Bag of Events: An Efficient Probability-Based Feature Extraction Method for AER Image Sensors

Xi Peng, Bo Zhao, *Member, IEEE*, Rui Yan, *Member, IEEE*, Huajin Tang, *Member, IEEE*, and Zhang Yi, *Fellow, IEEE*

***Abstract*— Address event representation (AER) image sensors represent the visual information as a sequence of events that denotes the luminance changes of the scene. In this paper, we introduce a feature extraction method for AER image sensors based on the probability theory, namely, bag of events (BOE). The proposed approach represents each object as the joint probability distribution of the concurrent events, and each event corresponds to a unique activated pixel of the AER sensor. The advantages of BOE include: 1) it is a statistical learning method and has a good interpretability in mathematics; 2) BOE can significantly reduce the effort to tune parameters for different data sets, because it only has one hyperparameter and is robust to the value of the parameter; 3) BOE is an online learning algorithm, which does not require the training data to be collected in advance; 4) BOE can achieve competitive results in real time for feature extraction (>275 frames/s and >120 000 events/s); and 5) the implementation complexity of BOE only involves some basic operations, e.g., addition and multiplication. This guarantees the hardware friendliness of our method. The experimental results on three popular AER databases (i.e., MNIST-dynamic vision sensor, Poker Card, and Posture) show that our method is remarkably faster than two recently proposed AER categorization systems while preserving a good classification accuracy.**

***Index Terms*— Address-event representation (AER), dynamic vision sensor (DVS), events-based categorization, neuromorphic computing, online learning, statistical learning method.**

I. Introduction

NEUROMORPHIC engineering develops hardware and software to mimic the working way of neural systems. It has attracted a lot of attention from the communities of machine intelligence, neuroscience, computer vision, data mining, and electronic circuits [1]–[5].

One of most successful neuromorphic system is the asynchronous time-based image sensor [6] and event-driven dynamic vision sensor (DVS) [7], [8]. Different from the traditional camera, DVS generates output (i.e., event) only when it captures the transient in a scene instead of sending entire images at fixed frame rates. Each DVS pixel (x, y) corresponds to a local receptive field and independently senses the light change, where x and y denote the positions of the pixel. If the light changes by a given relative amount, an event (x, y, p) will be generated, where the polarity $p = 1$ denotes the increasing light (i.e., dark-to-light) and $p = -1$ denotes the decreasing light (i.e., light-to-dark). There are cases wherein multiple DVS pixels request to output events at the same time and these events will be asynchronously output with submicrosecond delays. This flow of asynchronous events is usually in the format of address event representation (AER). In the following context, an AER sensor refers to DVS unless otherwise stated.

To process the output of DVS in the computer, AER is usually represented as a collection of the quadruples (t, x, y, p) [9], where t denotes the timestamp. As an illustration, Fig. 1 shows the event flow that corresponds to a rotating object. For each stimulus onset, DVS requests to send out four events at the same time, and these events are sequentially output in a fairly random manner [10]. The delay between two consecutive events is generally larger than 1 ns but smaller than 1 μs. Moreover, for a static background and a fixed DVS, the number of events generated by an moving object that is moving parallel to the focal plane mainly depends on the moving speed of the object.

AER sensors remove the data redundancy from the scene, which has an output-by-demand nature and energy-saving advantage. However, most existing methods cannot be directly used to handle the output of the sensor. To solve this problem, some impressive works have been proposed for object recognition [11]–[19], tracking [20]–[23], and visual information processing [24]–[29]. In this paper, we mainly focus on the problem of object recognition.

Similar to the traditional image categorization system, the AER classification system (AERCsys) also consists of two parts, i.e., the feature extraction module and the classification module. The major advantages of the AERCsys include high computational efficiency, hardware friendliness, and low latency. To exploit these advantages, several recent works have been proposed, which are inspired by the huge success of deep learning. Chen *et al.* [15] proposed a bioinspired feature extraction method. Extensive theoretical analysis

Manuscript received November 14, 2014; accepted February 26, 2016. This work was supported by the National Natural Science Foundation of China under Grant 61432012. *(Corresponding authors: Rui Yan and Huajin Tang.)*

X. Peng and B. Zhao are with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore 138632 (e-mail: pangsaai@gmail.com; zhaob@i2r.a-star.edu.sg).

R. Yan, H. Tang, and Z. Yi are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: ryan@scu.edu.cn; htang@scu.edu.cn; zhangyi@scu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2016.2536741

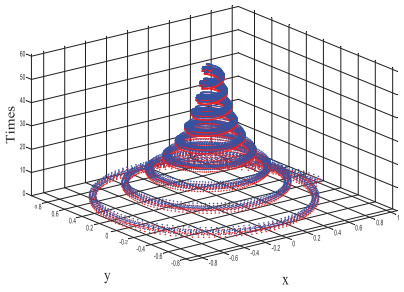
(a)

Timestamp (ns)	x	y	Polarity
192	1	2	1
282	2	3	-1
367	3	4	-1
396	2	3	1
10192	1	2	-1
...	...	...	...

event delay

stimulus delay

(b)



(c)

Fig. 1. Example to show the output of the DVS camera, where (a) and (c) are taken from [8] with permission. To generate points rotating with a controlled speed (500 Hz), one analog oscilloscope working on XY mode is used. (a) One snapshot of the input stimulus taken with a conventional camera. (b) Event flow of DVS camera which is asynchronously output with nanoseconds delay. (c) Spatio-temporal representation of the data generated by the DVS camera. Red dots: bright-to-dark events (Polarity $= -1$). Blue dots: dark-to-bright events (Polarity $= +1$).

and experimental results show that their method can extract scale- and translation-invariant features from the output of DVS. Pérez-Carrasco *et al.* [16] proposed an event-driven convolutional neural network, which achieves the pseudosimultaneity property between AER sensing and processing. In short, their method can handle the event steam very fast. O'Connor *et al.* [17] proposed a spiking deep belief network (SDBN) for feature extraction and classification. The method can perform feature extraction, information fusion, and classification at event level. Moreover, the experimental studies show that SDBN is robust to distraction, noise, scaling, translation, and rotation. Regarding the hardware implementation of SDBN, Stromatias *et al.* [30]–[32] recently conducted a series of works on the robustness of SDBN, power analysis of SpiNNaker, and a novel realization of an SDBN on the biologically inspired parallel SpiNNaker platform. Their works provide a comprehensive analysis in the scenario of hardware implementation and further promote the development of deep learning in the neuromorphic computation. Moreover, Zhao *et al.* [18] proposed another AER categorization system based on HMAX [33] and tempotron classifier [34]. Their method is also event-oriented and has achieved the state-of-the-art performance on a range of data sets.

Despite the success of these methods, it is still challenging to fully exploit the advantages of AER and design algorithms that can be easily implemented in hardware. Moreover, many existing works are based on deep learning, and few works are based on statistical and probability theory. Motivated by the works in the information theory [35] and document analysis [36], this paper proposes an online feature extraction method, named bag of events (BOE). The proposed method uses the joint probability distribution (JPD) of the consecutive events to represent each stimulus. In other words, BOE does not extract any visual features such as lines or shapes as many existing methods did. Our contributions can be summarized as follows.

1) BOE is a probability-based feature extraction method, which has the advantage of good interpretability in mathematics. Moreover, the method has only one hyperparameter and is robust to the value of the parameter, which significantly reduces the effort to tune parameters for a good performance.

TABLE I
NOTATIONS AND ABBREVIATIONS

Notation or Abbr.	Definition
n	the number of segments
m	the number of DVS pixels
k	the number of categories
e_i	the i-th event
s_j	the j-th segment
f_{ij}	The frequency of e_i within s_j
w_i	the weight factor over e_i
DVS	Dynamic Vision Sensor
AER	Address Event Representation
AERCsys	AER Classification System
SR	Segment Recorder
LIF neuron	Leaky Integrate-and-Fire neuron
HES	Hard Event Segmentation
SES	Soft Event Segmentation

2) Different from the existing deep learning based methods, BOE represents the stimulus using the JPD of multiple events instead of lines, corners, or other visual features. It only involves some basic operations with low latency, which implies the hardware friendliness of BOE.
3) BOE is an online learning algorithm, which does not require the whole training data set to be provided in advance. In other words, when the labeled (i.e., training data) and unlabeled events (i.e., testing data) are alternately received, BOE can smoothly handle the data and will not repeatedly train the feature extraction module.
4) An extensive experimental analysis shows that BOE extracts very simple, nonsymbolic features from a tiny BOE and can achieve competitive performance to existing, more sophisticated solutions.

Notations: Lowercase bold letters represent column vectors and uppercase bold ones denote matrices. $\mathbf{A}^T$ and $\mathbf{A}^{-1}$ denote the transpose and pseudoinverse of the matrix $\mathbf{A}$, respectively. Table I summarizes some mathematic notations and abbreviations used throughout this paper.

II. System Overview

Fig. 2 shows the architecture of the proposed system, which consists of three modules and two processes. The modules include an AER sensing hardware, a BOE feature extractor,

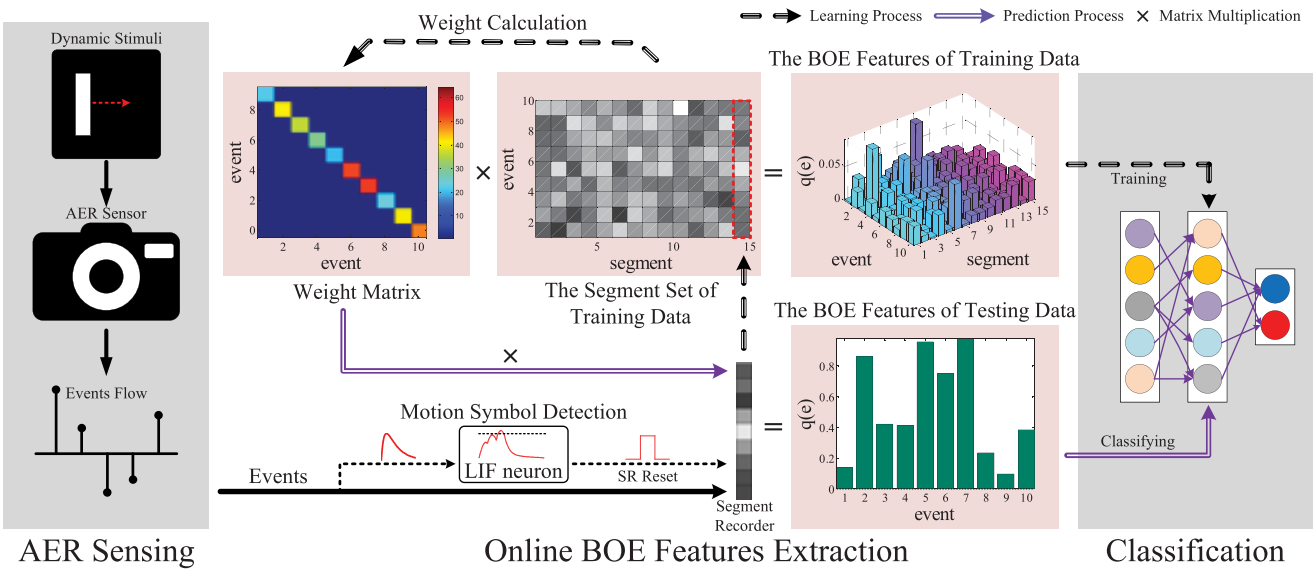


Fig. 2. Architecture of the proposed system. If the label of the coming event is known, then the learning process is adopted. Otherwise, the prediction process is adopted. This never-stop learning property is very attractive in practice.

and a classifier. The last two modules involve two processes, i.e., learning and prediction. The flow of information processing is as follows.

1) *AER Sensing:* Once the AER sensor captures the changes in a scene, a sequence of events will be output, and each event will be simultaneously sent to a motion symbol detector (MSD) and a segment recorder (SR). Different segments are caused by different stimuli (i.e., motions), and each stimulus may generate multiple events. To avoid performing feature extraction and classification all the time, we use an MSD to partition the event flow into multiple segments that are memorized into SR. At the initial state, SR can be simply regarded as a m-dimensional zero vector, where m equals to the number of DVS pixel and SR records the activated counts of DVS pixel.
2) *Motion Symbol Detection:* For computational efficiency and energy-saving, it is unnecessary to carry out feature extraction and classification all the time. In this paper, we introduce a leaky integrate-and-fire (LIF) neuron to distinct the events caused by different motions (i.e., stimuli). Each input event brings a postsynaptic potential (PSP) to this neuron. If the total potential exceeds a given threshold, the neuron will fire a spike. At that moment, learning or prediction process will be triggered.
3) *Learning Process:* If the events are caused by a labeled stimulus (i.e., training data) and the LIF neuron is fired, the system switches to the learning process, which includes the following steps: 1) append the segment in SR to the segment set of training data; 2) reset SR to the initial state; 3) calculate the weight matrix and the BOE features of the training data; and 4) train the classifier. Note that, step 1 collects the segments of labeled events to obtain the weight matrix, which seems to be memory-consuming. In hardware implementation, however, we only need to keep a vector to record the weights, and the size of vector is upper bounded by the number of DVS pixel. Therefore, our algorithm can be easily implemented in hardware. Section III-C will give more detailed analysis on this aspect.
4) *Prediction Process:* If the LIF neuron is fired, BOE will calculate the BOE feature of the current stimulus by weighting the segment vector in SR. After that, one resets the SR to the initial state and passes the BOE feature through a classifier to obtain its label. Note that the weight matrix is learned from the training data, but this step will be performed for all data, since the classification results are based on BOE features.

III. Online BOE Feature Extraction

A. Motion Symbol Detection Using an LIF Neuron

Although most AERCsys's are designed based on the event-driven nature, it is still a daunting task to explore how to use each single event as a source of meaningful information source. Thus, many works, such as the well-known pencil balancer demo [11] and the jAER software [37], accumulate the event flow into multiple segments (i.e., pseudopictures) and, then, perform feature extraction and classification based on these segments. The methods of accumulating events can be categorized into two classes, i.e., hard events segmentation (HES) and soft events segmentation (SES). HES divides the events into segments using fixed time slices (e.g., 20 ms) or fixed number of events (e.g., 200 events per segment). Different from HES, SES adaptively obtains the segments according to the statistical characteristics of the events. Thus, it is generally believed that SES is more flexible to capture the structure of the data set than HES.

In this paper, we present an SES method by introducing a single channel (i.e., synapse) LIF neuron [38]–[41]

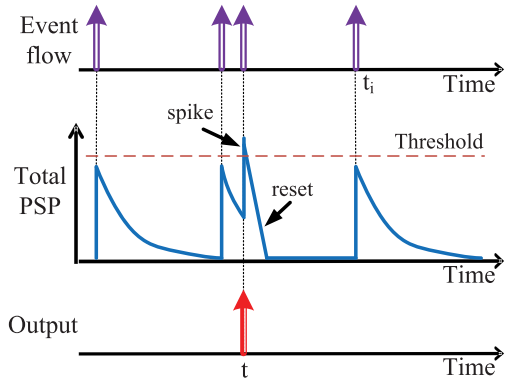


Fig. 3. Dynamics of an LIF neuron. t_i denotes the timestamp of the current event and t denotes the current spiking times of the LIF neuron.

as the MSD. As shown in Fig. 3, each input event initiates a PSP to the LIF neuron. For an input event received at time t_i, the PSP is defined as

$$\mathcal{K}(t_i) = \exp\left(-\frac{t - t_i}{\tau}\right) \tag{1}$$

where τ is the decay time constant of membrane integration. Then, the accumulated PSP within the time window $[t-1, t]$ is calculated by

$$\mathcal{K}(t) = \sum_{t_i \in [t-1,t]} \mathcal{K}(t_i) \tag{2}$$

where $t-1$ and t denote the previous and current spiking time of the neuron.

If $\mathcal{K}(t)$ is higher than a specified threshold, the neuron will be reset to 0 and a message of SR Reset will be sent out to reinitialize the SR.

B. Bag of Events

Like many existing works, the BOE algorithm divides the event streams into multiple segments. Each segment can be regarded as a bag, and the bagged events actually describe the corresponding stimulus. Note that BOE cannot be simply regarded as the process of event accumulation (i.e., bagging events). Event accumulation is widely adopted by almost all AERCsys including but not limited to [11] and [15]–[18]; however, these methods do not focus on how to use events as features to represent stimulus. In contrast, they represent each stimulus using lines, corners, shapes, and other visual features. In this paper, we propose a feature extraction method based on statistical principle, and the method does not extract any visual features. To obtain a comprehensive understanding on our algorithm, we present two different explanations. The first one intuitively shows that BOE is designed by combining the advantages of the metrics of popularity and specificity. The second one establishes the equivalence between the BOE and the expected mutual information.

Let $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ be a collection of segments and $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$ be a set of distinct events contained in $\mathcal{S}$, where n and m denote the number of segment and the number of DVS pixel, respectively. For each segment s_j,

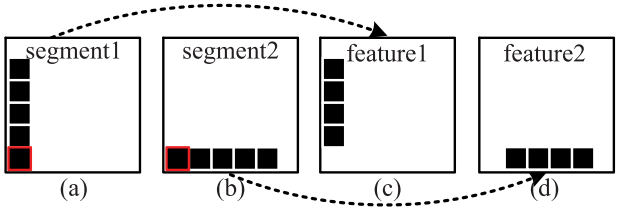


Fig. 4. Example illustration of measures of popularity and speciality. (a) and (b) Two segments output by the SR and each segment includes five different events (pixels). Red box: pixel that is frequently spiked. (c) and (d) Discriminative features for these two stimuli.

we use the JPD of $\mathcal{E}$ to represent s_j. Mathematically, $s_j = P(e_1, e_2, \ldots, e_m)$. By assuming the occurrences of the events in segments are statistically independent, then s_j can be represented as $[f_{1j}, f_{2j}, \ldots, f_{mj}]$, where f_{ij} is the frequency of e_i within s_j. We called this representation as event frequency (EF).

EF is a kind of measure of popularity, which assumes that the frequent events are important. The disadvantage of EF is that some frequent events are emphasized too much, but these events are always less discriminative [e.g., the pixels highlighted by the red rectangle in Fig. 4(a) and (b)]. Thus, EF is not good enough for classification task. As another measure, speciality allocates much more weight to the infrequent events, so that the obtained features are more discriminative [see Fig. 4(c) and (d)]. However, the measure of speciality is sensitive to the noises and outliers. Therefore, we aim to develop a method that has the advantages of the measures of popularity and speciality. We formulate the problem with

$$q_{ij} = w_i f_{ij} \tag{3}$$

where w_i and f_{ij} measure the speciality and popularity of e_i, respectively. Clearly, it is key to determine w_i, so that the popularity and the speciality are well balanced.

Let n_i be the number of the segment containing e_i. We use the self-information of e_i to weight over itself, that is

$$w_i = -\log \frac{n}{n_i}. \tag{4}$$

Self-information is derived in [35], which is used to measure the information content. By formulating the self-information into our method, the obtained result (4) depicts the speciality of the events, i.e., the infrequent events (i.e., infrequently activated DVS pixels) contain more discrimination information than the frequently occurring events. For example, suppose an event appears n times within n segments (i.e., $n_i = n$), it has a self-information measure of zero. This matches with the fact that the event is useless even harmful to the discrimination of the features.

By combining (3) and (4), BOE is defined as

$$q_{ij} = f_{ij} \log \frac{n_i}{n}. \tag{5}$$

From the above analysis, BOE combines two measures of information content, i.e., f_{ij} and w_i. The first metric f_{ij} is the estimation of the probability that the event e_i is actually observed. The second metric w_i reflects the change in the

amount of information after observing a specific event. The combination of these two measures makes BOE features discriminative.

Besides the above intuitive explanation of our method, motivated by [42], we have the following theorem, which provides another explanation toward BOE.

Theorem 1: Let $\mathscr{S}$ and $\mathscr{E}$ be the random variables defined over the space of $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ and $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$. $\mathcal{I}(\mathscr{S}; \mathscr{E})$ denotes the expected mutual information between $\mathscr{S}$ and $\mathscr{E}$ and BOE feature q_{ij} is the quantity for the calculation of $\mathcal{I}(\mathscr{S}; \mathscr{E})$, that is

$$\mathcal{I}(\mathscr{S}; \mathscr{E}) = \sum_{i=1}^{m} \sum_{j=1}^{n} q_{ij}. \tag{6}$$

Proof: Let $\mathcal{H}(\mathscr{S})$ be the marginal entropy of $\mathscr{S}$ and $\mathcal{H}(\mathscr{S}|\mathscr{E})$ be the conditional entropy of $\mathscr{S}$ for the given $\mathscr{E}$. Without loss of generality, we assume that all segments are equally likely observed, i.e., $p(s_j) = 1/n$, then we have

$$\begin{aligned} \mathcal{H}(\mathscr{S}) &= -\sum_{s_j \in \mathcal{S}} p(s_j) \log p(s_j) \\ &= -\log \frac{1}{n} \end{aligned} \tag{7}$$

and

$$\begin{aligned} \mathcal{H}(\mathscr{S}|e_i) &= -\sum_{s_j \in \mathcal{S}} p(s_j|e_i) \log p(s_j|e_i) \\ &= -\log \frac{1}{n_i} \end{aligned} \tag{8}$$

where n_i denotes the number of the segments containing e_i.

Based on (7) and (8), the expected mutual information between $\mathscr{S}$ and $\mathscr{E}$ is

$$\begin{aligned} \mathcal{I}(\mathscr{S}; \mathscr{E}) &= \mathcal{H}(\mathscr{S}) - \mathcal{H}(\mathscr{S}|\mathscr{E}) \\ &= \frac{1}{\lambda} \sum_{e_i \in \mathcal{E}} p(e_i)(\mathcal{H}(\mathscr{S}) - \mathcal{H}(\mathscr{S}|e_i)) \\ &= \frac{1}{\lambda} \sum_{e_i \in \mathcal{E}} p(e_i) \log \frac{n}{n_i} \\ &= \frac{1}{\lambda} \sum_{e_i \in \mathcal{E}} \sum_{s_j \in \mathcal{S}} f_{ij} \log \frac{n}{n_i} \end{aligned} \tag{9}$$

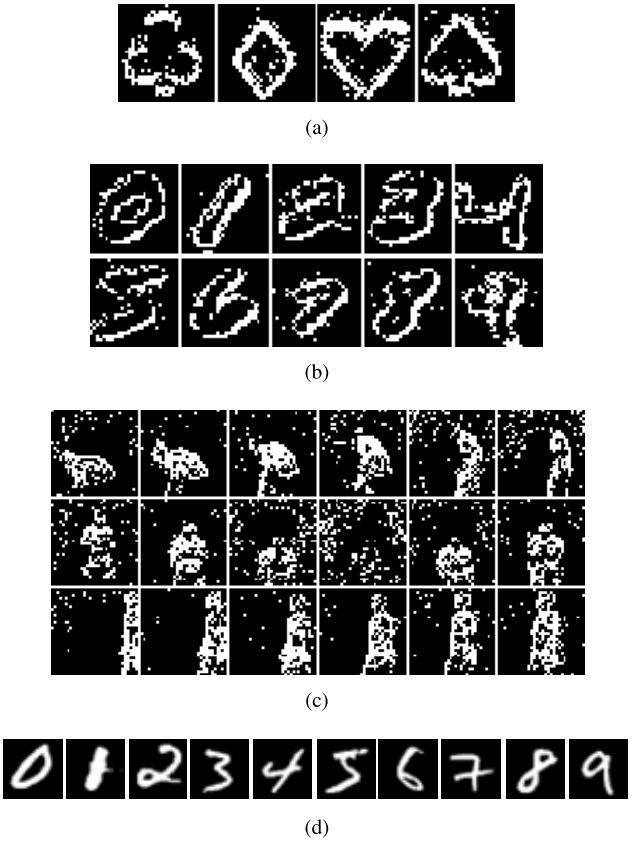
where f_{ij} is the frequency of e_i within s_j and λ is a constant factor which can be removed.

The proof is complete. □

Theorem 1 provides another way to understand BOE by bridging the connections between the BOE and the expected mutual information. According to the definition of expected mutual information, we find that the BOE actually quantizes the mutual dependence between $\mathcal{E}$ and $\mathcal{L}$, where $\mathcal{E}$ and $\mathcal{L}$ denote the set of events and labels, respectively. In other words, it measures the extent of $\mathcal{L}$'s uncertainty reduction by knowing $\mathcal{E}$, and vice versa.

C. Implementation Complexity Analysis

In this section, we investigate the complexity of the proposed feature extraction method from three stages as follows.



(a) (b) (c) (d)

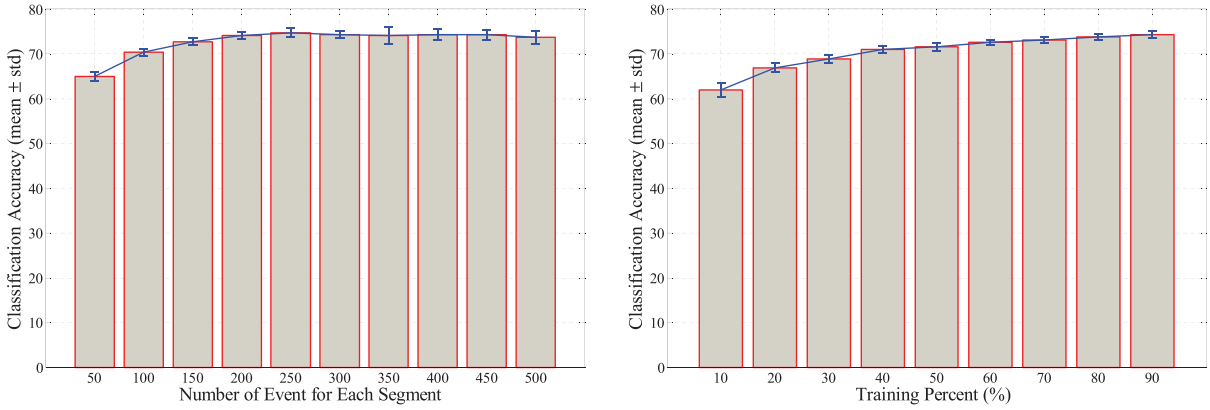
Fig. 5. Some reconstructed images from the used databases. (a) Card data set consists of four symbols, i.e., club, diamond, heart, and spade from left to right. (b) MNIST-DVS database includes ten classes, which correspond to digits 0–9. (c) Posture database includes three human actions, i.e., BEND, SITSTAND, and WALK from top to bottom. (d) Standard MINIST digital images.

1) Events Accumulation: On average, the DVS sensor sends α events to the SR, which consists of m counters (i.e., intrasegment counters), where m denotes the number of pixel addresses. This step involves α addition operations and usually $\alpha \ll m$.

2) Learning: If the events are labeled, the learning process will be triggered to update the weight matrix. More specifically, BOE will add 1 to d entries of an m-dimensional vector (i.e., intersegment counters), where these d entries correspond to d unique pixel addresses within α events. Like intrasegment counters, the number of intersegment counters also equals to the number of pixel addresses, and each counter corresponds to each address. In this step, each intersegment counter cumulatively records the number of segments that has received events at the corresponding pixel address. Moreover, the total number of training segments is also recorded, so that the weights can be scaled. Therefore, the learning process performs $d + 1$ addition operations, where $d \leq \alpha$.

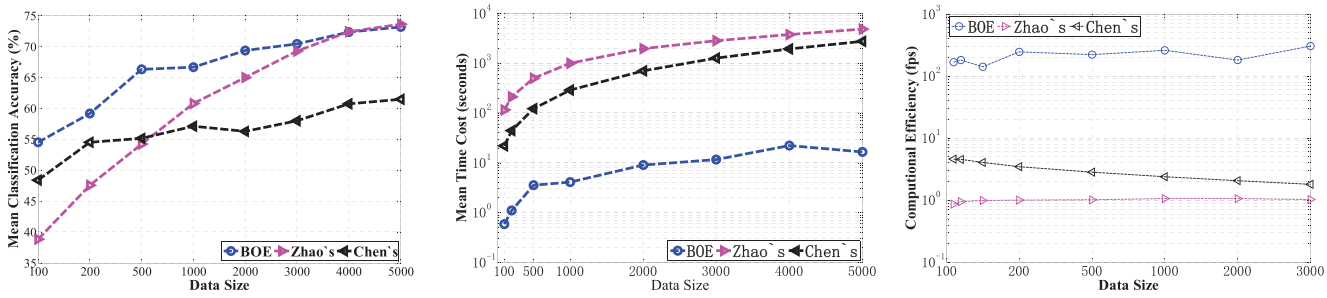
3) Prediction: To extract features from the obtained segments, two steps are required.

1) Computing the weight over each pixel address based on the records in the intersegment counters via (4). This step performs m division operations and logarithm computations. Note that BOE will only perform this step one time if no new training data are received.



(a) (b)

Fig. 6. Robustness of BOE to different hyperparameters. (a) Recognition rate on MNIST-DVS with increasing α, where 90% samples are used for training. (b) Recognition rate on MNIST-DVS with increasing training rate, where $\alpha = 300$.



(a) (b) (c)

Fig. 7. Scalability performance comparison on the MNIST-DVS database. (a) Average recognition rate. (b) Time cost for feature extraction and classification. (c) Efficiency for feature extraction and classification.

2) Extracting the feature by weighting the frequency of the current segment within d multiplication operations.

In summary, for n segments and each with α events, BOE totally performs $n(\alpha + d + 1)$ addition operations to update weights in the case of labeled events. Moreover, it also performs m division operations, m logarithm computations, and nd multiplication operations to obtain features.

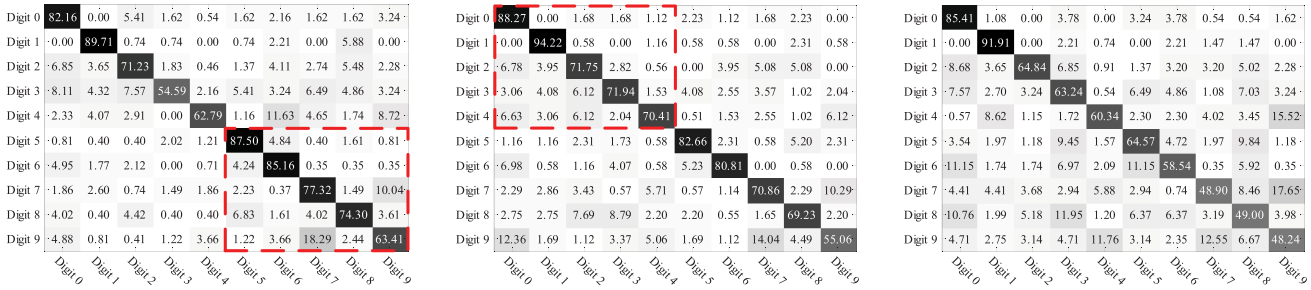
From the above analysis, our algorithm only involves some basic operations, i.e., addition, multiplication, division, and logarithm computation. The low complexity of the proposed method guarantees the hardware friendliness. About the memory requirement, our algorithm mainly needs two sets of memory to store the event frequencies (i.e., intrasegment counters) and the weights (i.e., intersegment counters). To be more accurate, BOE does not store the input events; instead, we only need to store those two sets of counters of which the intrasegment counter is like short-term memory and the intersegment counter is like long-term memory. Each of these two sets of counters has m entries. In addition, there is a number-of-training-segment counter. Thus, the memory requirement of BOE is only $2m + 1$ and can be easily implemented using the block RAM of an field-programmable gate array.

IV. Experimental Results

In this section, we investigate the performance of our method with respect to the classification accuracy and the efficiency. We also compare BOE with two recently proposed AER categorization systems on three popular AER data sets.

Benchmark Algorithms: The first method was proposed in [15], which extracts the features with a line detector from DVS output and performs classification using a nearest neighbor classifier with the Hausdorff distance [43]. The other benchmark algorithm was proposed in [18], which extracts high-level features by passing Gabor features into the HMAX model [33]. In addition, the method groups the new events using an event-driven tempotron classifier. Our categorization system uses a simple support vector machine with a linear kernel [44] as the classifier. For fair comparison, we follow the experimental setting in [15] and [18] to tune the parameters for Chen's method and Zhao's method. Moreover, we employ an HES method instead of SES to obtain the segments from the event steam by fixing the number of events within each segment. We obtain the MATLAB codes of the competing methods from the authors and carried out the experiments using MATLAB on a workstation with two Xeon E5 2.4-GHz CPUs and 32-GB RAM. The used data sets and the code of BOE are provided at the authors' Website http://machineilab.org/users/pengxi/.

Data Sets: Three DVS data sets are used in our experiments, i.e., MNIST-DVS [45], Posture [18], and Card [16], [46]. The MNIST-DVS database was generated from 10 000 original 28×28 MNIST digit images [47]. Each MNIST image was upscaled to three scales (scale-4, scale-8, and scale-16)



(a) (b) (c)

Fig. 8. Confusion table for the MNIST-DVS database. Average classification rates for individual classes are shown along the diagonal. Zhao's method achieves the best result on the first five digits and BOE outperforms Zhao's method and Chen's method on the last five digits (the best results are indicated by a red rectangle). The average accuracy of BOE, Zhao's method, and Chen's method are 75.09%, 75.35%, and 61.23%, respectively. (a) BOE. (b) Zhao's method [18]. (c) Chen's method [15].

TABLE II

COMPUTATIONAL EFFICIENCY OF DIFFERENT METHODS FOR FEATURE EXTRACTION AND CLASSIFICATION ON MNIST-DVS DATABASE. FOR ALL THE METRICS EXCEPT fps, THE VALUE IS BIGGER, AND THE PERFORMANCE IS BETTER

Algorithms	Feature Extraction					Classification				
	training(s)	testing(s)	total(s)	fps	tpe(s)	training(s)	testing(s)	total(s)	fps	tpe(s)
BOE	**27.89**	**27.28**	**55.17**	**402.65**	**8.28E-06**	**3.63**	**0.12**	**3.75**	**5926.63**	**5.62E-07**
Zhao's [18]	8601.10	955.68	9556.78	1.87	1.17E-03	204.11	26.93	231.05	77.23	2.82E-05
Chen's [15]	1208.38	134.26	1342.64	16.69	2.00E-04	-	7691.26	7691.26	2.91	1.14E-03

and was then displayed on a liquid crystal display monitor in slow motion. After that, the MNIST-DVS database was generated using a 128×128 AER sensor [45] to record the moving digit. As did in [18], the MNIST-DVS data set with scale-4 is used in our tests. Each recording has the duration of 100 ms within a resolution of 28×28. It should be pointed out that MINIST-DVS is more challenging than the standard MNIST due to the noises, blur, and other factors.

The Posture database was generated using an AER sensor to capture three human actions, i.e., bending to pick something (BEND), sitting down and standing up (SITSTAND), and walking back and forth (WALK). Each Posture image is in a scene of 32×32. The Card database is an event stream of poker card symbols with a spatial resolution of 32×32. It consists of four symbols, i.e., club, diamond, heart, and spade. Fig. 5 shows some samples of these three databases.

Besides these three DVS data sets, we also carry out experiments using the original MNIST digit images [47]. The used data set consists of 60 000 training samples and 10 000 testing samples. Fig. 5(d) shows some sample images.

Experimental Setups: In each test, we randomly partition the used data set into two parts for training and testing. Following the common benchmarking procedures, we repeat the experiment multiple times (e.g., ten times) with different training and testing data partitions. We report the final results with several measures, i.e., the mean, standard deviation, and median of the recognition rates and the time costs. Moreover, we also investigate the latency of BOE-based classification system.

A. Robustness to Hyperparameters

The proposed AER categorization system requires to specify two hyperparameters, i.e., the number of events within each segment (denoted by α) and the training data percentage γ. To investigate the influence of these two hyperparameters, we carry out experiments on MNIST-DVS.

Fig. 6(a) and (b) shows the classification accuracy of BOE when α increases from 50 to 500 and γ increases from 10% to 90%. From the results, we have the following observations.

1) BOE is robust to the number of event within each segment. While α ranges from 150 to 500, the classification

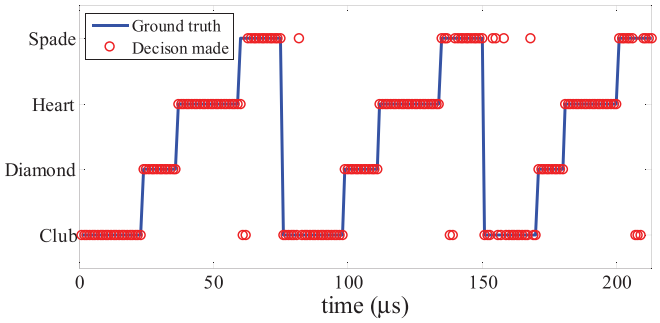


Fig. 9. Classification result of the proposed method on AER Poker Card database. We can see that the predicted labels by BOE match very well with the ground truth.

TABLE III

RECOGNITION RATE ON THE AER POKER CARD DATABASE. n DENOTES THE NUMBER OF SEGMENT AND α DENOTES THE AVERAGE NUMBER OF EVENT WITHIN EACH SEGMENT

Algorithms	mean	std.	median	n	α
BOE	**93.00%**	5.29%	**93.88%**	519	100
Zhao's [18]	91.76%	6.19%	92.59%	519	100
Chen's [15]	92.53%	4.45%	93.12%	519	100

TABLE IV

COMPUTATIONAL EFFICIENCY OF DIFFERENT METHODS FOR FEATURE EXTRACTION AND CLASSIFICATION ON THE AER POKER CARD DATABASE. FOR ALL THE METRICS EXCEPT fps, THE VALUE IS BIGGER, AND THE PERFORMANCE IS BETTER

Algorithms	Feature Extraction					Classification				
	training(s)	testing(s)	total(s)	fps	tpe(s)	training(s)	testing(s)	total(s)	fps	tpe(s)
BOE	**0.02**	**0.02**	**0.04**	**14828.57**	**6.74E-07**	**0.01**	**1.30E-03**	**0.01**	**36293.71**	**2.76E-07**
Zhao's [18]	70.23	7.80	78.03	6.65	1.50E-03	2.89	0.11	3.00	173.00	5.78E-05
Chen's [15]	30.99	3.44	34.43	15.07	6.63E-04	-	19.90	19.90	26.08	3.83E-04

rate almost remains unchanged, slightly varying from 74.10% to 74.15%.

2) BOE can achieve a good result even though a small amount of training data is available. While γ is increasing from 30% to 90%, the mean recognition rate of BOE is increasing from 70.99 to 74.28. This benefits from a fact that BOE is a statistical feature extraction method. In addition, the well-known advantages of statistical machine learning method are its good generalization ability. In other words, BOE can fit a latent distribution well with a small amount of samples.

B. Scalability Performance Comparison

The low computational cost and the high energy efficiency are two most important advantages of AER sensors. Thus, it is important to put these two advantages in the first place while designing an AER categorization system. In this section, we examine the scalability performance of our system, Zhao's method, and Chen's method. We carry out experiments on the MNIST-DVS database with increasing number of segments (i.e., n). For BOE and Chen's method, we set the value of the hyperparameter α as 300. For Zhao's method, we set the time constant τ_m of the MSD as 30 ms by following the configuration in [18], and the number of the corresponding bagged events is around 300.

We perform each algorithm ten times on ten different data partitions. For each test, 90% data are used for training, and the remaining data are used for testing. Fig. 7 shows the classification accuracy and the time cost, which shows that the following holds.

1) BOE is superior to the other investigated methods in classification accuracy. For example, when $n = 1000$, the accuracy achieved by BOE is 5.85% higher than that achieved by Zhao's method and is 12.43% higher than that achieved by Chen's method.
2) With the increasing of n, three methods achieve a better accuracy. When less data are available, Zhao's method achieves the worst result. The reason is that this method is based on deep learning methods, which need large-scale data to be well trained.
3) When n increases from 100 to 5000, the time cost taken by BOE only increases from 0.59 to 22.09 s. Under the same computational platform, Zhao's method used 115.09 and 4836.91 s to handle 100 and 5000 segments, respectively. The results show that our method finds a good balance between classification accuracy and time cost.

TABLE V

RECOGNITION RATE ON THE AER POSTURE DATABASE. n DENOTES THE NUMBER OF SEGMENT AND α DENOTES THE AVERAGE NUMBER OF EVENT WITHIN EACH SEGMENT

Algorithms	mean	std.	median	n	α
BOE	**98.66**%	0.23%	**98.65**%	24639	500
Zhao's [18]	95.61%	0.46%	95.50%	17414	714
Chen's [15]	91.88%	0.68%	91.97%	24639	500

4) Regarding the frame per second (fps) the performance of BOE ranges from 141.24 to 304.32, whereas Zhao's method can only handle 1 frame/s ([0.87, 1.06]) and Chen's method can only handle 1.81–4.58 frames/s. Note that there is no relationship between α and fps. α determines the number of segments for a given event flow, whereas the fps is only related to the computational power of the computer.

C. Performance on Different AER Data Sets

In this section, we report the performance of BOE on MNIST-DVS, Card, and Posture database with respect to the classification accuracy and efficiency. To obtain a more comprehensive comparison, we use five metrics to measure the computational efficiency of the tested algorithms for the feature extraction and classification. The metrics include the time cost for training, testing, total computation (i.e., training cost plus testing cost), fps, and tpe, where fps is short for frame per second and tpe denotes the time cost for processing each event.

1) On MNIST-DVS Data Set: We carry out the experiment on MNIST-DVS data set by repeating each method ten times. For each test, 90% samples are randomly selected for training, and the remaining data are used for testing. For BOE, we set $\alpha = 300$. Fig. 8 and Table II show the classification results and the time cost, respectively.

1) Fig. 8 shows that BOE and Zhao's method outperform Chen's method by a performance margin of 13.86% and 14.12%, respectively. If we average the accuracy on individual class instead of data points, the performance rates of BOE, Zhao's method, and Chen's method are 74.82 ± 11.77, 75.52 ± 11.17, and 63.50 ± 14.84, respectively.
2) Table II shows that Zhao's method takes more time for feature extraction than BOE and Chen's method. It only can process 1.87 segments within 1 s, whereas our method can run at 402.65 frames/s.

TABLE VI

COMPUTATIONAL EFFICIENCY OF DIFFERENT METHODS FOR FEATURE EXTRACTION AND CLASSIFICATION ON THE AER POSTURE DATABASE. FOR ALL THE METRICS EXCEPT fps, THE VALUE IS BIGGER, AND THE PERFORMANCE IS BETTER

Algorithms	Feature Extraction					Classification				
	training(s)	testing(s)	total(s)	fps	tpe(s)	training(s)	testing(s)	total(s)	fps	tpe(s)
BOE	**44.78**	**44.52**	**89.31**	**275.89**	**7.25E-06**	0.68	**0.12**	**0.80**	**30883.68**	**6.48E-08**
Zhao's [18]	11770.87	2942.72	14713.58	1.18	1.18E-03	23.62	5.08	28.71	606.55	2.31E-06
Chen's [15]	1548.19	387.05	1935.24	12.73	1.57E-04	-	11430.28	11430.28	2.16	9.28E-04

3) Chen's system employs a lazy classifier to perform categorization. Therefore, it does not need to train the classifier. Our AER categorization system can classify 5926.63 segments/s, which is 76.74 and 2034.15 times faster than Zhao's system and Chen's system, respectively.

2) On AER Poker Card Data Set: This section investigates the performance of three methods on AER Card database. For each method, we still randomly select 90% data for training and perform the evaluation 100 times. Fig. 9 shows the predicted labels of BOE ($\alpha = 100$) by passing the event stream into our system. Moreover, Tables III and IV show the performance comparison of the tested methods, which show that the following holds.

1) BOE achieves the highest classification accuracy and Chen's method archives the second best results. It should be pointed out that all the tested methods employ the same event segmentation method in this test, i.e., by fixing the number of event within each segment.
2) DVS camera asynchronously outputs the events with submicrosecond delay (i.e., 10^{-6} s. From Table IV, we can see that the proposed system processes each event at the temporal resolution of 10^{-7}, i.e., it can process the events in real time. This simultaneity or coincidence property is very attractive for the AER processing system, as pointed out in [16].

3) On AER Posture Data Set: In this section, we carry out the experiment on AER Posture database by repeating each algorithm ten times. In the test, we randomly select 80% actions for training and use the rest for testing. We fix $\alpha = 500$ for BOE and Chen's method and set the search range of MSD of Zhao's method as 30 ms.

From Tables V and VI, we can find that the following holds.

1) BOE outperforms Zhao's method and Chen's method by the performance margin of 3.05% and 6.78%. Note that Zhao's method achieves a correct rate of 99.48% in [18] when the search range of MSD is set as 1 s. Here, we set the search range as 30 ms for fair comparison, because the corresponding α is around 500. Note that a bigger search range means that less decisions are made, and thus, the classification accuracy may be higher.
2) For the feature extraction phase, the calculation speed of our method is 164.75 and 21.67 times faster than Zhao's method and Chen's method, respectively. Furthermore, our method also takes the minimal time to perform classification. It is 35.99 and 14 327.25 times faster than Zhao's method and Chen's method.

TABLE VII

PERFORMANCE OF THE EVALUATED ALGORITHMS ON 70 000 ORIGINAL MINIST IMAGES, WHERE nBOE DENOTES THE nBOE FEATURES

Methods	Acc.	Training Cost		Testing Cost	
		total(s)	fps	total(s)	fps
SVM	84.17	196.61	305.17	0.20	50000.00
BOE	88.09	4.93	12170.39	0.49	20408.16
nBOE	**91.82**	44.05	1362.24	0.33	30444.16

TABLE VIII

COMPARISONS WITH SPARSE REPRESENTATION ON A SUBSET OF THE ORIGINAL MINIST IMAGES

Methods	Acc.	Training Cost		Testing Cost	
		total(s)	fps	total(s)	fps
BOE	**86.50±2.42**	0.13	7692.31	2.50E-3	40000
Sparse Representation	82.30±4.03	265.90	3.76	19.79	5.05

4) Performance on the Standard MNIST Image Data Set: In this section, we evaluate the performance of BOE on the raw MNIST digital images. In experiments, we adopt the standard testing protocol [47] by using 60 000 samples for training and 10 000 samples for testing. In the test, we directly apply Support vector machine (SVM) over the original data to obtain a baseline result. Besides the results of BOE with SVM, we also carry out experiment by applying SVM on the normalized BOE (nBOE) features. More specifically, we normalize each BOE feature vector $\mathbf{x}$ by its maximal entry, i.e., $\mathbf{x} \leftarrow \mathbf{x}/\max(\mathbf{x})$, where the operator $\max(\cdot)$ achieves the maximal element of a given vector.

For each algorithm, we evaluate its efficiency with time cost as well as fps. For example, BOE takes 4.93 s to handle 60 000 training samples, which consists of the costs for extracting BOE features and training SVM. By dividing 60 000 by 4.93, we can see that BOE can handle 12 170.39 frames/s. Table VII shows the result, which shows that BOE cannot only improve the recognition accuracy but also speed up the convergence of SVM in the training phase.

5) Performance Comparisons With Sparse Representation: In this section, we compared the nBOE features with another well-known low-level feature extraction method, i.e., sparse representation (SR) [48]. Sparse representation represents each sample as a linear combination of a few of basis, which has attracted increasing

TABLE IX
LATENCY OF OUR CATEGORIZATION SYSTEM

Algorithmic Stage	Training Data			Testing Data		
	MNIST-DVS	Cards	Posture	MNIST-DVS	Cards	Posture
Event Accumulation (ms)	37.03	0.34	75.16	37.03	0.34	75.16
Feature Extraction (ms)	1.39	0.04	2.02	1.23	0.33	18.07
Classification (ms)	0.18	0.03	0.03	0.05	0.25	0.05
Total (ms)	38.61	0.41	77.21	38.31	0.92	93.28
fps	27.01	2941.18	13.30	27.01	2941.18	13.30

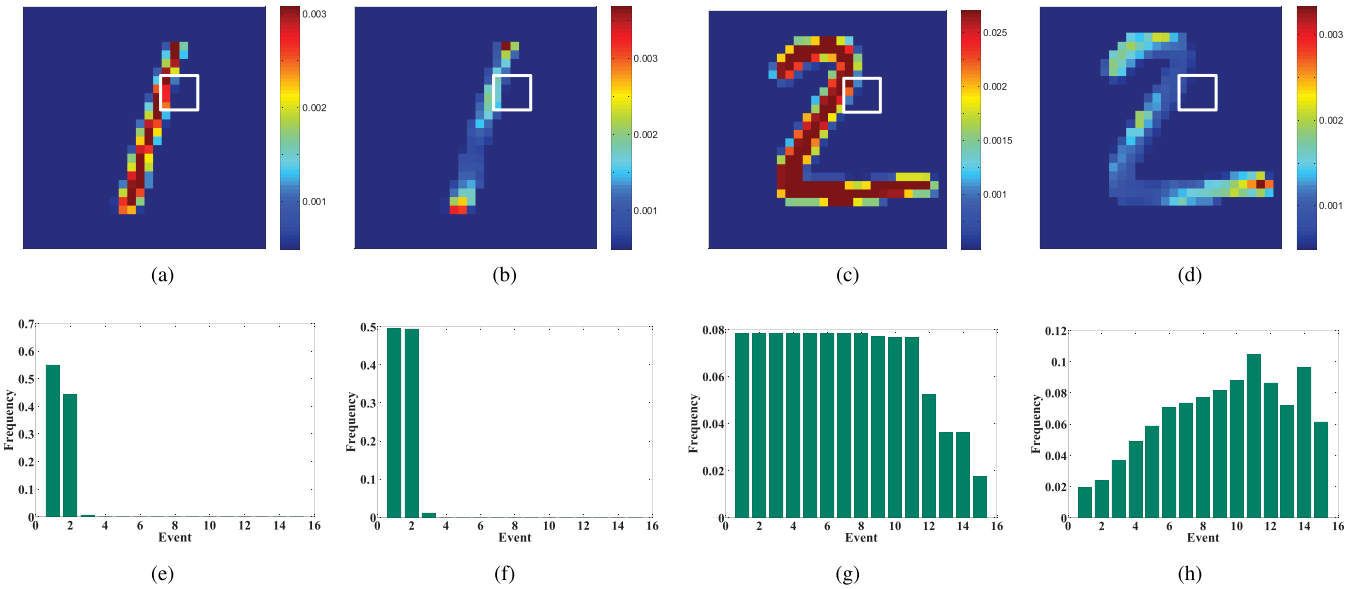


Fig. 10. Real example to show the discrimination of BOE features. (a) Sample of digit 1. (b) BOE feature of digit 1. (c) Sample of digit 2. (d) BOE feature of digit 2. (e) Local area of (a). (f) Local area of (b). (g) Local area of (c). (h) Local area of (d). For a given DVS sensor, it will activate different event addresses for different patterns, as shown in (a) and (c) (e.g., digits 1 and 2). Due to the existence of overlapping between different patterns, some event addresses will be frequently activated, which are less discriminative, e.g., the first and second event addresses will be used to represent digits 1 and 2 [see (e) and (g)]. By applying our BOE approach over the DVS output, the importance (i.e., frequency) of the first and second event addresses will be reduced, while increasing the frequency of the event addresses that are only activated for digit 2 [see (h)].

interests from machine learning and computer vision. In general, sparse representation can be achieved by solving a ℓ_1-minimization problem, which is a convex relaxation of the ℓ_0-minimization problem. In our experiments, we adopted the well-known homotopy solver [49] to calculate the sparse representation of the inputs and, then, performed recognition by passing the sparse representation into a linear SVM.

As sparse representation is computationally inefficient, we carried out experiments using a subset of the MNIST database consisting of 1000 training samples and 100 testing samples that are randomly drawn from the original MNIST database. Table VIII shows the results, and we can see that BOE remarkably outperforms sparse code in terms of recognition rate and computational efficiency.

D. Performance With Respect to Latency

In the above analysis, we investigated the computational cost of BOE with respect to the stage of feature extraction and classification. In this section, we further examine the end-to-end system latency, i.e., the delay between receiving the first event and outputting the corresponding label. This investigation involves three stages, i.e., event accumulation, feature extraction, and classification. For each stage, we calculate the mean time cost over all segments. Table IX shows the results from which we have the following observations.

1) The feature extraction and classification stages take much less time compared with the event accumulation stage. This is due to the low complexity of our algorithm.
2) The time cost of event accumulation depends on the value of α, which is determined based on the characteristics of data. For different applications, we can increase or decrease the latency of our AERCsys by changing the value of α. For example, we set $\alpha = 100$ for the Card database and $\alpha = 500$ for the Posture database, and thus, our system can handle 2941.18 and 13.30 segments within each second, respectively. This actually reflects some characteristics of these two stimuli, i.e., the movement of human is slower than that of poker cards in practice.

E. Why BOE Features Are Discriminative?

In this section, we investigate the discrimination of our BOE. In the experiments, we perform the BOE method on a subset of the MNIST database, which consists of all the testing samples of digit 1 [see Fig. 10(a)] and digit 2 [see Fig. 10(c)]. For better illustrations, we also show some

pixels into a given box [shown in Fig. 10(a)–(d)]. From the comparison between the original data and the corresponding BOE features, we can see that the BOE method will obtain a more discriminative feature by increasing the frequency of the events that are only activated by one digit (1 or 2), as well as decreasing the frequency of the events that are activated by both these two digits.

V. Conclusion

In this paper, we proposed a feature extraction method for AER image sensors based on the probability theory, namely, BOE. We provided two explanations to our method: the first one intuitively shows our basic idea, i.e., BOE is the combination of the information measurements of speciality and popularity and the second one theoretically shows the connections between the BOE and the quantity of the expected mutual information. Moreover, BOE is an online feature extraction method, i.e., it can handle the labeled and unlabeled data that are alternately received. Experimental results demonstrate that our method is significantly faster than two recently proposed methods while achieving a competitive recognition accuracy.

This paper can be extended or improved from the following aspects. First, BOE is an unsupervised method. It is possible to further improve the discrimination of BOE features by incorporating the label information, e.g., developing the supervised or semisupervised BOE method. Second, the basic formulation of BOE [i.e., (3)] might be extended into a more general, and thus, other information measurements, such as information gain, can be incorporated into our mathematical formulation. Third, like most existing AER feature extraction methods, BOE requires accumulating events into segments. Although BOE represents each stimulus using the JPD of consecutive events and does not extract visual features, such as lines from segments, it is more interesting and challenging to explore how to utilize each single event as a source of meaningful information without segment reconstruction. This daunting task has lied on the heart of current neuromorphic computing and will be explored in the future.

Acknowledgment

The authors would like to thank the Associate Editor and anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper.

References

[1] V. Chan, S.-C. Liu, and A. van Schaik, "AER EAR: A matched silicon cochlea pair with address event representation interface," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 1, pp. 48–59, Jan. 2007.

[2] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers Neurosci.*, vol. 5, p. 73, May 2011.

[3] T. Chang, Y. Yang, and W. Lu, "Building neuromorphic circuits with memristive devices," *IEEE Circuits Syst. Mag.*, vol. 13, no. 2, pp. 56–73, May 2013.

[4] D. Monroe, "Neuromorphic computing gets ready for the (really) big time," *Commun. ACM*, vol. 57, no. 6, pp. 13–15, 2014.

[5] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[6] 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.

[7] 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, Feb. 2008.

[8] J. A. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, "A 3.6 μs latency asynchronous frame-free event-driven dynamic-vision-sensor," *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1443–1455, Jun. 2011.

[9] R. Berner, T. Delbruck, A. Civit-Balcells, and A. Linares-Barranco, "A 5 Meps $100 USB2.0 address-event monitor-sequencer interface," in *Proc. 20th IEEE Int. Symp. Circuits Syst.*, New Orleans, LA, USA, May 2007, pp. 2451–2454.

[10] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 5, pp. 416–434, May 2000.

[11] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. J. Douglas, and T. Delbruck, "A pencil balancing robot using a pair of AER dynamic vision sensors," in *Proc. 22nd IEEE Int. Symp. Circuits Syst.*, Taipei, Taiwan, May 2009, pp. 781–784.

[12] R. Serrano-Gotarredona *et al.*, "CAVIAR: A 45k neuron, 5M synapse, 12G connects/s AER hardware sensory–processing–learning–actuating system for high-speed visual object recognition and tracking," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1417–1438, Sep. 2009.

[13] B. Zhao, X. Zhang, S. Chen, K.-S. Low, and H. Zhuang, "A 64 × 64 CMOS image sensor with on-chip moving object detection and localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 581–588, Apr. 2012.

[14] S. Chen, W. Tang, X. Zhang, and E. Culurciello, "A 64 × 64 pixels UWB wireless temporal-difference digital image sensor," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 12, pp. 2232–2240, Dec. 2012.

[15] S. Chen, P. Akselrod, B. Zhao, J. A. Pérez-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.

[16] J. A. Pérez-Carrasco *et al.*, "Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing—Application to feedforward ConvNets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2706–2719, Nov. 2013.

[17] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Frontiers Neurosci.*, vol. 7, p. 178, Oct. 2013.

[18] B. Zhao, R. Ding, S. Chen, B. Linares-Barranco, and H. Tang, "Feedforward categorization on AER motion events using cortex-like features in a spiking neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 1963–1978, Sep. 2015.

[19] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLoS Comput. Biol.*, vol. 3, no. 2, p. e31, Feb. 2007.

[20] M. Litzenberger *et al.*, "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *Proc. 4th Digit. Signal Process. Workshop, 12th Signal Process. Edu. Workshop*, Teton National Park, WY, USA, Sep. 2006, pp. 173–178.

[21] T. Delbruck and P. Lichtsteiner, "Fast sensory motor control based on event-based hybrid neuromorphic-procedural system," in *Proc. IEEE Int. Symp. Circuits Syst.*, New Orleans, LA, USA, May 2007, pp. 845–848.

[22] 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 Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1710–1720, Aug. 2014.

[23] J. H. Lee *et al.*, "Real-time gesture interface based on event-driven processing from stereo silicon retinas," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2250–2263, Dec. 2014.

[24] R. J. Vogelstein, U. Mallik, E. Culurciello, G. Cauwenberghs, and R. Etienne-Cummings, "A multichip neuromorphic system for spike-based visual information processing," *Neural Comput.*, vol. 19, no. 9, pp. 2281–2300, Sep. 2007.

[25] S.-H. Ieng, C. Posch, and R. Benosman, "Asynchronous neuromorphic event-driven image filtering," *Proc. IEEE*, vol. 102, no. 10, pp. 1485–1499, Oct. 2014.

[26] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 407–417, Feb. 2014.

[27] R. Benosman, S.-H. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan, "Asynchronous frameless event-based optical flow," *Neural Netw.*, vol. 27, pp. 32–37, Mar. 2012.

[28] L. A. Camuñas-Mesa, T. Serrano-Gotarredona, S. H. Ieng, R. B. Benosman, and B. Linares-Barranco, "On the use of orientation filters for 3D reconstruction in event-driven stereo vision," *Frontiers Neurosci.*, vol. 8, p. 48, Mar. 2014.

[29] C. Brandli, L. Muller, and T. Delbruck, "Real-time, high-speed video decompression using a frame- and event-based DAVIS sensor," in *Proc. 27th IEEE Int. Symp. Circuits Syst.*, Melbourne, VIC, Australia, Jun. 2014, pp. 686–689.

[30] E. Stromatias, D. Neil, M. Pfeiffer, F. Galluppi, S. B. Furber, and S.-C. Liu, "Robustness of spiking deep belief networks to noise and reduced bit precision of neuro-inspired hardware platforms," *Frontiers Neurosci.*, vol. 9, p. 222, Jul. 2015.

[31] E. Stromatias, F. Galluppi, C. Patterson, and S. Furber, "Power analysis of large-scale, real-time neural networks on SpiNNaker," in *Proc. 25th Int. Joint Conf. Neural Netw.*, Dallas, TX, USA, Aug. 2013, pp. 1–8.

[32] E. Stromatias, D. Neil, F. Galluppi, M. Pfeiffer, S.-C. Liu, and S. Furber, "Scalable energy-efficient, low-latency implementations of trained spiking deep belief networks on SpiNNaker," in *Proc. 27th Int. Joint Conf. Neural Netw.*, Killarney, Ireland, Jul. 2015, pp. 1–8.

[33] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neurosci.*, vol. 2, no. 11, pp. 1019–1025, Nov. 1999.

[34] R. Gütig and H. Sompolinsky, "The tempotron: A neuron that learns spike timing–based decisions," *Nature Neurosci.*, vol. 9, no. 3, pp. 420–428, 2006.

[35] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[36] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge Univ. Press, 2008.

[37] *Java AER Open Source Project*, accessed on Jan. 2016. [Online]. Available: http://sourceforge.net/p/jaer/wiki/

[38] L. Camuñas-Mesa, C. Zamarreño-Ramos, A. Linares-Barranco, A. J. Acosta-Jiménez, T. Serrano-Gotarredona, and B. Linares-Barranco, "An event-driven multi-kernel convolution processor module for event-driven vision sensors," *IEEE J. Solid-State Circuits*, vol. 47, no. 2, pp. 504–517, Feb. 2012.

[39] J. Hu, H. Tang, K. C. Tan, H. Li, and L. Shi, "A spike-timing-based integrated model for pattern recognition," *Neural Comput.*, vol. 25, no. 2, pp. 450–472, 2013.

[40] 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. Learn. Syst.*, vol. 24, no. 10, pp. 1539–1552, Oct. 2013.

[41] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns," *PLoS ONE*, vol. 8, no. 11, p. e78318, Nov. 2013.

[42] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Inf. Process. Manage.*, vol. 39, no. 1, pp. 45–65, 2003.

[43] Y. Gao and M. K. H. Leung, "Face recognition using line edge map," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 764–779, Jun. 2002.

[44] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.

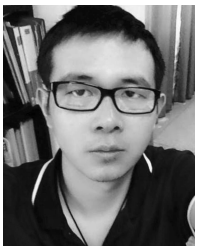
[45] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128 × 128 1.5% contrast sensitivity 0.9% FPN 3 μs latency 4 mW asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 827–838, Mar. 2013.

[46] B. Zhao, S. Chen, and H. Tang, "Bio-inspired categorization using event-driven feature extraction and spike-based learning," in *Proc. 26th Int. Joint Conf. Neural Netw.*, Beijing, China, Jul. 2014, pp. 3845–3852.

[47] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[48] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[49] M. R. Osborne, B. Presnell, and B. A. Turlach, "A new approach to variable selection in least squares problems," *IMA J. Numer. Anal.*, vol. 20, no. 3, pp. 389–403, 2000.



Xi Peng received the B.Eng. degree in electronics engineering and the M.Eng. degree in computer science from the Chongqing University of Posts and Telecommunications, Chongqing, China, and the Ph.D. degree from Sichuan University, Chengdu, China.

He is currently a Research Scientist with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. His current research interests include computer vision, image processing, and pattern recognition.

Dr. Peng is a recipient of the Excellent Graduate Student of Sichuan University, the National Graduate Scholarship, the Tang Lixin Scholarship, the CSC–IBM Scholarship for Outstanding Chinese Students, and the Excellent Student Paper of the IEEE Chengdu Section. He has served as a Guest Editor of *Image and Vision Computing*, a PC Member/Reviewer of ten international conferences, such as the Association for the Advancement of Artificial Intelligence, the International Joint Conference on Neural Networks, and IEEE World Congress on Computational Intelligence, and a Reviewer of over ten international journals, such as the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and the IEEE TRANSACTIONS ON CYBERNETICS.

Bo Zhao (M'11) received the B.Eng. and M.Eng. degrees in electronics engineering from Beijing Jiaotong University, Beijing, China, in 2007 and 2009, respectively, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2014.

He is currently a Research Scientist with the Institute of Infocomm Research, Agency for Science, Technology and Research, Singapore. His current research interests include neuromorphic vision processing, spiking neural networks, biologically inspired object recognition, and very large scale integration circuits and systems design.

Rui Yan (M'11) received the bachelor's and master's degrees from the Department of Mathematics, Sichuan University, Chengdu, China, in 1998 and 2001, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, in 2006.

She is currently a Professor with the College of Computer Science, Sichuan University. Her current research interests include intelligent robots, nonlinear control, neural computation, and power systems analysis and control.

Huajin Tang (M'01) received the B.Eng. degree from Zhejiang University, Hangzhou, China, in 1998, the M.Eng. degree from Shanghai Jiao Tong University, Shanghai, China, in 2001, and the Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2005.

He was a System Engineer with STMicroelectronics, Singapore, from 2004 to 2006. He was a Post-Doctoral Fellow with the Queensland Brain Institute, University of Queensland, Brisbane, QLD, Australia, from 2006 to 2008. He was a Group Leader of Cognitive Computing with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore, from 2008 to 2015. He is currently a Professor with the College of Computer Science, Sichuan University, Chengdu, China. He has authored one monograph (Springer-Verlag, 2007) and over 30 international journal papers. His current research interests include neuromorphic computing, cognitive systems, robotic cognition, and machine learning.

Dr. Tang serves as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS, and an Editorial Board Member of *Frontiers in Robotics and AI*. He served as the Program Chair of the Seventh IEEE International Conference on Cybernetics and Intelligent Systems and Robotics, Automation and Mechatronics (CIS-RAM) in 2015, and the Co-Program Chair of the Sixth IEEE International Conference on CIS-RAM in 2013.

Zhang Yi (F'15) received the Ph.D. degree in mathematics from the Institute of Mathematics, Chinese Academy of Sciences, Beijing, China, in 1994.

He is currently a Professor with the Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu, China. He has co-authored three books entitled *Convergence Analysis of Recurrent Neural Networks* (Kluwer Academic Publishers, 2004), *Neural Networks: Computational Models and Applications* (Springer, 2007), and *Subspace Learning of Neural Networks* (CRC Press, 2010). His current research interests include neural networks and big data.

Dr. Yi was an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS from 2009 to 2012 and the IEEE TRANSACTIONS ON CYBERNETICS in 2014.