Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Frequency spike encoding using Gabor-like receptive fields

Taras Iakymchuk, Alfredo Rosado-Muñoz, Manuel Bataller-Mompeán, Juan F. Guerrero-Martínez, Jose V. Francés-Víllora

GPDS. Dpt. Electronic Engineering. School of Engineering. University of Valencia, Spain

Abstract: Spiking Neural Networks (SNN) are a popular field of study. For a proper development of SNN algorithms and applications, special encoding methods are required. Signal encoding is the first step since signals need to be converted into spike trains as the primary input to an SNN. We present an efficient frequency encoding system using receptive fields. The proposed encoding is versatile and it can provide simple image transforms like edge detection, spot detection or removal, or Gabor-like filtering. The proposed encoding can be used in many application areas as image processing and signal processing for detection and classification.

1. INTRODUCTION

Spiking neural networks are called as the third generation of neural networks. They operate on spikes and are able to provide different information as well as new algorithms, which are based on spikes and thus, the processing requires less computational resources than other techniques. For this reason, SNN are becoming a widespread topic in last years. SNN are modelled on the same principles as the biological neurons, they are versatile and powerful tools, capable of solving a wide range of tasks and used for controlling manipulators and robots (Bouganis and Shanahan, 2010; Alnajjar and Murase, 2008), recognition and detection tasks (Perez-Carrasco et al., 2010; Botzheim et al., 2012), tactile sensing (Ratnasingam and McGinnity, 2011), or neuromedical data analysis (Fang et al., 2009).

For SNN to operate optimally, signals must be encoded into spikes according to optimal methods since data representation has a key role in network functionality (Gerstner and Kistler, 2002). For visual data representation, frequency orientation-dependent spike encoding is used, as this encoding is similar to the animal visual cortex data representation. In this work we are analysing visual encoding methods and proposing a Gabor-like spike encoding improving the results in further SNN processing, being low computationally intensive. This paper is organized as follows. Section 2 describes general spiking neuronal model. Section 3 describes the usage of receptive fields for feature extraction. Section 4 introduces the Address-Encoding Representation of spiking data, the encoding widespread used in neuromorphic hardware systems. Sections 5 and 6 describe the experimental set-up and results. And Section 7 contains conclusions from the obtained results.

2. SPIKING NEURON MODEL AND NETWORK TOPOLOGY

In general, SNN are composed by a set of input neurons, internal neurons (in several layers) and output neurons. Each neuron is connected to all the neurons in the next layer by a delayed and weighted connection, which means that the output signal of a neuron arrives at a different delay time to the neurons in the next layer and it will have a different weighted potential contribution. Delay is a specific characteristic not present in other neural network proposals, implying more flexible and powerful structures, but more difficult to train and configure due to the different possibilities of weight and delay adjustment. The possibility of learning by delay plasticity is studied in (Pham et al., 2007).

It is common that electrical input signals do not correspond to spikes since they are not generated by biological systems. Thus, such input signals must be encoded into spikes or spike trains to further feed the SNN as if biological spikes were received. Typically, the coding information method is the integrate and fire coding (Gerstner and Kistler, 2002) where the input neuron is firing a spike regularly every time interval while the input stimulus is active.

An approximation to the functionality of a neuron is given by electrical models which reproduce the functionality of nervous cells. One of the most common models is the Spike Response Model (SRM) due to the close approximation to a real biological neuron (Paugam-Moisy and Bohte, 2009; Booij, 2004). The main characteristic of a spiking neuron is the membrane potential. The transmission of a single spike from one neuron to another is mediated by synapses at the point where the two neurons interact. In neuroscience, a transmitting neuron is defined as a presynaptic neuron and a receiving neuron as a postsynaptic neuron. Neurons have a small negative electrical charge of -70 mV, which is called resting potential. When a single spike arrives into a postsynaptic neuron, it generates a Post Synaptic Potential — PSP (excitatory when increasing potential and inhibitory when decreasing). The membrane potential at an instant is calculated as the sum of all present PSP. When the membrane potential is reaching a critical value

called threshold, a postsynaptic spike is generated. The PSP function is given by (1), where τ_m and τ_s are time constants to control the steepness of rise and decay.

$$f_{PSP(t)} = e^{(\frac{-t}{\tau_m})} - e^{(\frac{-t}{\tau_s})} \quad (1)$$

Figure 1 shows different PSP as a function of time (ms) and weight value. Excitatory potential arises in case of red and blue lines, and inhibitory potential in case of green line.

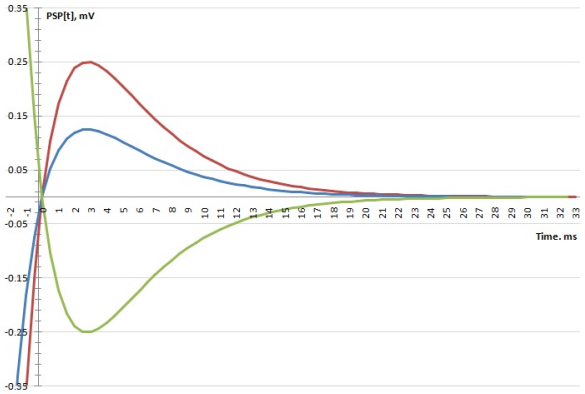

Fig. 1. Postsynaptic potential function (PSP) with weight dependency. Red line is for $\omega = 1$, green for $\omega = 0.5$ and blue is for $\omega = -1$.

Each input spike arriving to a neuron generates a PSP contributing to the membrane potential and is cumulative to other PSPs generated by other inputs. When the membrane potential reaches the threshold, the potential rises up rapidly and the neuron fires an output spike. After firing a spike, neuron potential goes into a short refractory period. During refractory time, the neuron cannot propagate a new action potential and voltage of soma membrane stays below the resting value. After the refractory period is finished, the neuron potential returns to its resting value and is ready to fire a new spike if membrane potential is above the threshold, i.e. when new spikes arrive. Let us consider an example shown in Fig. 2 where spikes from two presynaptic neurons trigger excitatory PSP in a postsynaptic neuron. The spike train generated by the presynaptic neurons change the membrane potential calculated as the sum of individual PSP generated by incoming spikes. When membrane potential reaches threshold, the neuron fires a spike at the time instant t_s.

If we denote the threshold value as v, the refractory period is defined according to (2) (Booij, 2004).

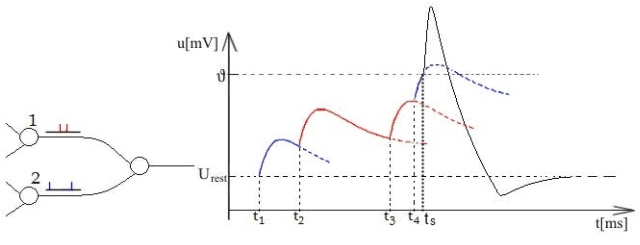

Fig. 2. Spike generation by a threshold process.

$$\eta(t) = -ve^{(\frac{t}{20})} \quad (2)$$

being $t_i^{(g)}$ the time when a spike is fired by a presynaptic neuron. This spike changes the potential of a postsynaptic neuron j at time t and the time difference between these two events is $t - t_i^{(g)}$. The travelling time between two neurons for a spike is defined by (3) where d_{ji} is the delay value of a synapse.

$$\Delta t_{ji} = t - t_i^{(g)} - d_{ji} \quad (3)$$

When a sequence of spikes $F_i = \{t_i^{(g)}, ..., t_i^K\}$ arrives to a neuron j, the membrane potential changes according to the PSP function and refractory period, and thus, an output spike train is propagated by neuron j as $F_j = \{t_j^{(f)}, ..., t_j^N\}$. The equation for the $j-th$ neuron potential u_j is obtained in (4), where the refractory period is also considered.

$$u_j(t) = \sum_i^K \sum_{t_i^{(g)} \in F_i} w_{ij} PSP(\Delta t_{ji}) + \sum_{t_j^{(f)} \in F_j} \eta(t - t_j^{(f)}) \quad (4)$$

3. NEURONAL RECEPTIVE FIELDS AND GABOR-LIKE FILTERS

3.1 Receptive Field

Visual neural cortex is one of the best studied parts of the brain. The Receptive Field (RF) of a visual neuron is the specified part of the image affecting the neural input. The size and shape of receptive fields vary heavily depending on neuron position and purpose. By modifying the receptive field a neuron can be more sensitive to an object position, orientation or shape. In each subsequent layer of visual cortex, receptive fields of the neurons cover bigger and bigger regions, convoluting the outputs of previous layer.

Mammalian retinal ganglion cells located at the center of vision in the fovea, have the smallest receptive fields and those located in the visual periphery have the largest receptive fields (Martinez and Alonso, 2003). The large receptive field size of neurons in the visual periphery explains the poor spatial resolution of our vision outside the point of fixation (other factors are photoreceptor density and optical aberrations). Only a few cortical receptive fields resemble the structure of thalamic receptive fields, while others have elongated subregions that respond to either dark or light spots, others respond similarly to light and dark spots through the entire receptive field and others do not respond to spots at all.

3.2 Receptive field neuron response

The neurons in the receptive or sensory layer generate responses defined by the(5).

$$R_{RF} = \sum I_{ij} * W_{ij} \quad (5)$$

The matrix W_{ij} defines a receptive field of the neuron, where i is X axis resolution and j is Y axis resolution. The field can be off-centered or on-centered as it is shown

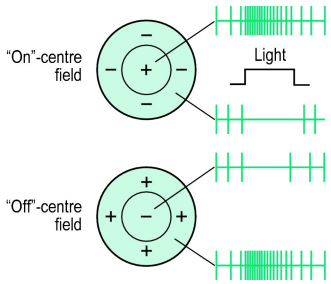

Fig. 3. Off-centered and on-centered neural receptive field and corresponding spike trains. *Source: Millodot: Dictionary of Optometry and Visual Science, 7th edition. 2009 Butterworth-Heinemann*

in Fig. 3, or, alternatively, it can describe a Gabor-like filter as shown in Fig. 4. These RFs can be used as line detectors, small circle detectors, performing feature extraction for higher layers. Simple classification tasks (i.e. the inclination of the line, circle or non-circle object) can be performed by single-layer receptive field neurons. Having normalized input and weights, the maximum excitation will be achieved when the input exactly matches the weight matrix.

Sensory layer neurons generate spikes at a frequency proportional to their excitation. As the neuron firing frequency can not be infinite, the maximum firing rate is limited, and thus, the membrane potential is normalized. The spiking response firing rate (FR_n) is described by (6), where $RPmax$ is the defined minimum refractory period and $max(R)$ is the maximum possible value of membrane potential.

$$FR_n = \frac{1}{RPmax * \frac{R_n}{max(R)}} \tag{6}$$

3.3 Gabor filters

Gabor filter took the name after Dennis Gabor defined a band-pass filter widely used in digital signal processing in its time domain form, and image analysis in its space domain form. A family of two-dimensional Gabor functions was proposed by Daugman (Daugman, 1985) as the mathematical models of simple cells receptive fields in visual cortex. These functions are described by (7).

$$\begin{aligned} g_{\lambda,\theta,\phi,\sigma,\gamma}(x,y) = exp(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2})cos(2\pi\frac{x'}{\lambda}+\phi) \\ x' = xcos\theta + ysin\theta \\ y' = -xsin\theta + ycos\theta \end{aligned} \tag{7}$$

The shape and orientation of such receptive field can be selected by modifying the parameters of (7). Standard deviation σ determines the linear size of the receptive field. The value of γ determines the spatial aspect ratio, i.e. the ellipticity of the RF. Value λ is the wavelength and thus, by modifying the σ/λ ratio, the number of inhibitory and excitatory zones in the RF can be modified. The ϕ value responds to the symmetry of the RF and θ specifies the orientation of the RF.

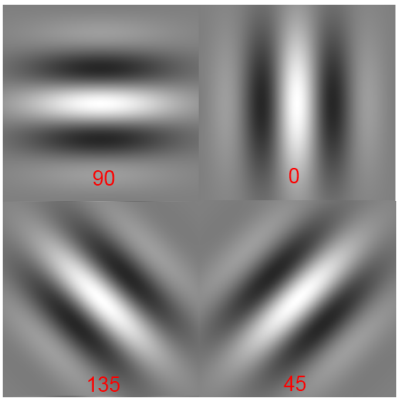

Fig. 4. Different types of Gabor RF.

Gabor fields have a good orientation selection properties, and are widely used for image decomposition. For example, the image of a car in Fig. 5 is converted with four Gabor fields with orientations of 45°, 135°, 0°, and 90°. It is clearly seen in Fig. 6 that the lines in the direction corresponding to the receptive field are promoted, while other lines are depressed.

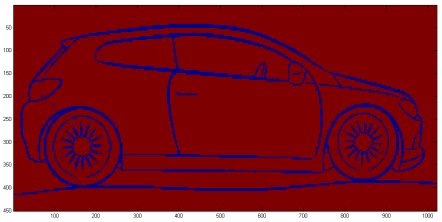

Fig. 5. Sample image used for testing Gabor fields.

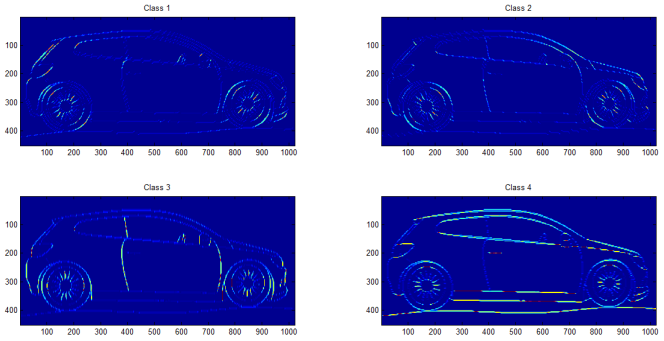

Fig. 6. Sample image converted with different receptive fields according to four orientation values.

4. ADDRESS-EVENT REPRESENTATION

The encoded data should be represented in format convenient to use in spiking neural networks. The format should have maximum data density, be easily encoded, decoded and routed in real time. Data representation inside the network should be more or less uniform to allow scalability and modularity of the network. A special protocol and interface, called Address-Event Representation (Mahowald, 1992) become widespread for this task. Massive parallel designs, like spiking neural networks suffer from so-called

'connectionist problem': the amount of connections is several times in magnitude greater than the number of processing elements, but the number of simultaneously active inputs is relatively low. In such situation, a significant part of the design can be serialized for a small decrease of network throughput. Actually, in real designs, due to technology restrictions, serial processing can be even faster than parallel.

In the Address-Event representation (AER), wide and sparse spike trains are serialized: each spike becomes replaced with a serial number of synapse firing. Then, these numbers are put into a one-dimensional vector ordered by time of spike occurrence. In case of two spikes (also called events) with the same time the order is either random or by number of synapse generating the spike. Thus, zero values in the spike train are not transmitted, only spikes are encoded. Additional signals in the protocol are request-acknowledge signals for flow control. Such protocol is very simple and can be implemented in very limited hardware, including mixed-signal specific neuromorphic chips (Moradi and Indiveri, 2013). On the receiver side the stream of event coordinates is routed to the synapses according to the received data.

Address-Event Representation became a mainstream concept of frameless vision systems (Perez-Carrasco et al., 2013; Kasabov and K. Dhoblea, 2013). Recent neuromorphic hardware projects (CAVIAR (Serrano-Gotarredona et al., 2009), SCANDLE) are using AER chips for processing, existing sensors with AER output as in (Perez-Carrasco et al., 2013) and (Chan et al., 2006). However, data must be AER-encoded before use. With AER bus the tasks of pooling or network pruning do not exist until the neurons are not removed from the network and the architecture always remains the same. High-level AER protocol, with handshaking became a de-facto standard for neuromorphic hardware in the last years. The described encoding algorithm produce data output compartible with the existing AER software and hardware, allowing the software generated spike stream to be directly injected into a hardware or software SNN. Data conversion to AER in real time is one of the key points in the real-world applications of spiking neural networks.

5. SOFTWARE EMULATION OF VISUAL NEURONS

The neuron-layer simulation is consuming lots of computational resources, thus, the encoding algorithm for AER was modified. The encoding is done in four steps: input image is normalized, then the selective receptive field is applied pixel-wise to the image. The resulting matrix contains the neuron firing frequencies (the higher is the image intensity, the faster the neuron is firing), and based on this matrix, a parallel spike train is generated. After the generation, the spike train is converted to the AER stream. In case of very large spike streams with thousands of neurons and thousands of time units (TU), the spike train can be split into smaller chunks and processed piecewise.

For the demonstration purposes and for clarity, a set of 10x10 black and white images representing the alphabet was used. The output spike train has 100 neuron tracks. The counting window is set to be 30 TU, limiting the maximum spiking period. Each sample is presented during 100 TU. Orientation decomposition is presented in Fig. 7 and generated spike trains are shown in Fig. 8.

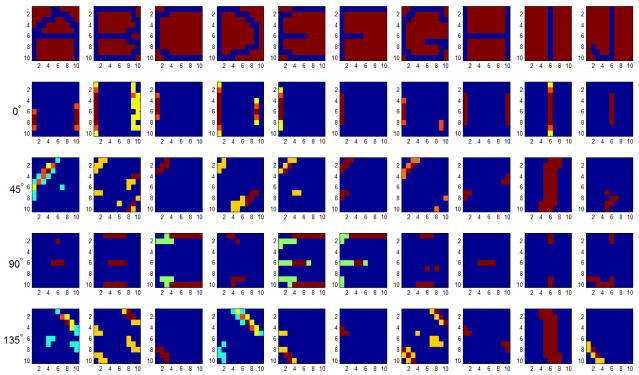

Fig. 7. Sample image set, converted with oriented Gabor receptive fields.

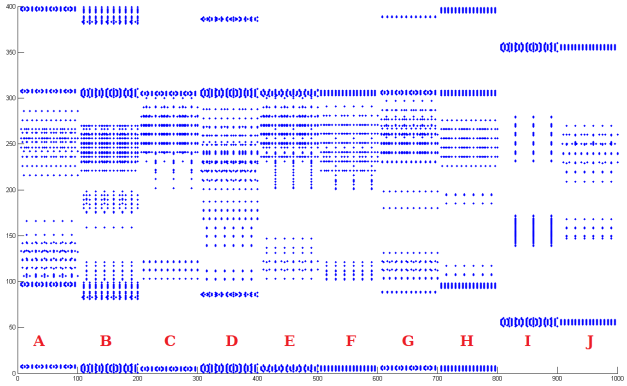

Fig. 8. Spike train, generated from sample image set

The generated spike train is then converted into AER stream, ready for use in the external device or AER-processing software as jAER (jAER software, 2012). This software supports a number of hardware AER devices and can process synthetically generated AER data which are split into 3-byte words, where first byte contains sensor or receptive field identifier, second byte is X coordinate of the event and third byte is Y coordinate. Such format allows up to 256 sensors or different receptive fields encoding of up to 256x256 pixels area. The data density is the highest possible with each word encoding one event with no empty events. Assigning one sensor ID to the timer, a timestamped AER stream is obtained.

6. ENCODING PERFORMANCE

Tests were performed on Intel i7-2632 processor on Matlab 2012 software in 32-bit environment. Images are black and white contours (letters, car2) or 8-bit grayscale (Car1, Lenna). Letters were encoded using four 5x5 orientation-selective fields, other samples were using 9x9 orientation-selective fields. The resulting data are shown in Table 6. AER Data are encoded as 16-bit integer vector and spike trains are stored as logic variables. As modern processors can not perform memory operations on data less than machine word (32 bit for 32-bit architectures), the redundancy of operations on spike train data is very large. Thus the AER representation is saving not only

memory occupation, but also processor time. In addition, AER data do not require wide data bus for transmission, which is also beneficial for reducing computations.

Table 1. AER encoding speed and size of data

	Image Size	Enc. time	AER Size	Train size
10 letters	10x10x10	0.105292	3632	400000
Car1	240x320	4.482844	252405	30720000
Lenna	512x512	15.095	826938	104857600
Car2	451x1051	24.74	1623478	184188400

7. CONCLUSIONS

In this paper we present an efficient frequency encoding method of image data using receptive fields. The algorithm can be easily paralellized and implemented in hardware, making it suitable for usage in embedded neuromorphic vision systems where computational burden is reduced. In addition, the proposed encoding based on receptive field image decomposition can work with spike data on input as well as with other stimuli. Proposed decomposition of visual data to simple forms and lines can serve as input convolution stage for Deep Belief Networks or Restricted Boltzmann Machines.

REFERENCES

F. Alnajjar and K. Murase. Sensor-fusion in spiking neural network that generates autonomous behavior in real mobile robot. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2200–2206, 2008. doi: 10.1109/IJCNN.2008.4634102.

O. Booij. Temporal pattern classification using spiking neural networks. Master's thesis, Intelligent Systems Laboratory (ISLA), University of Amsterdam, August 2004. Available from http://www.science.uva.nl/~obooij.

J. Botzheim, T. Obo, and N. Kubota. Human gesture recognition for robot partners by spiking neural network and classification learning. In *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on*, pages 1954–1958, 2012. doi: 10.1109/SCIS-ISIS.2012.6505305.

A. Bouganis and M. Shanahan. Training a spiking neural network to control a 4-dof robotic arm based on spike timing-dependent plasticity. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8, july 2010. doi: 10.1109/IJCNN.2010.5596525.

V. Chan, A. van Schaik, and Shih-Chii Liu. Spike response properties of an aer ear. In *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pages 859–862, 2006. doi: 10.1109/ISCAS.2006.1692721.

J. G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 2(7):1160–1169, 1985.

Huijuan Fang, Yongji Wang, and Jiping He. Spiking neural networks for cortical neuronal spike train decoding. *Neural Computation*, 22(4):1060–1085, Nov 2009. ISSN 0899-7667. doi: 10.1162/neco.2009.10-08-885.

Wulfram Gerstner and Werner M Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002. URL `http://www.loc.gov/catdir/samples/cam031`.

jAER software. Java tools for address-event representation (aer) neuromorphic processing. *SourceForge*, 2012. URL `http://sourceforge.net/projects/jaer/`.

N. Kasabov and G. Indiveri K. Dhoblea, N. Nuntalida. Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition. *Neural Networks*, 41:188–201, 2013. doi: 10.1016/j.neunet.2012.11.014.

M. A. Mahowald. *VLSI analogs of neuronal visual processing: a synthesis of form and function*. PhD thesis, Department of Computation and Neural Systems, California Institute of Technology, Pasadena, CA., 1992.

Luis M Martinez and Jose-Manuel Alonso. Complex receptive fields in primary visual cortex. *The Neuroscientist: a review journal bringing neurobiology, neurology and psychiatry*, 9(5):317–331, Oct 2003. ISSN 1073-8584. PMID: 14580117.

S. Moradi and G. Indiveri. An event-based neural network architecture with an asynchronous programmable synaptic memory. *IEEE Transactions on Biomedical Circuits and Systems*, pages 1–10, March 2013. doi: 10.1109/TBCAS.2013.2255873.

Helene Paugam-Moisy and Sander M. Bohte. *Handbook of Natural Computing*, chapter Computing with Spiking Neuron Networks. Springer-Verlag, sep 2009. URL `http://liris.cnrs.fr/publis/?id=4305`.

J.A. Perez-Carrasco, B. Acha, C. Serrano, L. Camunas-Mesa, T. Serrano-Gotarredona, and B. Linares-Barranco. Fast vision through frameless event-based sensing and convolutional processing: Application to texture recognition. *Neural Networks, IEEE Transactions on*, 21(4):609 –620, april 2010. ISSN 1045-9227. doi: 10.1109/TNN.2009.2039943.

J.A. Perez-Carrasco, Bo Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, Shouchun Chen, and B. Linares-Barranco. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2706–2719, 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.71.

D.T. Pham, M.S. Packianather, and E. Y A Charles. A self-organising spiking neural network trained using delay adaptation. In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, pages 3441–3446, 2007. doi: 10.1109/ISIE.2007.4375170.

Sivalogeswaran Ratnasingam and T.M. McGinnity. A spiking neural network for tactile form based object recognition. In *The 2011 International Joint Conference on Neural Networks (IJCNN)*, pages 880–885, 2011. doi: 10.1109/IJCNN.2011.6033314.

R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, Shih-Chii Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A.C. Ballcels, T. Serrano-Gotarredona, A.J. Acosta-Jimenez, and B. Linares-Barranco. Caviar: A 45k neuron, 5m synapse, 12g connects/s aer hardware sensory

processing learning actuating system for high-speed visual object recognition and tracking. *Neural Networks, IEEE Transactions on*, 20(9):1417–1438, 2009. ISSN 1045-9227. doi: 10.1109/TNN.2009.2023653.