A Bio-inspired Feedforward System for Categorization of AER Motion Events

Bo Zhao[1], Qiang Yu[2], Hang Yu[3], Shoushun Chen[3] and Huajin Tang[1]

[1]Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore
[2]Department of Electrical and Computer Engineering, National University of Singapore
[3]VIRTUS IC Design Centre of Excellence, School of EEE, Nanyang Technological University, Singapore

***Abstract*—This paper introduces an event based feedforward categorization system, which takes data from a temporal contrast Address Event Presentation (AER) sensor. The proposed system extracts bio-inspired cortex-like features and discriminates different patterns using AER based tempotron classifier (a network of leaky integrate-and-fire (LIF) spiking neurons). One appealing character of our system is the event-driven processing. The input and the features are both in the form of address events (spikes). Experimental results on a posture dataset have proved the efficacy of the proposed system.**

I. Introduction

Recent years have witnessed increasing efforts in event-based neuromorphic sensory systems. One desire behind is to emulate the biological computation architecture which features asynchronous, sparse, event-driven signaling and processing. Address event representation (AER) sensors naturally provide a way to incorporate demand-based computation, where an asynchronous stream of digital data encapsulates only the relevant information. In particular, AER vision sensor allows pixel-parallel image processing at the focal plane. Each pixel in the sensor can individually monitor the relative change in light intensity and report an event if a threshold is reached. The output of AER vision sensor is a stream of address events. These sensors are often categorized as "temporal contrast" AER silicon retinas.

In order to fully utilize the power of AER sensors, the concept of "event" should be applied to every signal processing stage. Event-based object segmentation and tracking is studied in [1]. An embedded vision system is designed and combined with an AER vision sensor to achieve real-time object tracking through efficient event-based clustering. In addition, event-based convolution is applied to Convolutional Networks (ConvNets) in [2] to generate a frame-free AER-based ConvNet for feature extraction and categorization on AER visual event flow. Due to the frame-less event-based processing, AER-based ConvNets have a significant improvement in terms of input-output latency. However, the learning of the AER-based ConvNets is based on mapping from frame-based ConvNets but not naturally spike-based learning.

In this paper, we introduce an event-based feedforward categorization system, which consists of a temporal contrast AER vision sensor [3], a bio-inspired two-layer feature extraction unit, and a spiking neural network based on tempotron

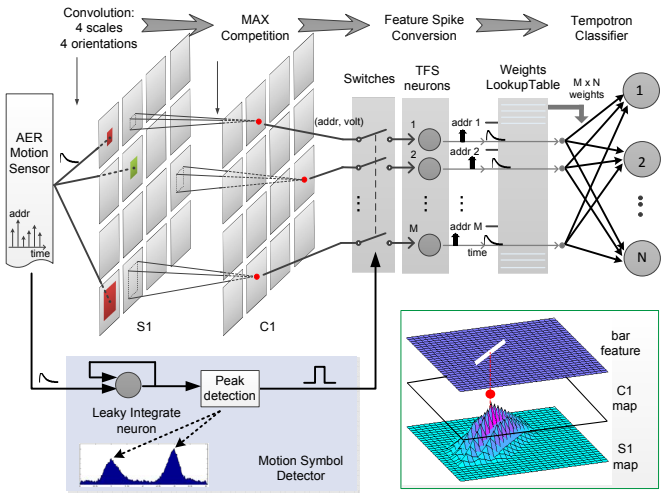

Fig. 1. Architecture of the categorization system. The system consists of several building blocks, namely convolution and competition, feature spike conversion, motion symbol detector and tempotron classifier. The lower right part of the figure illustrates the concept of our feature extraction. One S1 map is shown in the bottom. Corresponding $C1$ map in the middle has only one survived neuron due to MAX competition. The neuron's position is the same as that of $S1$ peak. The survived $C1$ neuron represents a bar feature of a certain size and orientation at that position.

neurons [4]. Each address event is sent in parallel to a group of Gabor filters to build up a set of $S1$ feature maps, then MAX-like neuron competition happens between $S1$ neurons and only the winners can survive in the higher layer $C1$. We proposed an asynchronous time-domain motion symbol detector to activate $C1$ feature spike generation. The dynamics of the survived neurons go through a small set of neurons that work in a Time-to-First Spike (TFS) fashion. The generated feature spikes, again in the form of address events, are fed to a network of tempotron neurons for classification. Our major contribution resides in two areas: 1) an efficient time-domain clustering algorithm to capture "motion symbols" and 2) a fully event based architecture that can emulate biology's use of asynchronous, sparse, event-driven signaling.

The rest of this paper is organized as follows. Section II describes system architecture, section III-V illustrate the system building modules. Experimental results are reported in section VI and conclusions are drawn in section VII.

II. System Architecture

Fig. 1 shows the architecture of the proposed system. One appealing character of our system is the fully event-driven processing. The flow of information processing is as follows.

1) **Feature Map Construction and Neuron Competition** Each address event from the AER vision sensor is projected onto a group of Gabor filters to build $S1$ feature maps [5], [6]. The response of each $S1$ neuron is changing dynamically due to on-the-fly convolution with a forgetting mechanism. Each $S1$ neuron competes with its local neighbors and only the winners can survive in the higher $C1$ layer.
2) **Motion Symbol Detection and Feature Spike Generation** Note that $S1$ and $C1$ maps are updated for each incoming AER motion event. In order to avoid doing classification on the feature maps all the time, a "Motion Symbol Detector" module is introduced into our system. This module consists of a leaky integrate type neuron and a peak detection unit. Each input event initiates a postsynaptic potential to this neuron. The peak detection unit monitors the total potential and will generate a pulse if a temporal potential peak is detected. The generated pulse will turn ON the switches in Fig. 1. At that particular moment, survived $C1$ neurons are fed to a set of "TFS neurons". where each feature is encoded into a time domain spike. Since only a small portion of neurons survive in $C1$ feature maps, we only need to build a few "TFS neurons".
3) **Categorization by Spiking Neuron Network** The classifier we use is a network of tempotron neurons. In principle, we need all the $C1$ feature spikes for classification and each tempotron neuron needs as many weights as the number of $C1$ responses. Thanks to our MAX operation and the AER spike representation, we can achieve a virtually fully connected system by physically activating only a very small subset of the network. Each feature spike is associated with an address, which can be used to access to a lookup table (LUT) and fetch a corresponding weight.

III. Feature Map Construction and Neuron Competition

Primates vision is extremely accurate and efficient in object categorization. The current theory of the cortical mechanism responsible for "rapid categorization" has been pointing to a hierarchical and mainly feedforward organization [7]. Inspired by feedforward models of cortical information processing [8], [9], we propose a two-layer hierarchical convolutional network to extract features from motion events. The overall data flow can be summarized as $motion\ events \rightarrow leaky\ postsynaptic\ potential \rightarrow S1\ maps \rightarrow C1\ maps$.

Each input event initiates a simplified postsynaptic potential (PSP) as shown in Fig. 2. At the time when we receive an input event, the PSP immediately increases to a unit potential and then decreases with a constant leakage rate as time goes by.

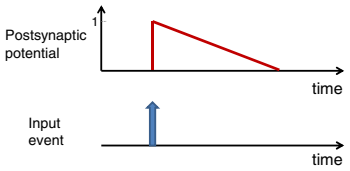

Fig. 2. Dynamic of the leaky neuron in on-the-fly convolution. Each input event initiates a PSP which first immediately increases to a unit potential and then decreases with a constant leakage rate.

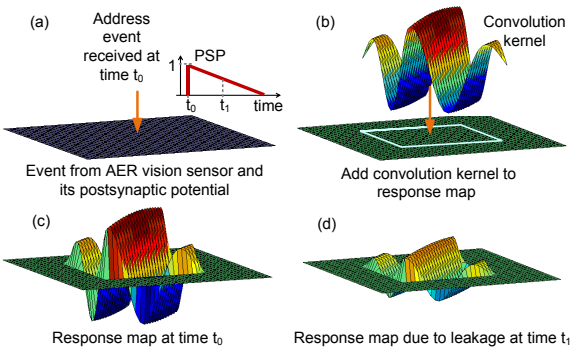

Fig. 3. On-the-fly convolution with a forgetting mechanism.

Simple cells ($S1$) are used to build object-selectivity. This is done by convolving the input event's postsynaptic potential with a network of Gabor filters. Each filter models a "neuron" cell with certain size of receptive field and responses best to basic feature at certain orientation. For the sake of hardware friendly implementation, we trade-off the network to 4 scales (ranging from 3 to 9, with a step length of 2) and 4 orientations (0^o, 45^o, 90^o and 135^o). The function of Gabor filter can be described as:

$$G(x,y) = \exp\left(-\frac{X^2+\gamma^2Y^2}{2\sigma^2}\right) \times \cos\left(\frac{2\pi}{\lambda}X\right) \qquad (1)$$

where $X = x\cos\theta + y\sin\theta$ and $Y = -x\sin\theta + y\cos\theta$. The filter parameters (orientation θ, aspect ratio γ, effective width σ and wavelength λ) have been well tuned in pioneering work [9], and here we adopt a similar set of these parameters.

The on-the-fly convolution is illustrated in Fig. 3. When an input address event comes in, the convolution kernel is overlaid onto the response map at the position specified by the input event's address; and then each element of the convolution kernel is used as a gain (or weight) to the event's postsynaptic potential, the product is added with corresponding original response, therefore the response map is updated.

In this way, we get 16 $S1$ convolution maps. $C1$ cells are further obtained by performing a MAX-like operation over simple $S1$ units. The MAX operation is performed across local neighborhood to find the center of the feature. "Neurons" located in different-scale $S1$ maps have different receptive fields, such as 3×3, 5×5, 7×7 and 9×9. Each "neuron" competes with all the other "neurons" located within its receptive field. It can survive in $C1$ layer only when itself is the MAX in this area.

After the MAX operation, each survival "neuron" on $C1$ map represents a feature, i.e, a line segment with certain size and orientation (see the lower right part of Fig .1).

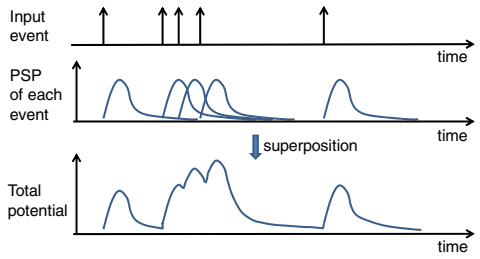

Fig. 4. Dynamics of the leaky integrate neuron in Motion Symbol Detector. Each input event generates a postsynaptic potential. The integration neuron's total potential can then be obtained by superposition.

IV. Motion Symbol Detection and Feature Spike Generation

In our system, the $C1$ feature map is updated with each input event coming in. To avoid doing classification all the time, we introduce a "Motion Symbol Detector" module to the system to select a good time for classification.

The term "symbol" is borrowed from speech recognition. The AER motion sensor only outputs a few noise events when capturing static scene, whereas generates a burst of output events when presented with moving objects. Here we use the word "symbol" to denote one slice from such a burst of output events. The "Symbol Detector" module consists of a leaky integrate type neuron and a peak detection unit. As illustrated in Fig. 4, each input event will contribute a postsynaptic potential (PSP) to this neuron. For an input event received at time t_i, the normalized PSP kernel K is defined as:

$$K(t-t_i) = V_0 \times (\exp(\frac{-(t-t_i)}{\tau_m}) - \exp(\frac{-(t-t_i)}{\tau_s})) \quad (2)$$

where τ_m and τ_s denote decay time constants of membrane integration and synaptic currents. For simplicity, τ_s is set to be $\tau_m/4$. V_0 normalizes PSP so that the maximum value of the kernel is 1.

The neuron's total potential is obtained by superposition.

$$V(t) = \sum_{t_i} K(t-t_i) + V_{rest} \quad (3)$$

where V_{rest} is the rest potential of this neuron, which is typically set to be 0. A peak detection unit is thereafter applied on the neuron's total potential to locate temporal peaks. The principle of peak detection is as follows. For one certain timing t_0, the potential at that timing is considered as a peak if the following criterion is met:

$$V(t_0) \geq V(t);\ \forall\, t \in [t_0 - t_{SR}/2, t_0 + t_{SR}/2] \quad (4)$$

where t_{SR} denote the time span of the search range. This means potential at time t_0 compares itself with all the potentials within its search range $[t_0 - t_{SR}/2, t_0 + t_{SR}/2]$, if its potential is the maximum, then it is considered as a peak.

When a peak is detected, the "Motion Symbol Detector" will generate a pulse, which will turn ON the switches in Fig. 1. At that particular moment, $C1$ feature maps are fed to the following processing stages. Note that, in order to avoid detection of very small peaks (for instance, when there is only one event happens during some time interval), a threshold should also be applied. In addition, we can also add a refractory time to limit the frequency of output pulses.

$C1$ features maps at a certain moment selected by symbol detector will be fed forward to a set of "TFS neurons". Each TFS neuron is in charge of the conversion of one response. The higher the response, the shorter the time to first spike. All TFS neurons work in parallel and should be triggered simultaneously. Thanks to our MAX surviving operation, only a small amount of neurons survive in $C1$ layer. Instead of using all $C1$ responses, we only forward the survival neurons' responses (non-zero ones) together with their unique addresses (positions within 16 $C1$ maps). In this way, the features are encoded into AER spikes again (a combination of time stamp and address). Time stamp is inversely proportional to the strength of the $C1$ feature, and the address indicated its unique position in $C1$ feature maps.

V. Categorization by Spiking Neural Network

In this section, we will illustrate how we perform classification on extracted feature spikes using a network of tempotron spiking neurons. Tempotron is a model of supervised temporal learning that allows a spiking neuron to efficiently discriminate spatiotemporal spike patterns [4]. It utilizes spike timing information and integrates postsynaptic potentials from afferent spikes with different addresses. These properties make tempotron by nature a perfect match for our extracted AER feature spikes.

A. Tempotron Learning Rule

Tempotron uses leaky integrate-and-fire (LIF) neuron model. Each input spike initiates a postsynaptic potential (PSP) which has an exponential decaying shape as described in Equation (2) The neuron's total membrane potential is the weighted summation of PSP from all input spikes:

$$V(t) = \sum_i \omega_i \sum_{t_i} K(t-t_i) + V_{rest} \quad (5)$$

where ω_i and t_i are the synaptic efficacy and the firing time of the ith afferent, respectively. V_{rest} is the resting potential of the neuron. K denotes the normalized PSP kernel.

If the neuron's potential is higher than a specified threshold, the neuron will fire an output spike. After firing, the neuron shunts all the following input spikes and the potential decreases to the resting level. In other words, the spikes arrive after the firing time have no impact on the postsynaptic potential any more. If the membrane potential fails to cross the threshold then the neuron will not fire.

The tempotron learning rule aims to train the weights of afferent synapses so that the output neuron can fire or not according to its class label. If the neuron is supposed to fire (or not fire, on the other hand) but it actually fails to do so (or does fire, vice versa), then the weights of afferent synapses should be modified in the following way: first, we find the peak potential during the effective period and label the corresponding time stamp as t_{max}; second, update the weights using the equation:

$$\Delta\omega_i = \lambda \sum_{t_i < t_{\max}} K(t_{\max} - t_i) \tag{6}$$

B. Virtually Connected Tempotron

In principle, we need all the $C1$ feature spikes for classification and each tempotron neuron needs as many weights as the number of $C1$ responses. Assume the resolution of AER vision sensor is $m \times n$, the number of responses in $C1$ layer will be $16 \times m \times n$ (as we use 16 filters). Therefore, each tempotron neuron should have $16 \times m \times n$ weights. In a N-class categorization task, we need N tempotron neurons (one for each category). The total number of weights are $16 \times m \times n \times N$. This means huge hardware resource. However, thanks to the MAX operation and the AER feature spike representation, we can achieve a virtually fully connected system by physically activating only a very small subset of the network. We use a lookup table (LUT) to store the weights. Each feature spike is associated with an address, which can be used to visit the LUT and fetch its corresponding weight.

During the training process, for a N-class categorization task, we label the N tempotron neurons using one-hot coding scheme. During testing, the decision making for each input pattern is easy: just to check which neuron fires. In order to further improve the performance, we can use multiple neurons for each category [10]. Since the initial weights are set randomly, these neurons will have different weights after training. We use majority voting scheme to make a final decision: to check which category's neurons fire the most.

VI. Experimental Results

We have evaluated the performance of the proposed algorithm on real AER motion events. We captured three human actions, namely bending to pick something up ($BEND$), sitting down and standing up ($SITSTAND$), and walking back and forth ($WALK$). Fig. 5 shows a few reconstructed sample images. Each row is corresponding to one action; images are reconstructed from the AER motion events, using a frame interval of 20 ms.

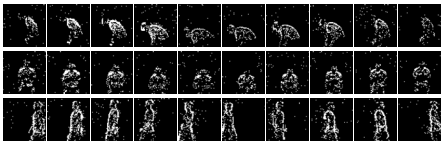

Fig. 5. Some reconstructed frames from our posture dataset. There are three kinds of human actions, each row shows an action.

The posture dataset consists of 191 $BEND$, 175 $SITSTAND$ and 118 $WALK$ actions. We randomly pick out 80% of these actions for training and the rest 20% are used for testing. By repeating this evaluation process five times, we get the average performance. For the training set, we obtain a correct rate of 100%; while for the testing set, the rate is 98.45% by average, with a standard deviation of 0.82%.

We also compared our approach with two popular biologically inspired algorithms: the original HMAX scheme [8] and the model by Serre et. al [9]. Both approaches use Support Vector Machine (SVM) as classifier. To perform multi-class categorization, one-versus-one (OVO) SVM scheme is employed. Since these two methods are designed for frames/images instead of events, our posture dataset can not be used directly. We first use the aforementioned "Motion Symbol Detector" to pick some slices from the event flow, and reconstruct those slices of events into images. The comparison results are promising, as shown in Fig. 6.

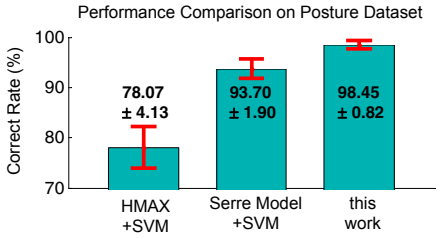

Fig. 6. Performance comparison on posture dataset. We compare our algorithm with HMAX [8] and Serre model [9].

VII. Conclusion

This paper presents a feedforward categorization system on AER motion events. The system uses event-based processing at every signal processing stage. Cortex-like features are extracted by a two-layer hierarchical convolutional network. Due to the MAX-like competition, features are encoded into a limited number of spikes. A virtually connected spiking neural network efficiently discriminates the feature spike patterns. Promising result has been obtained on a posture dataset and we are now working toward hardware implementation of the system. The aim of this work is to develop a realtime human posture categorization system using AER motion sensor.

References

[1] M. Litzenberger, C. Posch, D. Bauer, A. Belbachir, B. K. P. Schon, and H. Garn, "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *12th Signal Processing Education Workshop*, Sept. 2006, pp. 173–178.

[2] J. Perez-Carrasco, B. Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, S. Chen, and B. Linares-Barranco, "Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate-coding. Application to feed forward ConvNets," *IEEE Trans. Pattern Anal. Mach. Intell*, 2013.

[3] X. Zhang and S. Chen, "A high-speed-pass asynchronous motion detection sensor," in *IEEE ISCAS'13*, Beijing, China, May 2013.

[4] R. Gutig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions," *Nature Neuroscience*, vol. 9, no. 3, pp. 420–428, Feb. 2006.

[5] S. Chen, P. Akselrod, B. Zhao, J. Perez-Carrasco, B. Linares-Barranco, and E. Culurciello, "Efficient feedforward categorization of objects and human postures with address-event image sensors," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 34, no. 2, pp. 302–314, Feb. 2012.

[6] B. Zhao and S. Chen, "Realtime feature extraction using max-like convolutional network for human posture recognition," in *IEEE ISCAS'11*, May 2011, pp. 2673–2676.

[7] S. J. Thorpe and M. Fabre-thorpe, "Seeking categories in the brain," *Science*, vol. 291, pp. 260–263, 2001.

[8] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, pp. 1019–1025, 1999.

[9] 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, 2007.

[10] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Rapid feedforward computation by temporal encoding and learning with spiking neurons," *IEEE Trans. Neural Netw. Learning Syst*, in press 2013.