Investigation of event-based memory surfaces for high-speed tracking, unsupervised feature extraction and object recognition

Saeed Afshar1, student Member IEEE, Gregory Cohen1, student Member IEEE, Chetan Singh Thakur1, Student Member IEEE, Jonathan Tapson1, Member IEEE, Tara Julia Hamilton1, Member IEEE, André van Schaik1, Fellow IEEE.

1Biomedical Engineering and Neuroscience Program, The MARCS Institute, Western Sydney University, Australia
s.afshar@westernsydney.edu.au

Abstract— In this paper an event-based tracking, feature extraction, and classification system is presented for performing object recognition using an event-based camera. The high-speed recognition task involves detecting and classifying model airplanes that are dropped free-hand close to the camera lens so as to generate a challenging highly varied dataset of spatio-temporal event patterns. We investigate the use of time decaying memory surfaces to capture the temporal aspect of the event-based data. These surfaces are then used to perform unsupervised feature extraction, tracking and recognition. Both linear and exponentially decaying surfaces were found to result in equally high recognition accuracy. Using only twenty-five event-based feature extracting neurons in series with a linear classifier, the system achieves 98.61% recognition accuracy within 156 milliseconds of the airplane entering the field of view. By comparing the linear classifier results to a high-capacity ELM classifier, we find that a small number of event-based feature extractors can effectively project the complex spatio-temporal event patterns of the dataset to a linearly separable representation in the feature space.

I. INTRODUCTION

The last decade has seen significant development of event-based cameras. Cameras such as the Dynamic Vision Sensor (DVS) [1] and the Asynchronous Time-based Image Sensor (ATIS) [2] attempt to model the operation of the human retina by generating events at each pixel in response to changes in illumination. These cameras have spurred the development of a range of visual processing algorithms to tackle existing problems such as optical flow detection [3], scene stitching [4], tracking [5], motion analysis [6] hand gesture recognition [7], hierarchical feature recognition [8], and unsupervised visual feature extraction and learning [9][10]. Of these works [9] is most relevant to this work, where a sequence of edges and digits are presented to the DVS camera and were learnt and predicted in an unsupervised manner operating in the temporal domain functionally similar to the feature extraction layer used in this work. Here we use the ATIS camera to perform a similar unsupervised feature extraction operation in addition to tracking and classification.

II. METHODOLOGY

A. Generating the Dataset

The system presented in this paper constitutes an event-based and high-speed classification system. Existing event-based datasets have been generated under constrained conditions especially in terms of the range of target object velocities. Variance in velocity produces significant variance in the spatio-temporal event patterns generated by event-based cameras. Since the focus of this work is on the comparison of different event-based processing approaches in the presence of such variance, a new dataset was created to test the presented event-based classification algorithms under less constrained conditions.

The dataset presented in this paper contains recordings of model airplanes as they pass through the field of view of an ATIS camera. The airplanes were dropped from variable heights close to the camera as shown in Figure 1 (a) such that they rapidly passed through the field of view. The targets crossed the field of view in an average of 242 ms ± 21 ms. The four targets used were steel models of a Mig-31, an F-117, a Su-24 and a Su-35 with wingspans of 9.1, 7.5, 10.3 and 9.0 cm respectively. The airplanes all painted uniform gray as shown in Figure 1 (b).

Figure 1(a) Setup used for recording dataset. (b) Top view of the four model airplanes.

The models were dropped free-hand from a distance in the range of approximately 120 to 160 cm in height and a distance of 40 to 80 cm from the camera. The dataset, consisting of 200 drops of each airplane type, intentionally contains significant variability in terms of speed, orientation and position relative to the camera as well as variable delay before and after the drop. These variations create a more realistic dataset, as shown in Figure 2, which must be handled by the tracking and recognition algorithm. Each recording was stored in the binary format detailed in [11], maximizing compatibility with a range of neuromorphic algorithms and systems.
B. Event-based Memory Surfaces

An event from the ATIS camera can be described by:
\[ e(\mathbf{u}, t) = [\mathbf{u}, t, p] \]  
(1)

Where \( \mathbf{u} = [x, y] \) is the spatial address of the source pixel, \( p \in \{-1, 1\} \) is the event polarity and \( t \) is the absolute time at which the event occurred. This timing and polarity information allows the generation of event surfaces described by (2) and (3), where the time and polarity of new events from the camera are stored in memory in the form of \( \Sigma_e \) and \( P_e \) respectively.

\[ \Sigma_e : \mathbb{R}^2 \rightarrow \mathbb{R} \]
\[ \mathbf{u} : t \rightarrow \Sigma_e(\mathbf{u}) \]  
(2)

\[ P_e : \mathbb{R} \rightarrow \{-1, 1\} \]
\[ \mathbf{u} : p \rightarrow P_e(\mathbf{u}) \]  
(3)

In order to be useful, an event memory surface also needs to possess an appropriate temporal memory of recent events. This can be realized using a time-decaying memory surface where, after the detection of a new event, the surface at \( \mathbf{u} \) is initially set to 1 or -1 depending on event polarity. This initial value then decays toward zero as a function of time. In this work the OFF events with \( p = -1 \) were discarded and only the ON events were used. This decay can be linear or exponential as described by (4) and (5) respectively.

\[ \Lambda_e(\mathbf{u}, t) = \begin{cases} 
P_e(\mathbf{u}), (1 + \frac{\Sigma_e(\mathbf{u}) - t}{\tau_L}), & \Sigma_e(\mathbf{u}) - t \geq \tau_L \\
0, & \Sigma_e(\mathbf{u}) - t < \tau_L 
\end{cases} \]  
(4)

\[ \Gamma_e(\mathbf{u}, t) = \begin{cases} 
P_e(\mathbf{u}), e^{\frac{\Sigma_e(\mathbf{u}) - t}{\tau_L}}, & \Sigma_e(\mathbf{u}) \leq t \\
0, & \Sigma_e(\mathbf{u}) > t 
\end{cases} \]  
(5)

Figure 2. Variation in the dataset in terms of position, scale, orientation and speed. Twenty random samples from the recorded airplane drop dataset are shown demonstrating the difficulty of the recognition task. The events, white = ON, black = OFF, were recorded during a 3 ms interval. Each of the four airplane types is shown five times. Airplane class key ordered from top left to bottom right: Mig-31: {2, 3, 7, 11, 12}, F-117: {9, 15, 16, 18, 19}, Su-24: {1, 5, 8, 14, 20} and Su-55: {4, 6, 10, 13, 17}.

Figure 3. Event time-stamp profiles from one hundred randomly sampled airplane drops from the dataset recordings. The profiles, which show the time stamp of each event as a function of the event’s index, demonstrate the variable rates of event generation over time and across different airplane drop recordings. These differences are a function of the speed, size and shape of the airplanes as well as its distance from the camera.

Figure 4. Use of linearly decaying memory surfaces for event-based feature generation and recognition. Panel (a) shows the effect of a newly detected event on a memory surface which decays over time. Panel (b) shows the exponentially decaying event memory surface \( \Gamma_e(\mathbf{u}, t) \) with \( \tau_L = 3 \) ms. Panel (c) shows the linearly decaying event memory surface \( \Lambda_e(\mathbf{u}, t) \) with \( \tau_L = 8 \) ms. The linear and exponential time constants \( \tau_L \) and \( \tau_x \) were selected so as to generate similar memory surfaces for the mean event rate in the dataset \((\sim 400 \text{ events/s})\).

C. Target tracking

Before performing feature extraction and recognition on the target airplanes their location in the field of view must first be ascertained. Fortunately, in this application, the relatively high speed of the airplane in comparison to other stimuli in the environment (such as the author’s body shown on the left in Figure 5d) separates the target from the background, as many more events are generated by the faster stimulus. Simple summation of events across the rows and columns of the camera’s field of view (after normalization and thresholding as shown in Figure 5b and c) provides a reliable signal to detect the airplane’s boundary reliably, even in the presence of other (relatively slow) moving distractors. For the entire dataset the mean time interval from the first valid object boundary detection event to the last was 156.2 ms seconds with a standard deviation of 17.8 ms.
Figure 5. Screenshot from a live demonstration of the airplane drop test after 0.08 seconds. (a) Shows the system’s output label resulting from the most recent frame as well as the overall system output label (so far) given all previous classifier outputs during this drop. Panel (b) and (c) are a summation of recent events across columns and rows respectively. Due to the relatively high speed of the plane these summations, when normalized and thresholded at 0.1, can reliably be used to extract the fast moving airplane from the static background or slower moving objects. The generated target object’s boundary is shown in (d). Note that movement of the body of the author (light vertical trace on the left) as he drops the airplane is slow relative to the airplane and generates relatively few events and so does not reach even the low set detection threshold level.

D. Event-based Feature Extraction

An event-based feature extractor was used to learn the most common spatio-temporal features generated by the recordings. We used an unsupervised spike-based feature extraction algorithm developed for hardware implementation, as previously described in [12]. In this algorithm, the Synapo-dendritic Kernel Adaptation Network, a single layer of neurons with adaptive synaptic kernels and adaptive thresholds compete in the temporal domain to learn commonly observed spatio-temporal spike patterns. These adaptive synapo-dendritic kernels provide an abstracted representation of the coupling of pre- and post-synaptic neurons via multiple synaptic and dendritic pathways allowing unsupervised learning and inference of precise spike timings. In [13] the algorithm was extended using a simplified model of Spike Timing Dependent Plasticity [14] to provide synaptic encoding of afferent SNR. In [15] the algorithm was used to perform real-time unsupervised hand gesture recognition using an FPGA. In this work, the event-based approach is continued at the feature extraction layer such that the output spike of the winning neuron represents a feature event. When the camera detects a new event, a 9×9 pixel region from the event memory surface around the new event is converted to a temporally coded spatio-temporal spike pattern. The normalized real-valued intensity of the surface is first rescaled from 0-1 to 0-255 and then mapped to an 8-bit unsigned integer. This integer representation of the local surface region is then encoded into spike delays forming a spatio-temporal spike pattern. This resultant pattern is then used as the input to a 25 neuron network. The neurons were trained using 100 recordings after which the encoded features were found to have stabilized. Learning (adaptation) in the neurons was then disabled. The resultant spatio-temporal features are shown in Figure 6.

Following feature extraction and with learning disabled, the neurons compete to recognize incoming spatio-temporal event patterns generated from the same 9×9 pixel region of the event memory surface following each new event with the spike output of the winning neuron representing a feature event. These feature events were then stored onto 25 separate feature memory surfaces, which were generated identically to the event surfaces described in II B.

E. Spatial Pooling of Feature Surfaces

In order to reduce the required processing and speed up simulation, the subsystems following the feature memory surfaces were operated in a frame-based manner such that at periodic intervals the estimated target region from each feature surface was sampled to generate feature frames. The interval used for sampling was the same as the linear decay interval \( T_L = 8 \) ms. To perform spatial pooling, the estimated object boundary region was summed along the rows and columns, generating two one dimensional feature vectors (histogram of feature events), one for the rows and one for the columns. The length of these vectors would vary at each feature frame depending on the size of the estimated target region. In order to provide the classifier with a uniform input layer size, the varied length feature vectors were therefore resampled to a uniform vector size of 72, which when multiplied by the number of pooling dimensions (2), and the number of features (25), produced a 3600 input layer for the classifier. The feature extraction and feature memory surface generation and pooling operations represented the bulk of the processing time and were performed only once for each of the decay functions when collating results. Early testing showed the same features were generated through multiple
simulations of the feature extraction network and different random samplings of the dataset. This invariance may potentially be explained by the large number of event presentations in the selected recordings (approximately 5M events) and/or the relative sparsity of underlying features in the dataset.

F. Classification

In order to obtain the recognition accuracy of the system, the dataset of 800 plane recordings was repeatedly (n=100) divided into 480 training and 320 test recordings. A linear classifier and an ELM classifier [16] with 36000 hidden layer neurons (fan out of 10) were compared for classification.

III. RESULTS

To evaluate the performance of the system two measures of recognition accuracy were considered: per frame accuracy and per drop accuracy. For the per frame measure the feature vectors described section II E were presented to the classifier at periodic intervals \( \tau_L \). At each frame the class with the highest output was selected as the winner for that frame. For the per drop accuracy measure the class with the highest number of per frame wins was selected. These per frame and per drop accuracy measures are presented in Table 1 for each of the different memory surface decay functions together with simulation runtimes. All simulations were performed using MATLAB 2015a running on a DELL Alienware18 laptop.

| Actual | Predicted | F117 | Mig-31 | Su-24 | Su-35 | Accuracy |
|--------|-----------|------|--------|-------|-------|----------|
| F117   |           | 24.85| 0      | 0     | 0     | 100      |
| Mig-31 | 0         | 24.18| 0.62   | 0.32  | 96.26 |
| Su-24  | 0         | 0    | 24.79  | 0.07  | 99.72 |
| Su-35  | 0.13      | 0.11 | 0.22   | 24.73 | 98.21 |
| precision | 99.50 | 99.57 | 96.72  | 98.21 |

Table 2. Confusion matrix generated for the per drop linear classifier using linear event-based decaying surfaces. Results averaged over 100 trials.

In early testing it was found that the use of OFF events, surfaces, and features did not produce an improvement in recognition or tracking, therefore the OFF events were not used in this work and were discarded at the initial stage as shown in Figure 7.

IV. DISCUSSION

The linearly decaying memory surfaces examined in this work were shown to perform at least as well as exponentially decaying surfaces in a challenging tracking and recognition task. While using linearly decaying surfaces allows slightly faster computation it can potentially allow significantly more efficient implementations of event surfaces in neuromorphic hardware relative to exponentially decaying surfaces.

While in this work only a single unsupervised feature layer was used, by following the event-based approach the feature events output by the feature extraction layer can form the input to a subsequent hierarchy of similar event-based unsupervised feature extractors and feature surfaces resulting in a deep unsupervised event-based spiking neural network.

V. CONCLUSION

In this work we investigated the use of event-based unsupervised feature extractors together with time decaying event and feature memory surfaces for use in tracking, feature extraction, and classification. Using a dataset featuring a range of target shapes, scales, orientations and velocities we observed that both linear and exponentially decaying memory surfaces perform equally well in terms of tracking and recognition accuracy.
REFERENCES

[1] P. Lichtsteiner, C. Posch, and T. Delbruck, “A 128×128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor,” IEEE J. Solid-State Circuits, vol. 43, no. 2, pp. 566–576, 2008.

[2] C. Posch, D. Matolin, and R. Wohlgenannt, “A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS,” IEEE J. Solid-State Circuits, vol. 46, no. 1, pp. 259–275, Jan. 2011.

[3] R. Benosman, S.-H. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan, “Asynchronous frameless event-based optical flow,” Neural Netw., vol. 27, pp. 32–7, Mar. 2012.

[4] P. Klein, J. Conradt, and S.-C. Liu, “Scene stitching with event-driven sensors on a robot head platform,” in 2015 IEEE International Symposium on Circuits and Systems (ISCAS), 2015, pp. 2421–2424.

[5] X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, “Asynchronous Event-Based Multikernel Algorithm for High-Speed Visual Features Tracking,” IEEE Trans. neural networks Learn. Syst., vol. 26, no. 8, pp. 1710–20, Aug. 2015.

[6] S. Litzenberger and A. Sabo, “Can Silicon Retina Sensors be used for optical motion analysis in sports?,” Procedia Eng., vol. 34, pp. 748–753, 2012.

[7] J. H. Lee, T. Delbruck, M. Pfeiffer, P. K. J. Park, C.-W. Shin, H. E. Ryu, and B. C. Kang, “Real-time gesture interface based on event-driven processing from stereo silicon retinas,” IEEE Trans. neural networks Learn. Syst., vol. 25, no. 12, pp. 2250–63, Dec. 2014.

[8] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, and R. Benosman, “HFirst: A Temporal Approach to Object Recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 10, pp. 2028–40, Oct. 2015.

[9] X. Lagorce, S. H. Ieng, X. Clady, M. Pfeiffer, and R. B. Benosman, “Spatiotemporal Features for Asynchronous Event-based Data,” Front. Neurosci., vol. 9, no. 46, 2015.

[10] M. Giulioni, F. Corradi, V. Dante, and P. del Giudice, “Real time unsupervised learning of visual stimuli in neuromorphic VLSI systems,” Sci. Rep., vol. 5, p. 14730, Jan. 2015.

[11] G. Orchard, A. Jayawant, G. Cohen, and N. Thakor, “Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades,” TBA, pp. 1–15, 2015.

[12] S. Afshar, L. George, J. Tapson, A. van Schaik, and T. J. Hamilton, “Racing to Learn: statistical inference and learning in a single spiking neuron with adaptive kernels,” Front. Neurosci., vol. 8, p. 377, 2014.

[13] S. Afshar, L. George, C. S. Thakur, J. Tapson, A. van Schaik, P. de Chazal, and T. J. Hamilton, “Turn Down That Noise: Synaptic Encoding of Afferent SNR in a Single Spiking Neuron,” IEEE Trans. Biomed. Circuits Syst., vol. 9, no. 2, pp. 188–96, Apr. 2015.

[14] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, “Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs,” Science (80-. )., vol. 275, no. 5297, pp. 213–215, 1997.

[15] R. J. Sofatzis, S. Afshar, and T. J. Hamilton, “Rotationally invariant vision recognition with neuromorphic transformation and learning networks,” in 2014 IEEE International Symposium on Circuits and Systems (ISCAS), 2014, pp. 469–472.

[16] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: Theory and applications,” Neurocomputing, vol. 70, no. 1–3, pp. 489–501, Dec. 2006.