NEURAL CODING STRATEGIES FOR EVENT-BASED VISION DATA

Shane Harrigan [⋆] *Sonya Coleman* [⋆] *Dermot Kerr* [⋆]
Pratheepan Yogarajah[⋆] *Zheng Fang*[†] *Chengdong Wu*[†]

[⋆] Intelligent Systems Research Centre, Ulster University, Northern Ireland, United Kingdom, BT48 7JL.
[†] Faculty of Robot Science and Engineering, Northeastern University, China, 110169.

ABSTRACT

Neural coding schemes are powerful tools used within neuroscience. This paper introduces three different neural coding scheme formations for event-based vision data which are designed to emulate the neural behaviour exhibited by neurons under stimuli. Presented are phase-of-firing and two sparse neural coding schemes. It is determined that machine learning approaches, i.e. Convolutional Neural Network combined with a Stacked Autoencoder network, produce powerful descriptors of the patterns within events. These coding schemes are deployed in an existing action recognition template and evaluated using two popular event-based data sets.

Index Terms— event-based vision, convolutional neural network, encoding scheme, feature extraction, object recognition

1. INTRODUCTION

Event-based vision pixels, such as the Dynamic Vision Sensor (DVS) pixel, are biologically inspired sensors designed to emulate the behaviour of the neural systems of the retina [1]. An event-based vision pixel emulates the retinal signal behaviour by producing an asynchronous spike, or *event*, when a change of luminance which is detected by sensor goes beyond an adaptive threshold maintained at the hardware level. Each event $e_i \in \langle t, l, p \rangle$ where $i = 0, \ldots, n$, t is a microsecond timestamp indicating when the change of luminance was detected, $l = (\langle x, y \rangle)$, where x, y correspond to pixel coordinates within a 2-D pixel array and p is a binary set $\{1, -1\}$ indicating whether the change of luminance was positive (increasing) or negative (decreasing). The processing of a stream of event-based vision data $S = \{e_0, e_1, \ldots, e_{nlast}\}$ is a relatively new domain within the computer vision field and, as such, lacks many of the established techniques, algorithms and tools which are readily available within conventional frame-based computer vision such as corner and edge detection. The datatype presented by event-based vision sensors requires a step-change from conventional computer vision processing paradigms.

This paper builds on existing work [2, 3], which introduced a template matching system for action classification known as Post-stimulus Time-dependent Event Descriptor (P-TED). This approach consists of two histograms called motion- and pattern-histogram. The motion histogram represents the underlying observed motion within event data while the pattern histogram is a version of the Post-Stimulus-Time Histogram (PSTH) neural coding scheme representing the affect a temporal stimulus has on the excitation of the event-based vision pixels (also known as the firing pattern). This paper extends on the above work by replacing the P-TED pattern histogram with alternative neural coding schemes representing the underlying event data. The presented work is compared against an existing work using the event-based version of the popular MNIST handwriting dataset which is known as MNIST-DVS and MNIST-FLASH-DVS [4].

Current approaches to the processing of event-based data have mainly focussed on adapting event-based data to image frames in order to work with conventional frame-based techniques [5] and/or the use of machine learning in order to handle the uniqueness of this data type [6, 7, 8, 9, 10]. The adaption of event-based data for frame-based technique requires the provision of a memory infrastructure which allows for access to events which have occurred previously; this memory infrastructure often takes the form of a time-surface [11] or Surface of Active Events (SAE) [12] which are 2-D lattices which record certain historical properties of events and can produce an energy (or *heat*) map. The 2-D lattice subsequently allows frame-based techniques to be used to process event-based data in a conventional frame-based manner. The use of neural coding strategies for event-based processing [13, 14] has received less attention than machine learning and clustering approaches [15, 16, 17] yet neural coding can be directly related to the biological structures on which event-based vision is based [6].

2. EVENT DATA FRAME REPRESENTATION

Conventional frame-based techniques do not work directly with event-based data because of the sparse and asynchronous nature of event-based data produced by vision sensors such as the DVS (Section 1). In this paper, the work presented in [18] is used a means of representing event data as a frame for the neural coding schemes discussed in Section 3. Event data is converted into frame-like representations with four channels. Each cell within the frame-like lattice corresponds to a real-world event-based vision pixel from the device sensor array. The first two channels are used to contain the number of positive and negative events emitted from each event-based vision pixel which can be used to represent the spatial characteristics within the event data. The last two channels are used to contain the temporal ratio $Z_{i,j} = \frac{t_{i,j} - t_{first}}{t_{last} - t_{first}}$ where $t_{i,j}$ is equal to the timestamp of last event occurring at pixel (i, j), t_{first} is the timestamp t of the first event e_0 within the event data and t_{last} is the timestamp t of the last event e_{last} within the event data. The ratios are used to estimate the lifetime of events [19].

3. NEURAL CODING SCHEMES

Neural coding, within the context of the work presented here, refers to the characterisation of the relationship between a stimulus and neuronal responses (individual or ensemble). As both digital and analog information can be encoded by neurons [20], most neural coding strategies should be capable of encoding the artificial neural responses represented within event-based data as shown in [21]. The work presented in [2, 3] focussed on the use of the PSTH neural coding scheme operating over the stream of event data S and presented a framework for action recognition known as P-TED. This paper explores the use of sparse [22] and phase-of-firing [23] coding schemes.

We first define the function $SUM()$ which is used for summing events within a temporal range as:

$$SUM(S, t_0, t_1) = |H(S, t_0, t_1)| \tag{1}$$

where the $SUM()$ function returns the number of events between the temporal values of t_0 and t_1 such that $t_0 \leq e_t \leq t_1$ within H where $S \supset H(S, t_0, t_1) = \{e_i | t_0 \leq e_i \leq t_1\}$.

3.1. Phase-of-Firing Coding

The phase-of-firing coding scheme is a temporal coding scheme which counts the number of events in a temporal window where the time references are based on oscillations by mapping an event to the phase of a high frequency oscillation (Figure 1B) [24]. This type of encoding represents the firing pattern characteristic of the event sensors during specific periods of an oscillating amplitude A. Oscillations are segmented into four partitions, allowing for four discrete values to be obtained per oscillation (Figure 1D). In the work presented here oscillations are simulated using a waveform function characterised with a temporal increase from 0 to +1 followed by an immediate reset (Figure 1C). Equation 2 shows the partitioning function $B(A)$ for A to obtain the four values for each partitions of the waveform.

$$B(A) = \begin{cases} 0 & 0 \leq |A| \leq 0.25 \\ 1 & 0.25 < |A| \leq 0.50 \\ 2 & 0.50 < |A| \leq 0.75 \\ 3 & 0.75 < |A| < 1 \end{cases} \tag{2}$$

where $Domain(B) = \{A \in \mathbb{R} | 0 \leq A \leq 1\}$. In the case of $A = 1$, A it subsequently becomes 0, indicating the beginning of the next waveform.

Each waveform produces a feature vector P (Equation 3) containing a value for each partition P_B of the waveform,

$$P = P_B = SUM(S, t_{start}, t_{finish}) \tag{3}$$

where the SUM function (Equation 1) returns the number of events between the temporal values of t_{start} and t_{finish}. Here, t_{start} is the starting time value of the current partition and t_{finish} is the finishing time value of the current partition determined by B. All P vectors are joined together to produce the final output of the phase-of-firing coding scheme (Equation 4).

$$Coding_{phase} = \|_{i=1}^{N} P_i = P_1 \parallel \ldots \parallel P_N \tag{4}$$

where N is the total number of waveforms produced.

The phase-of-firing coding scheme, in this work, makes use of the spike count at specific time windows which are controlled by reference to A. An added benefit of using the phase-of-firing coding scheme is that events produced by event-based vision pixels within a group are likely to occur at the same period of the oscillation and have a strong probability of being encoded together by this scheme. Here, the phase-of-firing coding scheme is referred to as P-TED(POF).

3.2. Sparse Coding

As the term suggests, sparse coding produces a sparse representation of data compared with the original data series on which it is based. Sparse autoencoders [25] are a popular form of sparse coding. Sparse autoencoders are a type of artificial neural network that learn a function $Q(S) \approx S$ such that the number of output layer neurons is equal to the number of input layer neurons. The autoencoder within the presented work accepts the input of events coded using the phase-of-firing scheme (Section 3.1) and uses a sigmoidal activation function. Figure 3 illustrates an autoencoder network using the phase-of-firing encoding (Section 3.1) as input to the network P-TED(SPOF).

A deep-learning approach which combines a convolutional neural network (CNN) with a stacked autoencoder (SAE) architecture has been presented in [26] and this is presented as P-TED(CNN). CNNs are neural networks consisting of multiple layers with several convolution-pooling layer pairs. CNNs are usually designed to recognize shapes

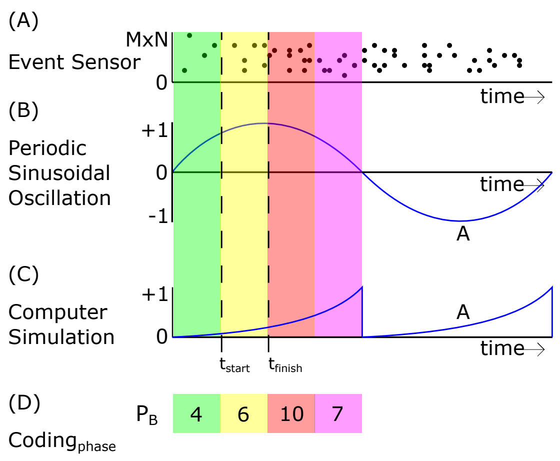

Fig. 1: An illustration of a periodic 2 Hz sinusoidal oscillation (B) overlaying event data (A) each dot representing an event triggered by a sensor on an array ranging from 0 to $M \times N$ (where M and N are the width and height of the device array respectfully) and the computer simulation of the oscillation (C) used to form the $Coding_{phase}$ histogram (D). Example t_{start} and t_{finish} are also shown.

within images and are known to be location invariant to an extent [27]. CNNs usually involve the convolution of several 2-D filters with an input image at the network convolutional layer and subsampling the output to a smaller size in the pooling layer. The back-propagation algorithm is normally used to learn the weights and filters in order to decrease the overall classification error. An SAE is a neural network consisting of many layers of autoencoders. SAE networks consist of an input layer, several unsupervised autoencoders and an output layer. Each autoencoder is trained separately and the output of one autoencoder within the hidden layer is used as input for the next layer. P-TED(SPOF) and P-TED(CNN) replace the P-TED pattern histogram within the P-TED framework. Figure 2 illustrates the architecture deployed in P-TED(CNN). The P-TED(CNN) CNN accepts $128 \times 128 \times 4$ event-frame (Section 2) inputs which are passed to four convolutional layers operating a 3×3 filter with a step of 2 where the channels for the four convolutional layers is 64, 128, 256 and 512 (as illustrated in Figure 2).

The P-TED(CNN) made use of the rectified linear unit (ReLU) function as the output function of the P-TED(CNN) CNN layers. The output of the CNN layer is given as input to two ResBlock layers, both with 3×3 filtering and a stride of 1. The number of channels within the ResBlock layers is 512. ResBlock was used to combat the vanishing feature problem within CNN architectures because of the high sensitivity of event data. The output of the ResBlock layers is input into a fully connected layer with 1024 nodes. The output of the fully connected layer is input into the auto-encoder structure attached at the end. The SAE portion output uses the midpoint of the final SAE layer (where the feature compression is at its maximum) with the softmax function for classification. For the CNN and the autoencoder, the cross entropy loss function and the Adam optimiser [28] were used in training.

4. EXPERIMENTS

For each coding scheme (Section 3) a version of the P-TED framework presented in [3] is generated with the pattern feature histogram ([3], Section 2) formed (or replaced in the case of the machine learning approaches) in using the coding schemes presented in this paper. Each version of the P-TED framework is evaluated using two datasets discussed in Section 4.1 and results are presented in Section 4.2.

4.1. Datasets

The MNIST-DVS and MNIST-FLASH-DVS [4] are event-based datasets based on the popular MNIST handwriting dataset. MNIST-DVS and MNIST-FLASH-DVS contains 30,000 samples consisting of 10 classes representing the digit range $0-9$. Each sample within MNIST-DVS and MNIST-FLASH-DVS contains the event-based response to the corresponding handwriting sample. Each response was captured by simulating programmatic motion (or in the case of MNIST-FLASH-DVS the digit was flashed five times per sample from a static position); the handwritten digit was displayed on a LCD screen for 2-3s with a high contrast and a controlled refresh rate. The high contrast is used to minimise refresh artifacts (as refreshing causes changes in luminance therefore triggering events). MNIST-DVS presents a unique challenges for recognition as event responses are continuous. The first recorded sample of each of the 10 classes within the dataset is selected to form the template which represents that class. In the case of P-TED(POF) and P-TED(SPOF) a 128Hz oscillation is used for the phase-of-firing coding scheme (Section 3.1), as previous research has shown that high oscillation frequency plays a significant role in visual communication [24]. For each 2 second MNIST-DVS and MNIST-FLASH-DVS sample, 1024 values are produced under the phase-of-firing coding scheme. For the P-TED(CNN) the datasets are split for training and testing in an 75% / 25% ratio.

4.2. Results

Results obtained are presented in Table 1. Each row shows the percentage accuracy obtained when running the P-TED framework using the corresponding coding scheme for the pattern histogram (Section 3). The machine learning coding schemes of P-TED(CNN) and P-TED(SPOF) (Section 3.2) outperform the other coding schemes, likely due to the ability of the architectures to extract temporal features and learn to

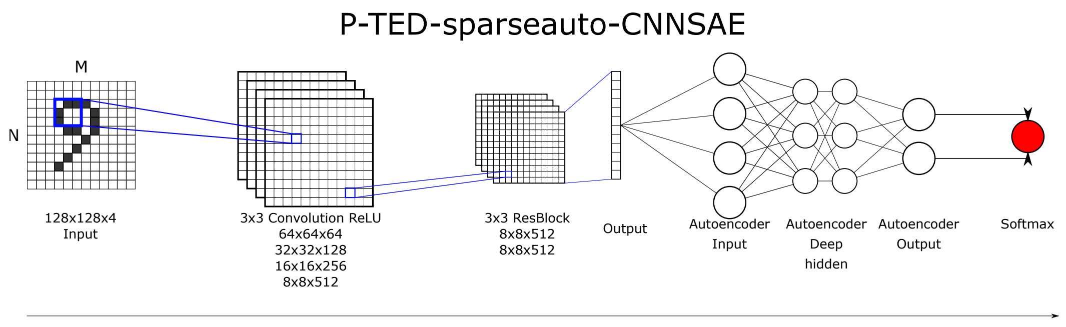

Fig. 2: An illustration of the P-TED(CNN) architecture using a $M \times N$ event-frame (where M and N are the width and height of the device array respectfully).

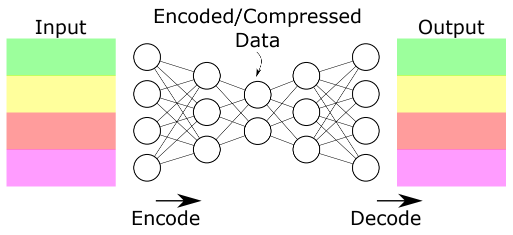

Fig. 3: An illustration of phase-of-firing encoded data inputted into the autoencoder neural network.

Table 1: Recognition accuracy using each type of neural pattern coding scheme over the MNIST-DVS and MNIST-FLASH-DVS datasets.

Coding Scheme	Data Accuracy	
	MNIST-DVS(%)	MNIST-FLASH-DVS(%)
P-TED	91.73	94.80
P-TED(POF)	91.19	96.28
P-TED(SPOF)	93.77	94.11
P-TED(CNN)	97.32	98.51

represent these efficiently. The P-TED(POF) coding scheme (Section 3.1) was the best performing in terms of accuracy after the machine learning techniques (and ranked second in the MNIST-FLASH-DVS portion). The original coding scheme used within [3] was the second best performing in terms of accuracy after the machine learning techniques (and ranked third in the MNIST-FLASH-DVS portion).

5. CONCLUSIONS

This paper presents the use of three different neural coding schemes within event-based template matching framework first presented in [3] and compared with the original coding scheme used. The neural coding schemes presented here are phase-of-firing coding (Section 3.1) and two sparse coding (Section 3.2) schemes. Coding schemes are evaluated using the MNIST-DVS and MNIST-FLASH-DVS datasets and results demonstrate that coding schemes which are constructed using machine learning techniques perform best followed closely by phase-of-firing coding. Future work will involve exploration of stacked autoencoders, active learning and adapting other frameworks to work with the coding schemes.

6. REFERENCES

[1] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 x 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[2] S. Harrigan, S. Coleman, D. Kerr, and P. Yogarajah, "From retina understanding to neuromorphic vision," in *IEEE world congress on computational intelligence*, Rio de Janeiro, Brazil, 2018.

[3] S. Harrigan, D. Kerr, S.Coleman, P. Yogarajah, Z. Fang, and C. Wu, "Neuromorphic event-based space-time template action recognition," 8 2019, Irish Machine Vision and Image Processing Conference, IMVIP ; Conference date: 28-08-2019 Through 31-08-2019.

[4] T. Serrano-Gotarredona and B. Linares-Barranco, "Poker-dvs and mnist-dvs. their history, how they were made, and other details," *Frontiers in neuroscience*, vol. 9, pp. 481, 2015.

[5] E. Mueggler, C. Bartolozzi, and D. Scaramuzza, "Fast Event-based Corner Detection," in *British Machine Vis. Conf. (BMVC)*, 2017, vol. 1, pp. 1–11.

[6] B. Ramesh, H. Yang, G. Orchard, N. Anh L. Thi, and C. Xiang, "Dart: distribution aware retinal transform for event-based cameras," *arXiv preprint arXiv:1710.10800*, 2017.

[7] J. Manderscheid, A. Sironi, N. Bourdis, D. Migliore, and V. Lepetit, "Speed invariant time surface for learning to detect corner points with event-based cameras,"

in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10245–10254.

[8] S. Mohapatra, H. Gotzig, S. Yogamani, S. Milz, and R. Zollner, "Exploring deep spiking neural networks for automated driving applications," *arXiv preprint arXiv:1903.02080*, 2019.

[9] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *arXiv preprint arXiv:1906.07165*, 2019.

[10] J. Thiele, O. Bichler, and A. Dupret, "A timescale invariant stdp-based spiking deep network for unsupervised online feature extraction from event-based sensor data," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[11] X. Lagorce, G. Orchard, F. Galluppi, B. Shi, and R. Benosman, "HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1346–1359, 2017.

[12] E. Adelson and J. Bergen, "Spatiotemporal energy models for the perception of motion," *Josa a*, vol. 2, no. 2, pp. 284–299, 1985.

[13] J. Pérez-Carrasco, B. Zhao, B. Serrano, C.and 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 and coincidence processing–application to feedforward convnets," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2706–2719, 2013.

[14] G. Haessig and R. Benosman, "A sparse coding multiscale precise-timing machine learning algorithm for neuromorphic event-based sensors," in *Micro-and Nanotechnology Sensors, Systems, and Applications X*. International Society for Optics and Photonics, 2018, vol. 10639, p. 106391U.

[15] M. Litzenberger, B. Kohn, A.N. Belbachir, N. Donath, G. Gritsch, H. Garn, C. Posch, and S. Schraml, "Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor," in *2006 IEEE intelligent transportation systems conference*. IEEE, 2006, pp. 653–658.

[16] E. Piątkowska, A.N. Belbachir, S. Schraml, and M. Gelautz, "Spatiotemporal multiple persons tracking using dynamic vision sensor," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2012, pp. 35–40.

[17] M Litzenberger, C Posch, D Bauer, AN Belbachir, P Schon, B Kohn, and H Garn, "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*. IEEE, 2006, pp. 173–178.

[18] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," *arXiv preprint arXiv:1802.06898*, 2018.

[19] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza, "Lifetime estimation of events from dynamic vision sensors," in *2015 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4874–4881.

[20] S. Thorpe, "Spike arrival times: A highly efficient coding scheme for neural networks," *Parallel processing in neural systems*, pp. 91–94, 1990.

[21] D. Moeys, F. Corradi, E. Kerr, P. Vance, G. Das, D. Neil, D. Kerr, and T. Delbruck, "Steering a predator robot using a mixed frame/event-driven convolutional neural network," in *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*. jun 2016, pp. 1–8, IEEE.

[22] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607, 1996.

[23] J. O'Keefe and M. Recce, "Phase relationship between hippocampal place units and the eeg theta rhythm," *Hippocampus*, vol. 3, no. 3, pp. 317–330, 1993.

[24] M.N Havenith, S. Yu, J. Biederlack, N.H. Chen, W. Singer, and D. Nikolić, "Synchrony makes neurons fire in sequence, and stimulus properties determine who is ahead," *Journal of neuroscience*, vol. 31, no. 23, pp. 8570–8584, 2011.

[25] C. Liou, W. Cheng, J. Liou, and D. Liou, "Autoencoder for words," *Neurocomputing*, vol. 139, pp. 84–96, 2014.

[26] Y. Tabar and U. Halici, "A novel deep learning approach for classification of eeg motor imagery signals," *Journal of neural engineering*, vol. 14, no. 1, pp. 016003, 2016.

[27] Wei Zhang, Kazuyoshi Itoh, Jun Tanida, and Yoshiki Ichioka, "Parallel distributed processing model with local space-invariant interconnections and its optical architecture," *Applied optics*, vol. 29, no. 32, pp. 4790–4797, 1990.

[28] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.