Modeling Activity-Dependent Plasticity in BCM Spiking Neural Networks with Application to Human Behavior Recognition

Journal:	*Transactions on Neural Networks*
Manuscript ID:	TNN-2011-P-3276.R1
Manuscript Type:	Paper
Keywords:	Spiking neural network, BCM model, Neural plasticity, Gene regulatory network, Evolution strategy, Human behavior recognition

Modeling Activity-Dependent Plasticity in BCM Spiking Neural Networks with Application to Human Behavior Recognition

Yan Meng, Yaochu Jin, and Jun Yin

***Abstract*—Spiking neural networks are considered to be computationally more powerful than conventional neural networks. However, the capability of spiking neural networks in solving complex real-world problems remains to be demonstrated. In this paper, we propose a substantial extension of the Bienenstock, Cooper, and Munro (BCM) spiking neural network model, in which the plasticity parameters are regulated by a gene regulatory network (GRN). Meanwhile, the dynamics of the GRN is dependent on the activation levels of the BCM neurons. We term the whole model GRN-BCM. To demonstrate its computational power, we first compare the GRN-BCM with a standard BCM, a hidden Markov model and a reservoir computing model on a complex time series classification problem. Simulation results indicate that the GRN-BCM significantly outperforms the compared models. The GRN-BCM is then applied to two widely used datasets for human behavior recognition. Comparative results on the two datasets suggest that the GRN-BCM is very promising for human behavior recognition, although the current experiments are still limited to the scenarios in which only one object is moving in the considered video sequences.**

***Index Terms*—spiking neural network, BCM model, neural plasticity, gene regulatory network, evolution strategy, human behavior recognition.**

I. Introduction

SPIKING neural networks (SNNs) are believed to be biologically more plausible [9, 44] and computationally more powerful than analog neural networks [46]. However, only a small number of successful application examples of SNNs to real-world problems have so far been reported.

Liquid state machine (LSM) [46] is a computational framework neural computation using SNNs. LSM contains a number of randomly connected spiking neurons, which also receive inputs from outside resources. The state (membrane potential) of the neurons can be read out and combined as a target output. Together with the echo state networks [33], LSMs have now become one of the two main streams in reservoir computing (RC) [55].

Among different learning algorithms for SNNs, the Bienenstock, Cooper, and Munro (BCM) model [8] was proposed for modeling synaptic plasticity of spiking neural networks, which is believed to be biologically plausible [17, 37, 53], as it has successfully predicted experimental data in the visual cortex and the hippocampus. However, a biological plausible mechanism for synaptic plasticity in the BCM model for spiking neural networks still remained to be established [36].

This work aims to model activity-dependent neural plasticity to enhance the computational power of the BCM model. Our idea is mainly inspired from biological findings indicating that both the structure and connecting weights of the neurons in the brain can change over time depending on the neuronal activities [34]. Recent evident also indicate that activity-dependent neural plasticity is resulted from changes in the expression of relevant genes [22]. To model the interaction between the dynamics of gene expression and neural activation, a new model called GRN-BCM is proposed in this paper, where a gene regulatory network (GRN) is developed to regulate the synaptic plasticity and meta-plasticity of a BCM spiking neural network.

To verify the capability of the GRN-BCM model for temporal information processing, a set of empirical studies has been performed by comparing the GRN-BCM model with a few other models for temporal information processing on a synthetic time series dataset and two datasets for human behavior recognition.

This paper is a substantial extension of our previous work [47], where some preliminary results were presented. Major contributions of this paper can be summarized as follows. First, a novel GRN-BCM model is proposed for the temporal pattern learning in human behavior recognition. In this model, the synaptic plasticity and meta-plasticity of the BCM model, such as the learning rate, forgetting factor and time-delay, are no longer a constant, rather a dynamic process regulated by a biological inspired GRN. The GRN dynamics is also influenced by the activation levels of the neurons it resides in, resulting in a coupled dynamic system. Second, the parameters of the GRN are optimized using an efficient evolutionary algorithm, the evolution strategy with covariance matrix adaptation (CMA-ES) [27, 28], which has shown to be efficient for real-valued optimization. Third, extensive experiments on two widely used datasets for human behavior recognition have been conducted using the proposed GRN-BCM model. The recognition performance of the GRN-BCM model outperforms many other state-of-the-art algorithms for human behavior detection on these two datasets.

Whereas the results in this work suggest that the GRN-BCM model is very promising for spatiotemporal pattern recognition such as human behavior detection, it should also be pointed that all scenarios in the two datasets used in this paper have only one single moving object. Although sophisticated methodologies

Yan Meng and Jun Yin are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA (Phone: +1 201-2165496, E-mail: {yan.meng; jyin}@stevens.edu).

Yaochu Jin is with the Department of Computing, University of Surrey, Guildford, Surrey, GU2 7XH, UK (Phone: +44 1483686037, E-mail: yaochu.jin@surrey.ac.uk).

for multiple-object detection and tracking have been reported [16, 51], the capability of the GRN-BCM model to deal with multi-object behavior recognition remains to be investigated.

The rest of the paper is organized as follows. Related work is discussed in Section II. Section III introduces the background of the BCM model, the GRN model, and the CMA-ES. The entire GRN-BCM model is described in detail in Section IV. The convergence property of the GRN-BCM model as well as its capability for time series data classification is investigated in Section V. To further demonstrate the computational power of the model, we apply the GRN-BCM model to human behavior recognition in Section VI together with a spatial feature extraction algorithm. For this purpose, a corner-based method is developed for extracting spatial features. The sequential frames with these extracted spatial features serve as the inputs to the GRN-BCM model. Section VII presents comparative studies on two benchmark problems for human behavior recognition. The paper is concluded by Section VIII.

II. Related Work

A. Spatiotemporal Feature Extraction

Using machine learning techniques to extract and learn temporal patterns from large amount of data has attracted increasing interests in various research areas, such as visual human behavior recognition, speech recognition and object tracking. In these real-world applications, temporal information is inherently embedded within the input data. However, how to extract and learn the temporal pattern is still a challenging problem. In computer vision, numerous approaches for analyzing human behaviours from video sequences, including human motion capturing, tracking, segmentation and recognition have been reported in recent surveys [38]. Much existing work on human behavior detection focuses on extracting a sequence of spatiotemporal features from the video sequences first, then employs some standard classifiers, such as support vector machines (SVM) [31, 56], neural networks [52], and relevance vector machines (RVM) [60] to classify these features. However, extracting efficient spatiotemporal features from video sequences is challenging.

Recently, some recurrent neural networks (RNN) models have been proposed to deal with temporal information by adding recurrent connections to the learning models. These RNN models have been applied to various real-world applications, such as object tracking and motion prediction [10], [44], event detection [30], and speech recognition [25]. Efforts have also been made to learn the temporal relationship between the sequential features using event-based analysis [68] and hidden Markov model (HMM) [18, 61, 67].

In human behavior recognition, feature extraction consists of not only spatial features but also temporal features. Ke et al. [35] applied spatiotemporal volumetric features that efficiently scan video sequences in both spatial and temporal dimensions from optical flow fields. Alternatively, local spatiotemporal patches of videos can be extracted [20]. Laptev et al. [40] presented a spatiotemporal interest point detector based on the idea of the interest point operator in Harris [29], where image values have significant local variations in both spatial and temporal dimensions. This spatiotemporal representation has been successfully applied to human action recognition combined with an SVM classifier in [56]. Dollar et al. [15] proposed an alternative approach to detecting sparse spatiotemporal interest points based on separable linear filters, and used a nearest neighbor based classifier for behavior recognition. The above mentioned approaches demonstrated that local spatiotemporal patches are useful for extracting semantic meaning from video events by providing a compact and abstract representation of patterns.

However, the pattern recognition models adopted in the above mentioned approaches are inadequate to handle complex scenarios where, e.g., dynamic backgrounds and occlusions are involved. In addition, the above methods often require a large number of training samples, which is time-consuming and difficult to meet real-time requirements for online behavior recognition. To tackle these issues, visual codebooks for object/human recognition have been investigated [11, 13, 41, 49]. The visual codebook was created by clustering the extracted feature descriptors in the training set, where each resulting center was considered as a codeword, and a set of code-words forms a ‘codebook’. The major limitation of this “bag-of-words” approach is that the relative position information of the “words” is lost.

The GRN-BCM model proposed in this work is capable of learning temporal information, and therefore, only spatial features need to be extracted for each image frame. Then the sequential frames with the extracted spatial features are used as the input to the GRN-BCM model so that spatiotemporal features can further extracted and classified coherently by the GRN-BCM model. We will discuss this point in more detail in Section VI.

B. Learning Algorithms and Biophysical Neural Models

Once spatial features are extracted from the data, the next step is to learn the temporal relationship between the frame sequences, which is presumably a very critical step in human behaviors recognition. Over the past decade, models and mechanisms for temporal sequence learning have received considerable research attention. In [42], an online sequential extreme learning machine (OS-ELM) was proposed for fast and accurate sequential learning. This algorithm can learn data either one-by-one or chunk-by-chunk. Tijsseling [59] proposed a categorization-and-learning-module (CALM) network with time-delayed connections for sequential information processing. In the CALM, the module can autonomously adjust its structure according to the complexity of the application domain. These methods can be used to explicitly represent temporal information as an additional dimension together with the pattern vector.

Spiking neural networks with Hebbian plasticity have been considered well suited for learning temporal patterns [23]. Arena et al. [3] applied a spiking neural network by using a causal Hebbian rule for a robot navigation control. Wu et al. [66] has suggested a motion detection algorithm using an integrate-and-fire spiking neuron model consisting of two

intermediate layers. The main idea is to repress the activation of neurons whose activity does not change over time and to reinforce those having changed activities. Their model is limited to motion detection only and cannot be used for temporal pattern recognition.

In recent years, a few biophysical neural network models have been proposed, in which temporal sensitivity of the neurons is exploited in information computation. Bohte et al. [9] developed an error regression learning rule to train a network of spiking neurons, which was not based on the Hebbian approach to synaptic weight modification. Indeed, many learning rules, developed for use on spike-time-dependent models [24], rely on the Hebbian correlation as the principal means for synaptic weight modification. In [54], a single biologically plausible spike-rate model is used to solve a nonlinear tractable task using back-propagation learning rules. Shin et al. [57] proposed a synaptic plasticity activity rule (SAPR) to learn connections between the spiking neurons for image recognition.

In the BCM model adopted in this work, weight plasticity is essential for storing temporal information from a sequence of data. Edelman et al. [19] used a single layer network and a learning rule based on BCM to extract a visual code in an unsupervised manner. The role of calcium concentration in the spike time dependent plasticity (STDP) of the BCM model neurons was investigated by Kurashige and Sakai [39]. Shouval et al. [58] revealed evidence for the calcium control hypothesis by comparing biologically motivated models of synaptic plasticity. A reinforcement learning STDP rule developed by Baras and Meir [4] was shown to exhibit statistically similar behavior to the BCM model. Similarly, Wade et al. [64] proposed a synaptic weight association training (SWAT) algorithm for spiking neural networks.

Computational neuro-genetic modeling (CNGM) [7] describes a class of neural models that take the genetic influence on neural dynamics into account. In their model [6], a gene regulatory network (GRN) is used to regulate the parameters of a classical spiking response model. A simple one-protein-one-neuronal function CNGM was shown to be able to reproduce the experimental data on the late long-term potentiation (L-LTP) in the rat hippocampal dentate gyrus [6].

III. BACKGROUNDS

A. BCM Spiking Neuron Models

Spiking neural networks are believed to be powerful computational models in that they can deal with temporal information processing more explicitly. The BCM based spiking neural model is of particular interest for studying neural plasticity as it has been shown that the BCM using a sliding threshold is the most accurate in predicting experimental data from synaptic-plasticity experiments [17, 37, 53]. This property makes the BCM model particularly attractive for learning temporal patterns.

For computational efficiency, we use a simplified, discrete-time version of the BCM-based spiking neural network [47], as shown in Fig. 1. The equations governing the behavior of synaptic weights of the spiking network shown in Fig. 1 can be described as follows:

$$w_i(t+1) = \eta x_i(t)\phi\big(y(t),\theta(t)\big) + \big(1-\varepsilon\big)w_i(t) \quad (1)$$

$$\phi\big(y(t),\theta(t)\big) = y(t)\big(y(t)-\theta(t)\big) \quad (2)$$

$$\theta(t) = \frac{\sum_{t'=t-h}^{t} y^2(t')\lambda^{t-t'}}{\sum_{t'=t-h}^{t} \lambda^{t-t'}} \quad (3)$$

where w_i denotes the weight of the i-th synapse. η is a constant learning rate. x_i is the pre-synaptic input of the i-th synapse. y is the post-synaptic output activity. Both x_i and y denote the spike trains in the spiking neural network. θ is a sliding modification threshold. $\phi(\cdot)$ is a non-linear activation function that swings with the sliding threshold θ, and can be defined by Eqn. (2). ε is a time-decay constant that is the same for all synapses. The interpretation of θ in Eqn. (3) clearly shows that the sliding threshold is a time-weighted average of the squared post-synaptic signals y within the time interval of h, where λ is a forgetting factor.

The neuron spiking behavior is conceptually simple. If the weighted sum of the pre-synaptic activation levels is less than θ, or if the neuron has fired in the previous time step, then the neuron activation level will decrease. Otherwise, the neuron will fire.

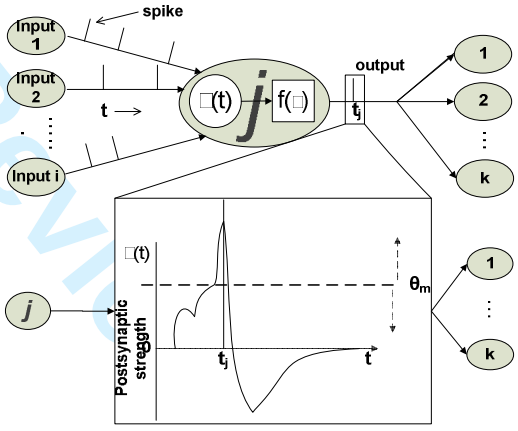

Fig. 1. The basic BCM-based spiking neural network model. θ_m is the modification threshold.

B. Gene Regulatory Network Models

When a gene is expressed, information stored in the genome is transcribed into mRNA and then translated into proteins [1].Some of these proteins are transcription factors that can regulate the expression of their own or other genes, thus resulting in a complex network of interacting genes termed as a gene regulatory network (GRN). A large number of computational models for GRNs have been developed [21].

Among others, ordinary or partial differential equations have been widely used to model regulatory networks. For example, Mjolsness et al. [48] proposed a GRN model for describing the gene expression data of developmental processes, where the dynamics of diffusion and cell-cell signaling has also been

taken into account:

$$\frac{dg_{ij}}{dt} = -\gamma_j g_{ij} + \phi\left[\sum_{l=1}^{n_g} W^{jl} g_{il} + \theta_j\right] + D_j \nabla^2 g_{ij}, \qquad (4)$$

where g_{ij} denotes the concentration of j-th gene product (protein) in the i-th cell. The first term on the right-hand side of Eqn. (4) represents the degradation of the protein at a rate of γ_j, the second term specifies the production of protein g_{ij}, and the last term describes protein diffusion at a rate of D_j. ϕ is an activation function for the protein production, which is usually defined as a sigmoid function $\phi(z) = 1/(1+\exp(-\mu z))$. The interaction between genes is described with an interaction matrix W^{jl}, the element of which can be either active (a positive value) or repressive (a negative value). θ_j is a threshold for activation of gene expression. n_g is the number of proteins.

C. Evolution Strategy with Covariance Matrix Adaptation

The evolution strategy with covariance matrix adaption (CMA-ES) [27, 28] is a stochastic, population-based search algorithm, which belongs to a class of evolutionary algorithms. In CMA-ES, the covariance matrix of the multivariate normal distribution used by the evolution strategy for mutation is adapted based on the search history. CMA-ES is well-known for its high efficiency and robustness for solving real-valued optimization problems. An up-to-date tutorial of CMA-ES can be found online [26].

IV. The Evolving GRN-BCM Model

A. The Whole Framework

The diagram of the evolving GRN-BCM model is shown in Fig. 2. The whole framework consists of different dynamics, i.e., the evolutionary search process, the gene regulation process and the neural dynamics. The dynamics of the BCM model can be described by Eqns. (1)-(3). The main challenge here is how to couple the neural dynamics with that of the gene regulatory network so that the plasticity parameters of the BCM model can be influenced by the activity of its neurons with the help of the gene regulatory network.

To this end, we make the following assumptions based on biological findings in gene regulated neural plasticity [50, 65]. First, we assume that the value of the plasticity parameters in the BCM model corresponds to an expression level of one gene, which directly determines the protein concentration that mediates the neural plasticity. Second, we assume that, through some chain of biochemical mechanisms, the concentration of the relevant proteins can be deduced from the sodium ion and/or calcium ion concentrations that reflects the neuronal activity level. Finally, we assume that the sodium ion concentration (represented by c_{Na^+}) is proportional to the input current, and the calcium ion concentration (represented by $c_{Ca^{2+}}$) is proportional to the activation threshold. Therefore, the inputs of the GRN that regulates the plasticity of the BCM consist of the sodium ion concentration c_{Na^+} and the calcium ion concentration $c_{Ca^{2+}}$.

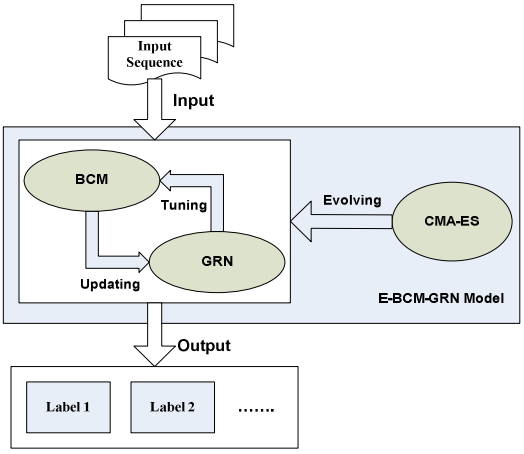

Fig. 2. The diagram of the evolving GRN-BCM model

The GRN that regulates neural plasticity can be described using a set of ordinary differential equations (ODE) similar to that in Eqn. (4). In the GRN model, several parameters need to be defined. In this work, we propose to employ the CMA-ES to evolve these parameters of the GRN model to optimize the performance of the entire model. As shown in Fig. 2, the CMA-ES will only be applied for optimization during the training phase, thus, it is an offline optimization. Once the parameters of the GRN model have been optimized during the training phase, these parameters will be fixed during the testing phase.

One important issue in coupling neural and gene regulatory dynamics is that neural dynamics and regulatory dynamics have different time scales. The timescale for electrical spike based neural dynamics is typically at the millisecond order. In contrast, gene regulation involves biochemical processes that can last seconds or even longer. Therefore, the neural dynamics should be updated at a higher frequency than that of the gene regulation dynamics. Since no theory can be used for choosing the timescales of the two systems, we conducted a few empirical studies to determine a ratio between the two timescales that is optimal to the behavior recognition performance. Refer to Section VI for more details.

B. The GRN Model

The gene expression levels that correspond to the three plasticity parameters (η, ϵ, λ) in the BCM model are defined as follows:

$$\eta(t+1) = (1-\gamma_\eta)\eta(t) + \alpha_\eta f\left(c_{Na^+}(t), c_{Ca^{2+}}(t)\right) \qquad (5)$$

$$\varepsilon(t+1) = (1-\gamma_\varepsilon)\varepsilon(t) + \alpha_\varepsilon g\left(c_{Na^+}(t), c_{Ca^{2+}}(t)\right) \qquad (6)$$

$$\lambda(t+1) = (1-\gamma_\lambda)\lambda(t) + \alpha_\lambda h\left(c_{Na^+}(t), c_{Ca^{2+}}(t)\right) \qquad (7)$$

Meanwhile, the protein expression levels for sodium ion and calcium ion concentration (c_{Na^+}, $c_{Ca^{2+}}$) are also influenced by the synaptic weights and weight plasticity of BCM model by

the following equations:

$$c_{Na^+}(t+1) = (1-\gamma_{c_{Na^+}})c_{Na^+}(t) + \alpha_{c_{Na^+}} \sum_i x_i(t) w_i(t) \quad (8)$$

$$c_{Ca^{2+}}(t+1) = (1-\gamma_{c_{Ca^{2+}}})c_{Ca^{2+}}(t) + \alpha_{c_{Ca^{2+}}} \theta(t) \quad (9)$$

where $\gamma_\eta, \gamma_\varepsilon, \gamma_\lambda, \gamma_{Na^+}$, and $\gamma_{Ca^{2+}}$ are decay factors and α_η, α_ε, α_λ, α_{Na^+}, and $\alpha_{Ca^{2+}}$ are coefficients. Functions $f(.)$,$g(.)$, and $h(.)$ are defined as the following sigmoid functions:

$$f(c_{Na^+}, c_{Ca^{2+}}) = \left(1 + e^{-\left(k_1 c_{Na^+} + k_2 c_{Ca^{2+}}\right)}\right)^{-1} \quad (10)$$

$$g(c_{Na^+}, c_{Ca^{2+}}) = \left(1 + e^{-\left(k_3 c_{Na^+} + k_4 c_{Ca^{2+}}\right)}\right)^{-1} \quad (11)$$

$$h(c_{Na^+}, c_{Ca^{2+}}) = \left(1 + e^{-\left(k_5 c_{Na^+} + k_6 c_{Ca^{2+}}\right)}\right)^{-1} \quad (12)$$

where k_1, k_2, k_3, k_4, k_5, and k_6 are coefficients.

C. Evolving the GRN-BCM Model

In this GRN-BCM model, 16 parameters in total, namely, $\gamma_\eta, \gamma_\varepsilon, \gamma_\lambda, \gamma_{Na^+}, \gamma_{Ca^{2+}}$, $\alpha_\eta, \alpha_\varepsilon, \alpha_\lambda, \alpha_{Na^+}, \alpha_{Ca^{2+}}$, k_1, k_2, k_3, k_4, k_5, and k_6 need to be defined, whose optimal value may be problem-specific. Fine tuning these parameters manually for optimal performance is tedious and time-consuming. For this reason, we use CMA-ES to optimize these parameters in this work.

In CMA-ES, k offspring are generated from μ parents ($\mu \le k$) by mutating the current best solution with a random number sampled from a normal distribution:

$$x_i^{(g+1)} \sim m^{(g)} + \sigma^{(g)} N_i(0, C^{(g)}) \quad \text{for } i = 1, \ldots, k \quad (13)$$

where "~" denotes the same distribution on the left and right side. $x_i^{(g+1)}$ is the i-th candidate solution for generation $g+1$, and x is a vector $x \in R^n$, which consists of five gamma $(\gamma_\eta, \gamma_\varepsilon, \gamma_\lambda, \gamma_{Na^+}, \gamma_{Ca^{2+}})$, five alpha $(\alpha_\eta, \alpha_\varepsilon, \alpha_\lambda, \alpha_{Na^+}, \alpha_{Ca^{2+}})$, and k_1, k_2, k_3, k_4, k_5, and k_6. The vector $m^{(g)} \in R^n$ represents the favorite solution at generation g, which is the mean of the k parents at generation g. $\sigma^{(g)}$ is the step-size which controls the severity of mutations at g. $C^{(g)}$ denotes the covariance matrix at generation g and $N_i(0, C^{(g)})$ is the i-th multivariate normal distribution with zero mean and covariance matrix $C^{(g)}$. Both the step-size and the covariance matrix are subject to adaptation based on the search history.

A fitness value that reflects the performance of the solution will be assigned to each of the k offspring. Here, the fitness value is defined as the classification error on the training data between the actual output and the desired output. μ offspring will be selected from k offspring based on their fitness values and passed to the next generation as parent individuals. This process continues until a stopping condition is met.

V. Empirical Study of GRN-BCM on Synthetic Data

To verify the convergence property of the GRN-BCM model trained using the CMA-ES and the computational power of the GRN-BCM for temporal information process, we first employed the GRN-BCM model for the exclusive-or (XOR) classification problem. Then, the GRN-BCM model is compared to the standard BCM spiking neural network, a HMM and a reservoir computing model on a synthetic time series classification problem.

A. Convergence of the Evolving GRN-BCM Model

To empirically verify if the proposed model can converge in learning, we apply the proposed model to solving the XOR classification problem that has been widely studied in training spiking neural networks. For simplicity, the spiking neural network consists of 2 input neurons, 5 hidden neurons and 1 output neuron. Similarly to [9], the input and output values are encoded by time delays, associating the analog values with the corresponding "earlier" or "later" firing times. The input neurons with a higher value "1" is given an early spike time, while a lower value "0" with a later spike time. The firing times for encoding the XOR problem are presented in Table I. The numbers in the table represent the spike times in one millisecond. For example, 0 and 6 represent the respective input spike times. The two output classes are represented by a spiking time of 8 and 15 milliseconds, respectively. As described in the previous sections, the synaptic plasticity of spiking neurons is regulated by the GRN, and CMA-ES is used to evolve the parameters of the GRN-BCM model. Fig. 3 illustrates the mean squared errors obtained on the XOR problem. We can see from Fig. 3 that the whole model converges gracefully within approximately 400 generations.

TABLE I: Encoding of Xor Problem

Input Pattern		Output Pattern
0	0	15
0	6	8
6	0	8
6	6	15

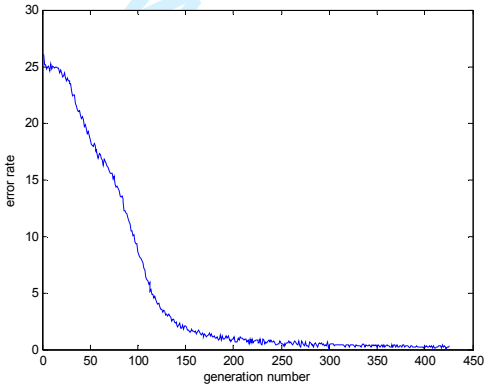

Fig. 3. Convergence of the XOR problem.

B. GRN-BCM for Time Series Classification

To demonstrate the strength of the proposed GRN-BCM model compared to alternative models which can also handle time-series data, such as the traditional BCM model, reservoir

computing models, and hidden Markov model (HMM), we conducted some experiments on a synthetic data. The time-series data is similar to the one used in [32] and is generated by triple interleaving generators, i.e., tent map, sine wave, and random constant value. One generator produces the chaotic iterated tent map [12], the second one produces a constant period sine wave ($0.5\sin(0.8t)+0.5$) within [0, 1], and the last one generates a random value between 0 and 1.

A one-dimensional signal $\tilde{x}(t)$ of length 6000 is generated by randomly switching between three different generators, in which 3000 data are used for training and the rest 3000 for testing. At each time step, the current generator is switched to another one with a probability of 0.05. Then, every 6 consecutive one-dimensional signals are coded into one sequence as the input using the following triangular membership functions:

$$x(i) = Tri(5\tilde{x}(t) - t + 1) \quad (t = 1, \cdots, 6)$$
$$Tri : u \mapsto \begin{cases} 1, & \text{if } abs(u) > 1 \\ 0, & \text{else} \end{cases} \tag{14}$$

Where $Tri : u$ means that u is a parameter of the function Tri.

Since the HMM model is not suited for the time series prediction problem, we have converted the time series prediction problem in [32] into a time series classification problem. To this end, the output of the time series is categorized into four patterns as follows:

$$y_1(i) = \begin{cases} 1, & \text{if } \sum_{t=1}^{3} abs(\tilde{x}(t))/3 > 1 \\ 0, & \text{else} \end{cases}$$
$$y_2(i) = \begin{cases} 1, & \text{if } \sum_{t=4}^{6} abs(\tilde{x}(t))/3 > 1 \\ 0, & \text{else} \end{cases} \tag{15}$$

The proposed GRN-BCM model, a traditional BCM spiking neural network, a LSM model, and a hidden Markov model (HMM) are applied to this synthetic time series classification problem. Both the proposed GRN-BCM model and the traditional BCM model adopt spiking neural networks, which consist of one input, 15 hidden neurons and two outputs. The BCM model is defined as in Eqns. (1)-(3), where three plasticity parameters (η, ϵ, λ) are directly evolved by the CMA-ES without being regulated by the GRN.

Since finding an optimal setup of the parameters and structure of LSM is itself a challenging research topic, we adopted the parameterization suggested in [45], with which very good performance has been reported. The reservoir consists of 60 internal neurons, and the probability of an existing connection between two neurons a and $b$in the reservoir is defined by $P(a,b) = C \cdot \exp(-D^2(a,b)/\lambda^2)$, where $D(a,b)$ is the Euclidean distance between neurons a and b, $C = 0.3$ and $\lambda = 2$ control the reservoir connectivity and dynamics. A simple linear projection is adopted as the readout function that linearly maps the states into the outputs. In addition, a supervised learning algorithm, called ridge regression, is applied to adapt the readout weights of the LSM model in an offline manner.

For the HMM model, we adopt the first-order HMM, which has three hidden states and six possible observations for each time-series sequence. Specifically, each state is assigned a probability of distribution from one state to another (i.e., state transition probability) and each observation is also assigned a probability at states (i.e., observation symbol probability). In general, the HMM parameters (i.e., state transition and observation symbol probabilities) are estimated by the expectation-maximization (EM) method[5, 14]. Moreover, the Viterbi algorithm [62] is used to find the most probable sequence of hidden states called Viterbi path for recognition.

The comparison results from the compared models on the synthetic data are listed in Table II. Both training and testing errors are listed Table II. From Table II, we can see that the GRN-BCM model outperforms other models for the considered time series classification problem.

TABLE II: COMPARATIVE RESULTS FOR SYNTHETIC DATA

Methods	Training Error	Testing Error
GRN-BCM	0.6%	2%
BCM	3.8%	7.8%
LSM	3.2%	6%
HMM	8.2%	11.4%

In the following, we discuss briefly the computational complexity of the compared models. The computational complexity of the GRN-BCM model is $O((I+H+J)NT)$, where J is the number of inputs, H is the time interval (Eqn. 3), I is the number of pre-synapses, N is the number of neurons and T is the length of input sample. Since the traditional BCM model does not take the mechanism of GRN into account, the complexity for BCM is $O((I+H)NT)$. LSM consists of a random fixed reservoir which is a recurrent structure. The complexity of the LSM is $O(N^2T)$. In general, the Viterbi algorithm is used for recognition when applying HMM. The complexity of HMM is based on the Viterbi algorithm, i.e., $O(S^2T)$, where S denotes the number of states. In general, the computational cost of GRN-BCM is slightly higher than that of the traditional BCM, and lower than that of LSM model. Meanwhile, depending on the number of states used in the HMM model, the computational cost of the GRN-BCM model may or may not be higher than that of HMM model, but the performance of the GRN-BCM model is much higher than the HMM model (from Table II).

VI. GRN-BCM MODEL FOR HUMAN BEHAVIOR RECOGNITION

A. The System Framework

To evaluate the learning performance of the GRN-BCM model on real-world applications, human behavior recognition is adopted for case study since extracting temporal patterns is critical for the success of human behavior recognition. The system framework for human behaviors recognition from sequences of visual data using the evolving GRN-BCM based spiking neural network model is shown in Fig. 4.

When a new video arrives, it often consists of a sequence of image frames. Each frame will be preprocessed to extract spatial features. The details of how to extract spatial features will be discussed in Section VI. B. We assume that there are three layers in the GRN-BCM model, which consists of middle layer 1, middle layer 2 and the label layer. Given each image frame with a size of $H \times W$, a sequence of frames with the extracted spatial features are fed into the middle layer 1 as the inputs. The temporal relationships between the extracted spatial features are encoded in the sequence of frames, which will be explicitly taken into account by the three-layer GRN-BCM model, together with the extracted spatial features on each frame. In this way, the behavior pattern can be recognized by the proposed model. Please be noted that we do not have explicit model for the behavior representation in this system because both the spatial and temporal patterns are handled within the whole system coherently. The details of how to construct this SNN and how to learn the temporal information using the proposed GRN-BCM model will be discussed in Section VI.C and Section VI.D, respectively.

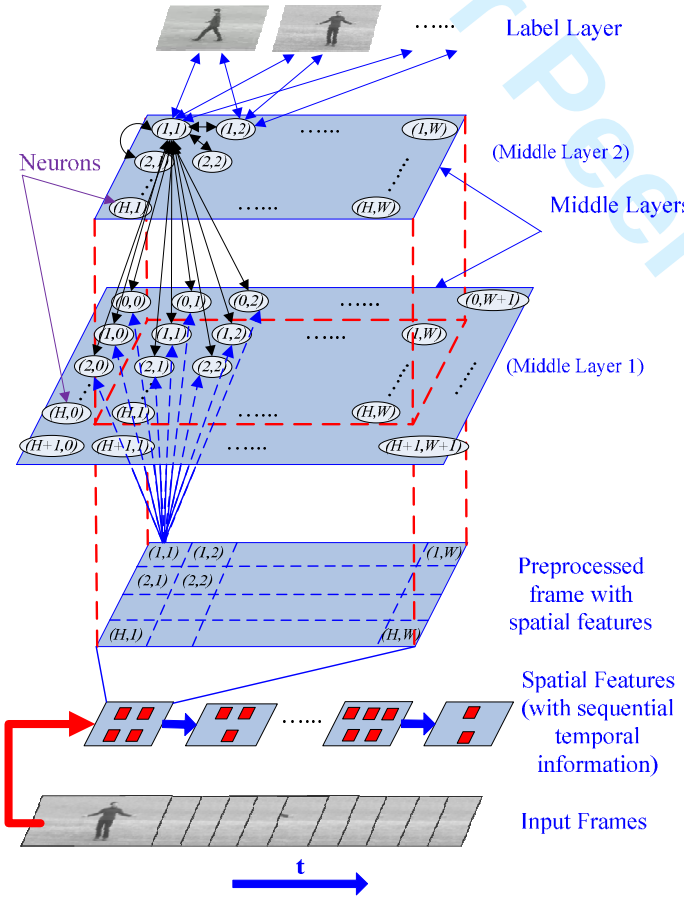

Fig. 4. The system framework using the GRN-BCM based SNN model as the classifier for visual human behavior recognition.

B. *Spatial Feature Extraction*

Generally, each video sequence consists of a set of sequential frames, which is the input of the bottom layer as shown in Fig. 4. 2D interest point detectors are adopted for detecting spatial features, which model the geometric relationships between different parts of the human body. These spatial features of the co-occurrence of motion pixels represent the low-level visual features, which are fairly reliable in a video sequence.

One of the most popular approaches to interest spatial point detection is based on the detection of corners. In this paper, corners are defined as regions where the local gradient vectors are in orthogonal directions. The gradient value at pixel $(i, j)$is denoted by $L(i, j, \sigma)$ and can be obtained by taking the first order derivative of a smoothed image as follows:

$$L(i, j, \sigma) = I(i, j) * g(i, j, \sigma) , \qquad (16)$$

where $I(i, j)$ denotes the pixel value at position (i, j) in an input image, $g(i, j, \sigma)$ is the Gaussian smoothing kernel, and σ controls the spatial scale where corners are detected. The gradient difference between the previous frame and the current frame is denoted by $R(i, j, \sigma) = (L_{cur}(i, j, \sigma) - L_{pre}(i, j, \sigma))^2$, where $L_{cur}(i, j, \sigma)$ and $L_{pre}(i, j, \sigma)$ denote the gradient values of the current frame and the previous frame, respectively.

To identify the spatial features based on this gradient difference value, we define a relative threshold. For example, we only need to extract 20 spatial features for each frame. Therefore, the pixels whose gradient difference value is within the highest top 20 will be treated as the location of the spatial features. Then, based on the locations of the extracted spatial features, we convert the original input frames into binary images as the inputs of the GRN-BCM spiking neural network, where all the pixel values of the extracted spatial features are set as 1 and the rest as 0 in the binary images for each frame. Fig. 5 shows one example of extracting the spatial features of "hand-waving" behavior from a video sequence, where the red shaded areas represent the extracted spatial features.

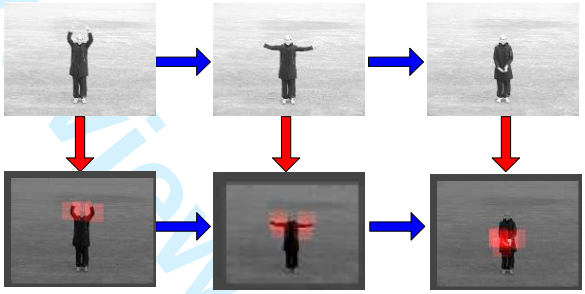

Fig. 5. An example of extracting the spatial features of a "hand-waving" behavior. The bottom row shows the detected spatial features (represented by red blocks) from original frames in the video.

After collecting the spatial features, the goal of our model is to classify a testing video sequence into one of the trained patterns from the training set. If all the activities performed in the training set are in the middle of the video frames, even if the activity in a testing video is not performed in the middle of the frames, the corresponding pattern can still be recognized since the most similar pattern to the current video will be selected from the trained patterns. As we mentioned in Section VI.A, there is no explicit model to represent the behavior patterns because both the spatial and temporal patterns of the behaviors are handled coherently by the whole system, as shown in Fig. 4.

C. The Spiking Neural Network

For computational efficiency, it is assumed that each neuron is only connected to its local neighboring neurons within and across the layers. Here, we elaborate how the neurons in the three-layer SNN are connected. Let $S^{(1)}_{(h_1,w_1)}$ ($h_1 = 0, 1, \dots, H+1$; $w_1 = 0, 1, \dots, W+1$) and $S^{(2)}_{(h_2,w_2)}$ ($h_2 = 1, \dots, H$; $w_2 = 1, \dots, W$) denote the neurons in middle layer 1 at location (h_1, w_1) and middle layer 2 at location (h_2, w_2), respectively. The neuron $S^{(2)}_{(h_2,w_2)}$ can connect up to 8 intra-layer local neighboring neurons, i.e., $S^{(2)}_{(h_2-1,w_2-1)}$, $S^{(2)}_{(h_2-1,w_2)}$, $S^{(2)}_{(h_2-1,w_2+1)}$, $S^{(2)}_{(h_2,w_2-1)}$, $S^{(2)}_{(h_2,w_2+1)}$, $S^{(2)}_{(h_2+1,w_2)}$, $S^{(2)}_{(h_2+1,w_2)}$ and $S^{(2)}_{(h_2+1,w_2+1)}$ within middle layer 2, and up to 9 inter-layer neighboring neurons, i.e., $S^{(1)}_{(h_2-1,w_2-1)}$, $S^{(1)}_{(h_2-1,w_2)}$, $S^{(1)}_{(h_2-1,w_2+1)}$, $S^{(1)}_{(h_2,w_2-1)}$, $S^{(1)}_{(h_2,w_2)}$, $S^{(1)}_{(h_2,w_2+1)}$, $S^{(1)}_{(h_2+1,w_2)}$, $S^{(1)}_{(h_2+1,w_2)}$ and $S^{(1)}_{(h_2+1,w_2+1)}$ in middle layer 1 in a bidirectional way. It is noted that only the neurons in two middle layers can have intra-layer connections. In our model, all the connections between neurons within the SNN are bidirectional, except for the ones from the input layer to the middle layer which are unidirectional.

Based on extracted spatial features, we convert the original input frames into binary images as the inputs of network, where each extracted feature is represented with the pixel value of 1 in the binary image, and the rest is represented with the pixel value of 0. As a result, the extracted features in each frame correspond to spikes fed into the middle layer 1 of SNN. Due to the assumption that connections only exist within local neighbors within each layer, the features at the border of the image have fewer connections than those in the center. To maximally deliver the input information into the network, the size of middle layer 1 is set to be $(H+2)\times(W+2)$. The size of middle layer 2 is set to be $H \times W$, which is the same as the input layer. The top layer is the label layer and is fully connected in a bidirectional way to the neurons at middle layer 2 (Fig.4). When a new video sequence comes, the strongest strength will be generated on one label neuron that corresponds to the given behavior at the label layer. In other words, each label neuron represents one type of human behavior pattern.

Here, each video sequence will be fed into the network frame by frame sequentially. Specifically, each frame with spatial features is discretely considered as one individual time step described in Eqns. (1)-(9). For example, if the frame frequency of a real-time video sequence is 30 frame/second, then the time step in Eqns. (1), (5), (6), (7), (8) is (1/30) seconds.

Given the input of a preprocessed frame with the size of $H \times W$ pixels, each pixel on the image frame is associated with an adjustable weight. Let $\omega_{i,j}$ denotes the weight from the input $x_{i,j}$ to the neuron $S^{(1)}_{(u,v)}$ at middle layer 1. Here, the input $x_{i,j}$ includes all the local neighboring neurons from the input layer (using pixel values), middle layer 2and middle layer 1 (using the membrane potentials, which is termed neuron values thereafter). Therefore, the neuron value of $S^{(1)}_{(u,v)}$ at middle layer 1 is denoted by $y^{(1)}_{u,v}$ and can be calculated by

$$y^{(1)}_{u,v} = f\left(\sum_{(i,j)\in R^{(1)}_{(u,v)}} \omega_{i,j} x_{i,j} \right), \tag{17}$$

where $f(.)$ denotes the activation function, which is defined as the same as in Eqn. (10), and $R^{(1)}_{(u,v)}$ is the receptive field of neuron $S^{(1)}_{(u,v)}$ which includes all the local neighboring neurons(from the input layer, middle layer 1 and 2) that are connected to neuron $S^{(1)}_{(u,v)}$. The neuron value of $S^{(2)}_{(u,v)}$ at middle layer 2 is denoted by $y^{(2)}_{u,v}$, which can be calculated by

$$y^{(2)}_{u,v} = f\left(\sum_{(i,j)\in R^{(2)}_{u,v}} \omega_{i,j} y^{(1)}_{i,j} \right), \tag{18}$$

where $R^{(2)}_{(u,v)}$ contains all the input neurons to neuron $S^{(2)}_{(u,v)}$ with the same rule as $R^{(1)}_{(u,v)}$. $y^{(2)}_{u,v}$ is rearranged into a column vector, and used as the input to the label layer as follows:

$$\{y^{(2)}_{u,v} \mid u = 1,\cdots,H_2; v = 1,\cdots,W_2\} \rightarrow \{y^{(2)}_m \mid m = 1,\cdots,H_2 \times W_2\} \tag{19}$$

For the label layer, let $\omega_{m,n}$ denotes the synaptic weight from neuron m in middle layer 2 to neuron n in the label layer. The neuron value of the label neuron at the label layer is denoted by y_n which can be calculated by

$$y_n = f\left(\sum_{m=1}^{H_2\times W_2} \omega_{m,n} y^{(2)}_m \right), \tag{20}$$

where each label represents one type of the human behaviors.

D. Supervised Learning using the GRN-BCM Model

A sequence of frames with the extracted spatial features on each frame is fed into the GRN-BCM model as input. Now the GRN-BCM needs to extract the temporal relationship of the extracted spatial features between frames. The major advantage of the GRN-BCM is that the temporal information is naturally embedded into the model through dynamic plasticity parameters of the GRN-BCM model. Therefore, the temporal feature extraction can be spared in the feature extraction phase, which considerably reduces the computational complexity and thus improves real-time performance.

To further improve the convergence speed of the GRN-BCM for video processing, a scalable training algorithm is adopted. The computational cost for neural network training depends heavily on the number of neurons in the network. In other words, the frame size in video directly determines the computational complexity. To reduce the computational cost while maintaining satisfactory performance, a scalable training algorithm is developed, as sketched in Fig. 6. We begin with the training set, defined at a higher resolution (in this case, 32×32). Each exemplar is coarsened by a factor of two in each direction using a simple grey scale averaging procedure. 2×2 blocks of pixels where all four pixels are mapped to a pixel, and those where three of the four are mapped to a 3/4 pixel, and so

on. In this way, each 32×32 exemplar is mapped to a 16×16 exemplar to preserve the large scale features of the pattern. The procedure is repeated until a suitable coarse representation of the exemplars (8×8) is reached.

The training begins with 8×8 pixels exemplars. These input vectors are fed into the neural network for training. After being trained for a predefined number of iterations, the network is "boosted" by manipulating the weights between the first and second layers. Each weight from the lower layer to the upper layer is split into four weights, where each one is 1/4 of the original size. The resultant network is trained for a few additional iterations by feeding in 16×16 pixels exemplars. Again, the trained network is boosted for the next training session. Similarly, the boosted network is fed in with 32×32 pixels exemplars for a number of iterations until convergence of the training error is achieved. The boosting and training process is repeated until the desired network is achieved.

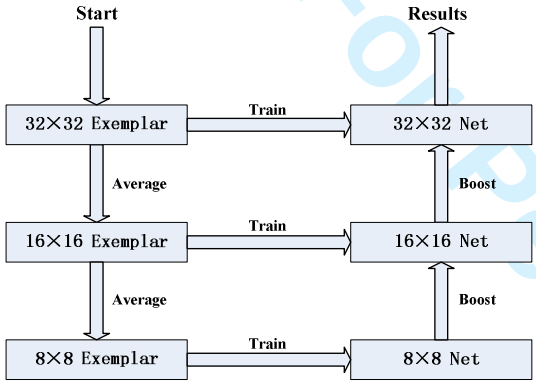

Fig. 6. A flowchart of the scalable training algorithm

In the supervised learning phase, the GRN model is used to regulate three plasticity parameters, i.e., (η, ϵ, λ) of the BCM-SNN model to learn specific behavior patterns using the training pairs of input video frames and output labels. More specifically, each label neuron will produce a response with respect to a new video sequence and the one having the maximum response activity corresponds to the behavioral pattern to which the new video belongs. So the architecture of the SNN must be constructed in a layered feed-forward way. To make it clear, the pseudo code of the learning process using the GRN-BCM model is given as follows:

Create the initial population 20 individuals each encoding 16 parameters of the GRN model, namely, $\gamma_\eta, \gamma_\varepsilon, \gamma_\lambda, \gamma_{Na^+}, \gamma_{Ca^{2+}}$, $\alpha_\eta, \alpha_\varepsilon, \alpha_\lambda, \alpha_{Na^+}, \alpha_{Ca^{2+}}$, and k_1, k_2, k_3, k_4, k_5 and k_6
For generation $l = 1$ to L
For individual $k = 1$ to K
For i = 1 to M (M is number of training samples, i.e. video sequences) *do*
For j =1 to T (T is number of fames in each sample) *do*
Update GRN model including sodium and calcium ion concentration (c_{Na^+}, $c_{Ca^{2+}}$) according to Eqns. (8) , (9), and three plasticity parameters (η, ϵ, λ) of BCM according to Eqns. (5)- (7);
Update w , $\phi(\cdot)$ and θ of BCM according to Eqns. (1)-(3);
Calculate the outputs y_n of the BCM according to Eqns. (17), (18) and (20);
End
End
End
Perform mutation, recombination and selection according to CMA-ES
End
The parameters encoded in the best solution of the final generation are adopted for the GRN-BCM model to be used for testing

Please be noted that this is an offline learning process. During the testing phase, the synaptic weights and weight plasticity of the BCM model are dynamically tuned by the GRN model with fixed parameters evolved using the CMA-ES during the training phase.

E. Computational Complexity of the Supervised Learning using the GRN-BCM Model

In this section, we analyze computational complexity of this learning algorithm. First, one epoch of the training process is defined as the time period needed to process one video sequence, which consists of a number of frames and represents one type of behaviors. For each epoch in the training phase, only one video sequence is fed into the network, and then the CMA-ES is applied to optimize the parameters for the GRN model. We have observed that the evolutionary search of the parameters will converge within a few generations.

In the following, we discuss the computational complexity of the proposed framework in one generation. Basically, five main factors need to be considered: the number of spiking neurons (i.e., the frame size), the number of time steps needed for training (i.e., the number of fames in a video sequence), the number of training samples, the dimension of the objective parameters and the population size (i.e., candidate solutions) in CMA-ES. As mentioned in previous section, there are $P = 16$ parameters need to be evolved and $K = 20$ individuals in the CMA-ES are used in our model. Assuming the number of training sample is M, the average number of time steps for each training sample is T , and the number of input spiking neurons is N, then the average computational complexity of the learning algorithm within one generation is $O(K(MNT + P^2))$. If the CMA-ES converges in L generations, the overall computational complexity for the training the GRN-BCM model is $O(LK(MNT + P^2))$.

For instance, 450 video sequences are picked from the KTH dataset as training samples, where each video sequence lasts about 18 seconds and the frame rate is 25 frames per second. Running the E-GRN-BCM on a PC with a CPU of the Intel Core 2 Duo Processor P8600 (2.4GHz, 4GB memory), the wall clock time for training is roughly672 seconds for one generation. If the evolutionary training converges within 10 generations (which is typically observed), the time for off-line

training of the GRN-BCM model is about 2 hours in total.

VII. Results on Human Behavior Recognition

A. Configuration of the BCM-based Spiking Neural Network

To evaluate the performance of the proposed GRN-BCM based SNN model, several experiments on different scenarios have been conducted on the KTH datasets (http://www.nada.kth.se/cvap/actions/) and Weizmann human motion datasets [68]. The effectiveness of the framework is measured by two criteria: recognition accuracy and robustness of the recognition performance in the presence of uncertainties in the images. In the training process, 4-fold cross-validation is applied to test the efficiency of our model in behavior recognition, where the original video sequences are randomly partitioned into 4 subsamples. For each run, 3/4 videos are used for training and the rest are used for test and validation. The cross-validation process is repeated for four times so that each of the subsamples is used once as the validation data.

For these two datasets, the video is usually captured in a resolution of 320x240 pixels per frame. If we use the pixel-level input to the spiking neural network, the overall computational cost for evolutionary optimization would be very expensive due to the large number of neurons. Therefore, as shown in Fig. 6, in the features layer, the inputs are reduced to 32x24 in size, which is proportional to the original pixel-level inputs. That is, the size of the input layer in Fig. 4is 320x240, and the size of the features layer is 32x24. As a result, middle layer 1 is composed of 34x26 neurons, and middle layer2 is composed of 32x24 neurons. The label layer is fully connected to middle layer 2.Since CMA-ES does not require a large population, we set $k = 20$ in the experiments. The initial step-size of the CMA-ES is set to $\sigma^{(g)} = 10$.

B. Evaluation of Multiple Time Scales

As we discussed in the previous section, the timescale of neural dynamics is much faster than that of the gene regulation dynamics. To determine the ratio between the two time scales, we conduct a set of experiments with the ratio of 1:1, 1:5, 1:10 and 1:15, respectively. The parameters $(\gamma_\eta, \gamma_\varepsilon, \gamma_\lambda)$ from one neuron are drawn for each different ratio, as shown in Fig. 7.

It can be seen from Fig. 7 that, although the ratios are various from 1:1 to 1:15, no significant difference in the dynamics of the parameters can be observed. In other words, different ratios in timescale between the SNN and the GRN have no significant influence on the performance of the proposed model. However, the neural dynamics stabilizes a bit faster at the ratio of 1:5 than in other cases. It is noted that the computational cost is also reduced compared to the case where the same timescale is adopted for neural and regulatory dynamics.

It is worthy of pointing out that although the values of parameters for different neurons may be different, overall behaviors of these parameters from each neuron is just slightly different given different time scales. Fig. 7 only shows the parameter behaviors of one neuron (randomly selected) to demonstrate whether different ratios in timescale will affect the system performance of the proposed method.

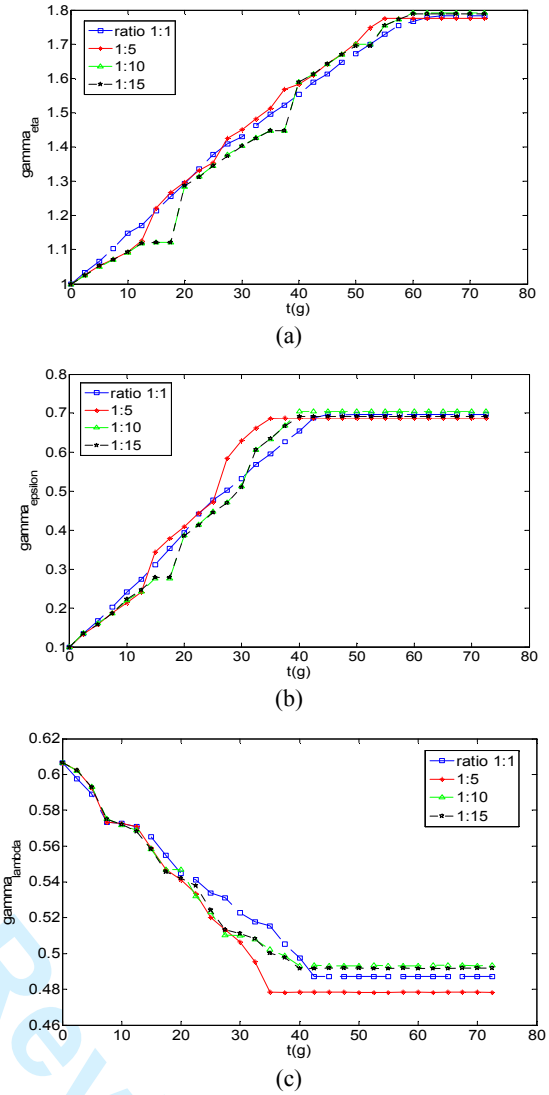

Fig. 7. The profile of parameters of $(\gamma_\eta, \gamma_\varepsilon, \gamma_\lambda)$ from one neuron arbitrarily chosen for four different ratios. (a) γ_η, (b) γ_ε, and (c) γ_λ. t(g) denotes the time with the unit of generation.

C. Behavior Recognition on KTH Dataset

The KTH video database contains image data in which six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) are repeatedly performed by 25 participants in four different scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4), as shown in Fig. 8. We test 599 video sequences of different participants. Fig. 8 demonstrates the extracted spatial features from the original frames for three types of behaviors, i.e., walking, hand waving, and boxing.

The recognition rate is defined as the percentage of correctly recognized behaviors from the number of all samples, which is

$$R = \frac{N_R}{N} \times 100\% \qquad (21)$$

where R is the recognition rate, N is the number of input behaviors, and N_R is the number of correctly recognized behaviors. Table III shows the recognition results. From Table III, we can see that the behavior recognition performance of the

GRN-BCM model on the KTH dataset is satisfactory with an overall recognition rate of 84.81%. This indicates that our proposed model is effective for human behavior recognition.

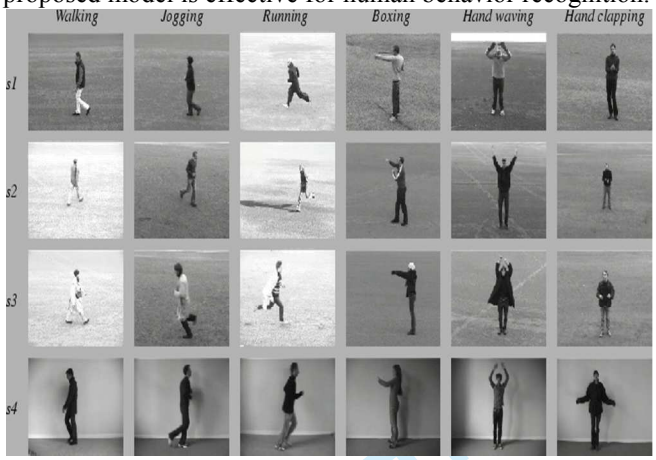

Fig. 8. Example images from video sequences in KTH dataset corresponding to different types of actions and scenarios.

TABLE III: BEHAVIOR RECOGNITION RESULTS

Behaviors	N	N_R	R(%)
Walking	100	85	85
Jogging	99	82	82.8
Running	100	82	82
Hand-waving	100	87	87
Boxing	100	85	85
Hand-clapping	100	87	87
Overall	**599**	**508**	**84.81**

TABLE IV: CONFUSION MATRIX FOR THE KTH DATASET

	Walk	Jog	Run	Hand-wave	Box	Hand-clap
Walk	0.85	0.09	0.06	0	0	0
Jog	0.06	0.83	0.12	0	0	0
Run	0	0.18	0.82	0	0	0
Hand-wave	0	0	0	0.87	0	0.13
Box	0	0	0	0.02	0.85	0.13
Hand-clap	0	0	0	0.07	0.06	0.87

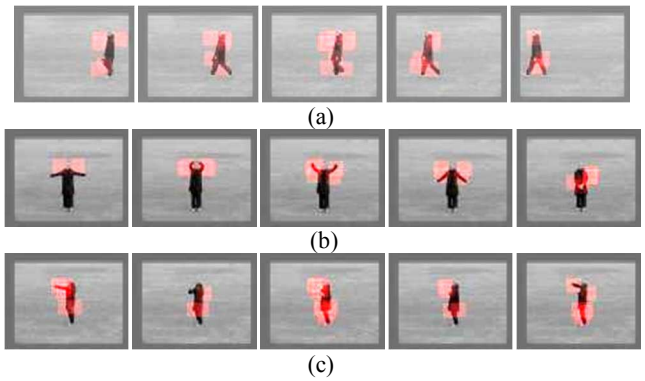

Fig. 9. In the KTH dataset, the original video sequences for behavior patterns of "walking", "hand-waving" and "boxing" at top rows, and the extracted spatial features (represented by the red blocks in the images) from these behavior patterns at the bottom rows. (a) "walking" behavior, (b) "hand-waving" behavior, (c) "boxing" behavior.

To figure out why some action sequences are misclassified, a confusion matrix with respect to visual inputs is calculated and listed in Table IV. The elements of each row in the confusion matrix represent the misclassification rate. From the confusion matrix in Table IV, we can see that confusions commonly occurs for similar behaviors, for example, "running" and "jogging", which is reasonable since running sometimes may appear to be jogging and vice versa, even judged by a human user.

To evaluate the performance of the proposed model, first we compare it with the traditional BCM model by using the same spatial features extracted from the videos. In the traditional BCM model, the structure of the SNN is the same as that in the GRN-BCM model (as shown in Fig. 4), and the BCM model is described by Eqns. (1)-(3) where three plasticity parameters (η, ϵ, λ)are directly evolved by the CMA-ES without being regulated by the GRN. In addition, we also compare the performance of the GRN-BCM model with a liquid state machine model (LSM) using the same spatial features. In the LSM model we adopted here, the reservoir is built from 32x24x2 spiking neurons with fixed random connections. We adopted the same LSM model used for the synthetic time series classification problem described in Section V.B. Please refer to Section V.B for the detailed parameter setup for the LSM model. For computational efficiency, we stick to the first-order HMM, which has five hidden states. The experimental results are listed in Table V. Obviously, with the same spatial features, the GRN-BCM model outperforms the traditional BCM, LSM, and HMM models. Moreover, the low recognition rate of the traditional BCM-SNN clearly indicates that GRN regulated neural plasticity significantly improve the computational power of spiking neural networks. It can also be seen that the HMM model exhibited the worst performance (an accuracy of 43.07%), which is consistent with the experimental results on the synthetic dataset (as shown in Table II).

TABLE V: Comparison of Different Classifiers on KTH Dataset

Methods	Features	Accuracy (%)
GRN-BCM	Spatial features	84.81
BCM	Spatial features	65.28
LSM	Spatial features	67.11
HMM	Spatial features	43.07

To provide a more complete evaluation of the proposed method, a few other state-of-the-art algorithms for human behavior detection [2, 15, 35, 56] have also been adopted for comparison. The results are listed in Table V, where both the feature extraction methods and classification methods in each approach have been listed as well. Note that all the compared methods use complex and sophisticated spatiotemporal features during the feature extraction phase except for our model, in which only spatial features are used.

TABLE VI: Comparison of Different Methods on KTH Dataset

Methods	Features	Classifier	Accuracy (%)
GRN-BCM	Spatial features	GRN-BCM	84.81
Schuldt et al. [56]	Spatiotemporal interesting points	SVM	71.83
Ke et al. [35]	Optical flow	Boosting	62.97
Dollaret al. [15]	Gabor filters	1-NN	81.17
Antonios et al. [2]	B-splines	Gentleboost + RVM	80.8

From Table VI, it can be seen that the proposed GRN-BCM model posses the highest average recognition rate of 84.81% among the compared methods even with the much spatial features only (which are easier to extract), suggesting that the proposed GRN-BCM model is more efficient and powerful than the compared methods for the human behavior recognition from video sequences.

The main reason that we did not apply spatial features for the state-of-the-art methods in comparison is that in the methods listed in Table VI, the classifiers are developed based on the spatial domain and cannot encode the temporal information directly. As a result, these methods cannot operate on video sequences only relying on the spatial features. To address this issue, in these methods the temporal information involved in the video data must be defined as the third dimension with respect to two spatial dimensions, and implicitly described as "special" spatial features. Therefore, those classifiers need to be provided with spatiotemporal features as the inputs for the human behavior recognition. From this point of view, the proposed GRN-BCM model has a clear advantage over the compared ones since it can considerably reduce the computational complexity for video processing for the whole system.

D. Behavior Recognition on Weizmann Dataset

Then we evaluate the performance of the GRN-BCM model on another commonly used dataset, Weizmann human action dataset [68]. The corresponding video material can be found at http: // www. wisdom. weizmann .ac. il /~vision/ SpaceTimeActios. html. This database contains 90 low-resolution video sequences showing 9 participants, each performing 10 natural actions. Fig. 10 shows some examples of these 10 behaviors from this database. The individual recognition rates of all these 10 behaviors and the overall average recognition rate using the GRN-BCM model are presented in Table VII.

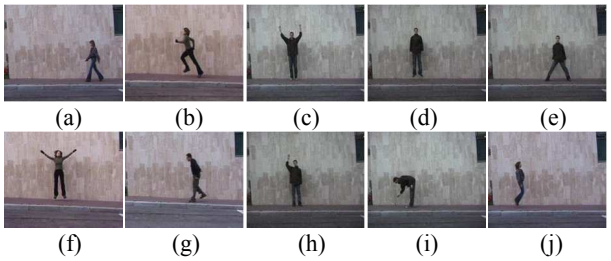

Fig. 10. The example images from different behavior video sequences in Weizmann dataset. (a) walking, (b) running, (c) waving2, (d) pjumping, (e) siding, (f) jack, (g) skipping, (h) waving1, (i) bending, and (j) jumping.

From Table VII, it can be seen that the recognition rates are lower than those on KTH dataset, dropping from 84.81% to 74.44%. The reason might be that the Weizmann dataset has much fewer training and testing samples compared to the KTH database. In the KTH dataset, there are about 100 testing samples for each behavior, while Weizmann dataset only has about 9 test samples for each behavior although it has more categories of behaviors. Therefore, it becomes harder for the models to achieve a high recognition rate.

TABLE VII: BEHAVIOR RECOGNITION RESULTS

Behaviors	**N**	**N_R**	**R (%)**
Walking	9	7	77.78
Running	9	7	77.78
Waving2	9	7	77.78
Pjumping	9	6	66.67
Siding	9	7	77.78
Jack	9	6	66.67
Skiping	9	5	55.56
Waving1	9	7	77.78
Bending	9	8	88.89
Jumping	9	7	77.78
Overall	**90**	**67**	**74.44**

TABLE VIII: CONFUSION MATRIX FOR THE WEIZMANN DATASET

	Walking	Running	Waving2	Pjumping	Siding	Jack	Skipping	Waving1	Bending	Jumping
Walking	7	0	0	0	1	0	0	0	0	1
Running	0	7	0	0	0	0	0	0	0	2
Waving2	0	0	7	0	0	0	0	2	0	0
Pjumping	0	0	0	6	2	0	0	1	0	0
Siding	1	0	0	1	7	0	0	0	0	0
Jack	0	0	2	0	0	6	0	1	0	0
Skipping	3	1	0	0	0	0	5	0	0	0
Waving1	0	0	1	0	0	1	0	7	0	0
Bending	0	0	0	0	0	0	1	0	8	0
Jumping	1	0	0	0	0	0	1	0	0	7

In addition, some categories of human behaviors are very similar to others in Weizmann dataset from the motion features point of view, which also can cause confusion in recognition, as confirmed by the confusion matrix given in Table VIII. It can be seen from Table VIII that confusion mainly comes from those behaviors with similar motion features, like "waving1" and "waving2", "pjumping" and "siding", etc. However, the confusion degrees are relatively small compared to the correct recognition rate, which means that the proposed approach is effective in distinguishing similar behaviors.

To compare the experimental results with other alternative methods, we conduct another set of experiments on the Weizmann dataset. Similarly to the previous dataset, we first compare the GRN-BCM with a traditional BCM model and an LSM model using the same spatial features as input. The experimental setup for the compared models is the same as that used for the KTH dataset. The results are listed in Table IX. From Table IX, it can be seen that the GRN-BCM model has much better performance compared with the other two models, which confirms that GRN-based modeling of neural plasticity enhances the computational power of spiking neural networks. Again, a few other state-of-the-art methods using different features have been compared on the dataset and the comparison results are provided in Table X. The HMM model is excluded in this comparative study due to its poor performance on the synthetic data as well as on the KTH dataset.

From Table X, we can see that the performance of the

GRN-BCM model is much better than that of the methods reported in [43, 49, 68]. Due to the small samples of Weizmann dataset, the overall recognition rates of all the compared methods are relatively low. Again complex and sophisticated spatiotemporal features are needed in the methods proposed in [43, 49], whereas only spatial features are required in our method. Even with the much simpler spatial features, the proposed GRN-BCM model can outperform other methods and successfully extract the temporal information from the video without decreasing the accuracy. These comparative results further verify the advantage of the proposed GRN-BCM model over the-state-of-the-art for human behavior recognition.

TABLE IX: Comparison of Different Classifiers on Weizmann Dataset

Methods	Features	Accuracy (%)
GRN-BCM	Spatial features	74.44
BCM	Spatial features	57.78
LSM	Spatial features	55.56

TABLE X: Comparison of Different Methods on Weizmann Dataset

Methods	Features	Classifier	Accuracy (%)
GRN-BCM	Spatial features	GRN-BCM	74.44
Zelniket al. [68]	Histograms	Statistical Distance Measure	58.91
Niebles et al. [49]	Spatiotemporal features	SVM	72.8
Liu et al. [43]	Spatiotemporal volumes	K-Nearest Neighborhood	68.4

VIII. Conclusion and Future Work

This paper proposes a new BCM-based spiking neural network model for temporal pattern learning and recognition, termed GRN-BCM, where GRN is used to regulate the plasticity parameters of the BCM model. An evolution strategy, the CMA-ES is employed to fine tune the parameters of the GRN model to achieve optimal performance for any specific task in hands. To evaluate the computational power of the GRN-BCM model, the proposed model is empirically compared with other machine learning models both on a time series classification problem and two human behavior recognition datasets. Combined with a corner-based spatial pattern extraction algorithm, the model is shown to be well suited for learning spatiotemporal patterns. Extensive experimental results performed on two behavior recognition datasets have demonstrated that the proposed GRN-BCM model is very efficient for behavior pattern recognition compared to other state-of-the-art the methods with the same or different extracted features reported in the literature.

Extracting visual features is critical for the recognition of human activity. The spatial features based on the corner descriptor in our work, however, are relatively simple and have their limitations. For example, the corner-based spatial feature extraction algorithm is sensitive to the noise and is not scale invariant. The proposed feature extraction method may not be able to be applied to a very complex industrial environment [63]. Thus, our future work includes developing a more robust and powerful algorithm for spatial feature extraction to handle more complex recognition tasks, such as human behavior recognition with dynamic backgrounds (i.e., moving objects in the background), or multi-object human activity recognition (i.e., several activities are occurring at the same time or multiple people are conducting different activities). Furthermore, we will investigate how to improve the self-adaptation of the proposed model so that the model can incrementally learn unseen behaviors from the online video sequences under various complex scenes and backgrounds.

References

[1] U. Alon, *An introduction to systems biology: design principles of biological circuits*, 1 ed.: Chapman and Hall/CRC, 2006.

[2] O. Antonios, P. Maja, and P. Ioannis, "Sparse B-spline polynomial descriptors for human activity recognition," *Image and Vision Computing,* vol. 2, pp. 1814-1825, 2009.

[3] P. Arena, L. Fortuna, M. Frasca, and L. Patane, "Learning Anticipation via Spiking Networks: Application to Navigation Control," *IEEE Transactions on Neural Networks,* vol. 20, pp. 202-216, 2009.

[4] D. Baras and R. Meir, "Reinforcement learning, spike time dependent plasticity and the BCM rule," *Neural Computation,* vol. 19, pp. 2245–2279, 2007.

[5] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Annals of Mathematical Statistics,* vol. 41, pp. 164-171, 1970.

[6] L. Benuskova, V. Jain, S. G. Wysoski, and N. K. Kasabov, "Computational neurogenetic modelling: a pathway to new discoveries in genetic neuroscience," *International Journal of Neural Systems,* vol. 16, pp. 215-227, 2006.

[7] L. Benuskova and N. Kasabov, *Computational Neurogenetic Modeling*: Springer, 2007.

[8] E. L. Bienenstock, L. N. Cooper, and P. W. Munro, "Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex," *The Journal of Neuroscience,* vol. 2, pp. 32–48, 1982.

[9] S. M. Bohte, J. N. Kok, and H. L. Poutre, "Error back- propagation in temporally encoded networks of spiking neurons," *Neuron computing,* vol. 48, pp. 17-37, 2002.

[10] H. Burgsteiner, "On learning with recurrent spiking neural networks and their applications to robot control with real-world devices," Ph.D, Graz University of Technology, 2005.

[11] C. Christodoulou, G. Bugmann, and T. G. Clarkson, "A spiking neuron model: Applications and learning," *Neural Netw,* vol. 15, pp. 891-908, 2002.

[12] P. Collet and J.-P. Eckmann, *Iterated Maps on the Interval as Dynamical System*: Birkhäuser Boston, 1980.

[13] C. Dance, J.Willamowski, L. Fan, C. Bray, and G. Csurka, "Visual categorization with bags of keypoints," in *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[14] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society,* vol. 39, pp. 1-38, 1977.

[15] P. Doll´ar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *VS-PETS 2005*, 2005, pp. 65-72.

[16] A. Doulamis and N. Matsatsinis, "Visual understanding industrial workflows under uncertainty on distributed service oriented architectures," *Future Generation Computer Systems,* 2011.

[17] S. M. Dudek and M. F. Bear, "Homosynaptic long-term depression in area CAl of hippocampus and effects of N-methyl-D-aspartate receptor blockade," in *Proc. Natl. Acad. Sci*, May 1992.

[18] T. Duong, H. Bui, D. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-Markov model," in *IEEE Conference Computer Vision and Pattern Recognition*, 2005, pp. 838-845.

[19] S. Edelman, N. Intrator, and J. S. Jacobson, "Unsupervised learning of visual structure," in *H.H. Bülthoff et al. (Eds.): BMCV 2002, LNCS 2525*, 2002, pp. 629–642.

[20] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *IEEE International Conference on Computer Vision*, 2003, pp. 726-733.

[21] D. Endy and R. Brent, "Modeling cellular behavior," *Nature,* vol. 409, pp. 391-395, 2001.
[22] S. Flavell and M. E. Greenberg, "Signaling mechanisms linking neuronal activity to gene expression and plasticity of the nervous system," *Annual Review of Neuroscience,* vol. 31, pp. 563-590, 2008.
[23] W. Gerstner and W. Kistler, *Spiking Neuraon Models*: Cambridge University Press, 2002.
[24] W. Gerstner and W. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, 1st ed. ed. Cambridge, U.K.: Cambridge Univ. Press, 2002.
[25] A. Graves, D. Eck, N. Beringer, and J. Schmidhuber, "Biologically plausible speech recognition with LSTM neural nets," in *Proceedings of BIO-ADIT*, 2004, pp. 127-136.
[26] N. Hansen. The CMA Evolution Strategy: A Tutorial [Online]. Available: http://www.lri.fr/~hansen/cmatutorial.pdf
[27] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary Computation,* vol. 11, pp. 1-18, 2003.
[28] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation,* vol. 9, pp. 159-195, 2001.
[29] C. Harris and M. J. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988, pp. 147-152.
[30] J. Hertzberg, H. Jaeger, and F. Sch¨onherr, "Learning to ground fact symbols in behavior-based robots," in *Proceedings of the 15th European Conference on Artificial Intelligence*, 2002, pp. 708-712.
[31] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang, "Action detection in complex scenes with spatial and temporal ambiguities," in *IEEE International Conference on Computer Vision*, 2009.
[32] H. Jaeger, "Discovering multiscale dynamical features with hierarchical Echo State Networks," Jacobs University, Technical report 10, 2007.
[33] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks," German National Research Center for Information Technology, 2001.
[34] J. W. Kalat, *Biological Psychology*, 10 ed.: Wadsworth Publishing, 2008.
[35] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *IEEE International Conference on Computer Vision*, 2005, pp. 166-173.
[36] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *IEEE International Conf. on Computer Vision*, 2007.
[37] A. Kirkwood, M. G. Rioult, and M. F. Bear, "Experience-dependent modification of synaptic plasticity in visual cortex," *Nature,* vol. 381, pp. 526–528, June 1996.
[38] V. Krüger, D. Kragic, A. Ude, and C. Geib, "The meaning of action: a review on action recognition and mapping," *Advanced Robotics,* vol. 21, pp. 1473-15-1, 2007.
[39] H. Kurashige and Y. Sakai, "BCM-type synaptic plasticity model using a linear summation of calcium elevations as a sliding threshold," in *King et al. (Eds.): ICONIP 2006, Part I, LNCS 4232*, 2006, pp. 19–29.
[40] I. Laptev and T. Lindeberg, "Space-time interest points," in *IEEE International Conference on Computer Vision*, 2003, pp. 432-439.
[41] F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 524-531.
[42] N. Y. Liang, G. B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw,* vol. 17, pp. 1411-1423, Nov. 2006.
[43] J. Liu, S. Ali, and M. Shah, "Recognizing human actions using multiple features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
[44] W. Maass, R. A. Legenstein, and H. Markram, "A new approach towards vision suggested by biologically realistic neural microcircuit models," in *Proc. of the 2nd Workshop on Biologically Motivated Computer Vision*, 2002.
[45] W. Maass, T. Natschl¨ager, and H. Markram, "A model for real-time computation in generic neural microcircuits," in *Proceedings of neural information processing systems*, 2003, pp. 229-236.
[46] W. Maass, T. Natschl¨ager, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Computation,* vol. 14, pp. 2531-2560, 2002.
[47] Y. Meng, Y. Jin, J. Yin, and M. Conforth, "Human activity detection using spiking neural networks regulated by a gene regulatory network," in *International Joint Conference on Neural Networks*, Barcelona, 2010.
[48] E. Mjolsness, D. H. Sharp, and J. Reinitz, "A connectionist model of development," *Journal of Theoretical Biology,* vol. 52, pp. 429-453, 1991.
[49] J. C. Niebles and F. Li, "A Hierarchical Model of Shape and Appearance for Human Action Classification," in *Proceedings of Computer Vision and Pattern Recognition*, 2007, pp. 1-8.
[50] D. D. O'Leary and S. Sahara, "Genetic regulation of arealization of the neocortex," *Current Opinion in Neurobiology,* vol. 18, pp. 90-100, 2008.
[51] J.-M. Odobez, D. Gatica-Perez, and S. O. Ba, "Embedding Motion in Model-Based Stochastic Tracking," *IEEE Transactions on Image Processing,* vol. 15, pp. 3514-3530, 2006.
[52] V. Petridis, B. Deb, and V. Syrris, "Detection and identification of human actions using predictive modular neural networks," in *Mediterranean Conference on Control and Automation*, 2009, pp. 406-411.
[53] C. D. Rittenhouse, H. Z. Shouval, M. A. Paradiso, and M. F. Bear, "Monocular deprivation induces homosynaptic long-term depression in visual cortex," *Nature,* vol. 397, pp. 347-350, Jan 1999.
[54] P. Rowcliffe, J. Feng, and H. Buxton, "Spiking perceptions," *IEEE Trans. Neural Netw,* vol. 17, pp. 803-807, 2006.
[55] B. Schrauwen, D. Verstraeten, and J. M. V. Campenhout, "An overview of reservoir computing: theory, applications and implementations," in *Proceedings of ESANN*, 2007, pp. 471-482.
[56] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *International Conference on Pattern Recognition*, 2004, pp. 32-36.
[57] J. H. Shin, D. Smith, W. Swiercz, K. Staley, T. Rickard, J. Montero, L. A. Kurgan, and K. J. Cios, "Recognition of Partially Occluded and Rotated Images With a Network of Spiking Neurons," *IEEE Transactions on Neural Networks,* vol. 21, pp. 1697-1709, 2010.
[58] H. Z. Shouval, G. C. Castellani, B. S. Blais, L. C. Yeung, and L. N. Cooper, "Converging evidence for a simplified biophysical model of synaptic plasticity," *Springer-Verlag. Biol. Cybern,* vol. 87, pp. 383–391, 2002.
[59] A. G. Tijsseling, "Sequential information processing using time-delay connections in ontogenic CALM networks," *IEEE Trans. Neural Netw,* vol. 16, pp. 145-159, 2005.
[60] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *The Journal of Machine Learning Research,* vol. 1, pp. 211-244, 2001.
[61] B. Van, S. Garcia-Salicetti, and B. Dorizzi, "On using the Viterbi path along with HMM likelihood information for online signature verification," *IEEE Trans. Syst., Man, Cybern. B, Cybern,* vol. 37, pp. 1237-1247, 2007.
[62] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory,* vol. 13, pp. 260-269, 1967.
[63] A. Voulodimos, D. Kosmopoulos, G. Vasileiou, E. S. Sardis, A. D. Doulamis, V. Anagnostopoulos, C. G. Lalos, and T. Varvarigou, "A Dataset for Workflow Recognition in Industrial Scenes," in *IEEE International Conference on Image Processing*, 2011.
[64] J. J. Wade, L. J. McDaid, J. A. Santos, and H. M. Sayers, "SWAT: A Spiking Neural Network Training Algorithm for Classification Problems," *IEEE Transactions on Neural Networks,* vol. 21, pp. 1817-1830, 2010.
[65] R. O. L. Wong and A. Ghosh, "Activity-dependent regulation of dendritic growth and patterning," *Nature Reviews Neuroscience,* vol. 3, pp. 803-812, 2002.
[66] Q. Wu, T. M. Mcginnity, L. Maguire, J. Cai, and G. D. Valderrama-Gonzalez, "Motion detection using spiking neural network model," in *International conference on Intelligent Computing: Advanced Intelligent Computing Theories and Applications - with Aspects of Artificial Intelligence*, Shanghai, China, 2008, pp. 76-83.
[67] T. Xiang and S. Gong, "Video behavior profiling for anomaly detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 893-908, 2008.
[68] L. Zelnik-Manor and M. Irani, "Event-based analysis of video," in *Proceedings of Computer Vision and Pattern Recognition*, Sept. 2001, pp. 123-130.