around it. In reconstructed image this effect is observed as random noise over the image.

For calculating how far is reconstructed value of desired, statistic variance is calculated according to following equation.

$$\sigma^2 = \frac{\sum (x_i - x_{mean})^2}{n}$$

Having that x_i is the reconstructed value, x_{mean} is the expected value (V_{in}*Prob.) and n the number of events. As showed in figure 6, maximum variance values are obtained when probability is around 50% (most randomness factor). For low probability values (most of events are not sent) and high probability values (most of event are sent), less indeterminacy are expected, so less error (variance) is observed.

In blue, variance observed for low integration time (below 10 frames). In red, integration time are increased for a thousand frames (1000 frames) for obtaining values less dependent of individual experiment.

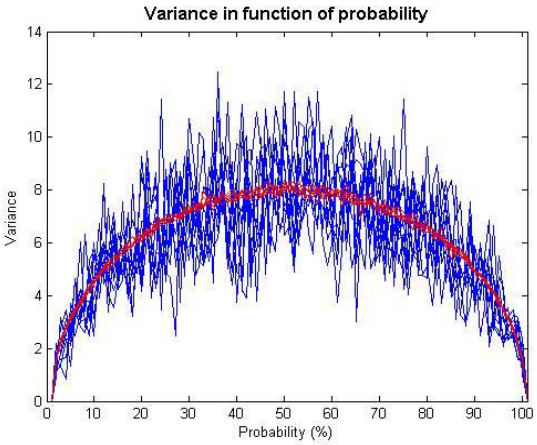

Figure 6. Variance vs probability

To reduce this noise, a possible solution is to increase the integration time, in the order of several times the frame, so resulting image is the average of several frames. Figure 7 shows normalized reconstructed value in function of the number of frames that integration process lasts, for a probability of 50%, so maximum random error influence is expected. For integration periods below five times frame-time, reconstructed value oscillates about a 15% around expected one. Increasing the integration time, reconstructed values oscillation reduces below 5%.

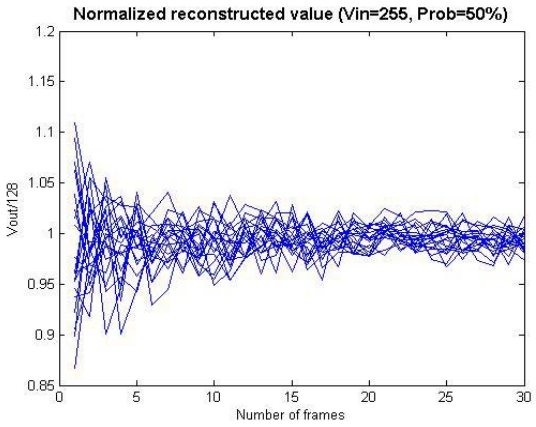

Figure 7. Normalized value vs integration time(number of frames)

VI. CONCLUSIONS

A bio-inspired method for real time image convolution based on AER bus and probabilistic event mapping has been presented on this paper. Some advantages on this probabilistic approach over integrate+convolve+generate approach is simplicity and that convolution is a time continuous process and not frame oriented.

Main disadvantage of this method is randomness of probabilistic event emission. Non homogeneous event distribution makes instant frequency reconstruction method to reconstruct higher error on images due to his higher sensibility to event time-location than traditional frame integration method. Better results are obtained when selected integration time on reconstruction is greater than a single frame time so more events are available for averaging.

ACKNOWLEDGMENT

This work was in part supported by Spanish project SAMANTA II (TIC-2006-08164-C03-02) and by the Andalussian Council project BrainSystem (P06-TIC-01417).

REFERENCES

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

[2] A. Cohen, R. Douglas, C. Koch, T. Sejnowski, S. Shamma, T. Horiuchi, and G. Indiveri, Report to the National Science Foundation: Workshop on Neuromorphic Engineering, Telluride, Colorado, USA, June-July 2001. [www.ini.unizh.ch/telluride]

[3] A. Wilen, J.Schade, R. Thornburg. "Introduction to PCI Express", Intel Press, 2003.

[4] Rafael Serrano Gotarredona, Matthias Oster, P. Lichtsteiner, Alejandro Linares Barranco, Rafael Paz Vicente, Francisco Gomez Rodriguez, H. Kolle Riis, T. Delbruck, S. C. Liu, S. Zahnd, A. M. Wathley, R. Douglas, P Halfliger, G. Jiménez, A Civit, Teresa Serrano, Antonio Jose Acosta, Bernabe Linares: Aer Building Blocks for Multi-Lawyer Multi-Chip Nueromorphic Vision Systems. Proceeding of the Neural Information Processing Systems Conference. Nips 2005: Vancouver, Canada. 2005. Pag. 1217-1225

[5] A. Linares-Barranco, G. Jimenez-Moreno, A. Civit-Ballcels, and B. Linares-Barranco. "On Algorithmic Rate-Coded AER Generation". IEEE Transaction on Neural Networks. May-2006

[6] M. Oster, Serverbased Software Architecture for AER systems [http://www.ini.unizh.ch/~mao/AerSoftware/SoftwareOverview.pdf]

[7] R. Paz-Vicente, A. Linares-Barranco, D. Cascado, S. Vicente, G. Jimenez, A. Civit. PCI-AER interface for Neuro-inspired Spiking Systems. ISCAS 2006. Kos, Greece.

[8] "Reverse engineering of the visual system using networks of spiking neurons". S.J. Thorpe, A. Delorme, R. Van Rullen, W. Paquier. ISCAS-2000, Geneva, Switzerland.

[9] Toby Delbrück (group leader) Neuromorphic Hardware group at the Institute for Neuroinformatics (INI) in Zurich, Switzerland [http://www.ini.unizh.ch/~tobi/caviar/index.php]

[10] AER-tools: http://www.atc.us.es/AERtools

[11] Rafael Serrano Gotarredona, Bernabe Linares Barranco, Teresa Serrano Gotarredona, Antonio Jose Acosta Jimenez, Alejandro Linares Barranco, Rafael Paz, Francisco Gomez, Gabriel Jimenez, Antonio Abad Civit Balcells: "High-Speed Image Processing With Aer-Based Components". IEEE International Symposium on Circuiits and Systems. ISCAS 2006 (). Kos, Grecia. Pag. 951-954

[12] Serrano-Gotarredona, R.; Serrano-Gotarredona, T.; Acosta-Jimenez, A.; Linares-Barranco, B.;A Neuromorphic Cortical-Layer Microchip for Spike-Based Event Processing Vision Systems. IEEE Transactions on [Circuits and Systems I Circuits and Systems I. Volume 53, Issue 12, Dec. 2006 Page(s):2548 - 2566