



# Networks of spiking neurons in modeling and problem solving

Krzysztof J. Cios<sup>a,b,c,d,\*</sup>, Waldemar Swiercz<sup>b</sup>, William Jackson<sup>c</sup>

<sup>a</sup>*Department of Computer Science and Engineering, University of Colorado at Denver,  
Campus Box 109, Denver CO 80217-3364, USA*

<sup>b</sup>*University of Colorado at Boulder, USA*

<sup>c</sup>*University of Colorado Health Sciences Center, Denver, CO, USA*

<sup>d</sup>*4cData, Golden, CO, USA*

<sup>e</sup>*Barbara Davis Center for Juvenile Diabetes, Denver, CO, USA*

Available online 17 June 2004

## Abstract

In this paper, we describe the networks of spiking neurons and show their applications for modeling and problem solving. We have used integrate-and-fire neuron model that closely simulates a biological neuron's behavior. First, we model the somatosensory system with Hebbian-type spike-time-dependent plasticity and show the ability of the network to self-organize. Second, we apply a network of spiking neurons for identification, via clustering, of diabetic retinopathy images using temporal correlation learning rule. Results show that the network distinguishes the diabetic objects of interest from the image background.

© 2004 Elsevier B.V. All rights reserved.

**Keywords:** Networks of spiking neurons; Integrate-and-fire neuron model; Diabetic retinopathy; Clustering; Brain modeling

## 1. Design of a network of spiking neurons

To define a network of spiking neurons we need to choose the spiking neuron model, the plasticity rule and the network topology. The first two components are described below.

\* Corresponding author. Department of Computer Science and Engineering, University of Colorado at Denver, Campus Box 109, Denver CO 80217-3364, USA. Tel.: +1-303-5564314; fax: +1-303-5568369.

E-mail address: [krys.cios@cudenver.edu](mailto:krys.cios@cudenver.edu) (K.J. Cios).

### 1.1. The neuron model

The well-known Hodgkin–Huxley neuron model is very complex and computationally demanding, because it models the membrane potential with detailed spike generation. The model accounts for several ion channels, thus, it can imitate biological neurons' way of stimulation and response. Additional accuracy is achieved by considering the neuron's compartmental geometry (nucleus, dendrites, and axon) [10], however, this usually results in very high computational complexity of the networks that are employing this type of the spiking neuron model.

Although ever-increasing computational capabilities make such models more efficient, using the same computer with a less complex neuron model allows for building of much larger networks. Furthermore, less computationally demanding models perform very well, for example, in modeling a learning process.

It is important to note that for accurate modeling it is crucial that the spiking nature of the biological neurons is preserved. Biological neurons, both excitatory and inhibitory, generate a series of action potentials (spikes, train of spikes) in response to stimulation. The spikes train duration time and frequency depend on the character and strength of the stimulus [12]. In this work we use integrate-and-fire MacGregor's neuron model [14] because it is less complex than the often used Hodgkin–Huxley model. It models the biological neuron behavior closely as regards subthreshold membrane potential and potassium channel response, while the spike generation is simplified. This decreases the model's computational complexity because simplified modeling of spike generation allows for larger time step for integration. Additionally, instead of modeling each individual channel, it imitates the neuron's excitation and inhibition properties. The model also simulates the neuron's adaptation to the stimulus. It is described by Eqs. (1.1)–(1.4).

*Spike generation:*

$$S = \begin{cases} 1 & E \geq T_h, \\ 0 & E < T_h. \end{cases} \quad (1.1)$$

*Refractory properties:*

$$\frac{dG_K}{dt} = \frac{-G_K + B \cdot S}{T_{G_K}}. \quad (1.2)$$

*Threshold accommodation:*

$$\frac{dT_h}{dt} = \frac{-(T_h - T_{h0}) + c \cdot E}{T_{th}}. \quad (1.3)$$

*Transmembrane potential:*

$$\frac{dE}{dt} = \frac{-E + G_K \cdot (E_K - E) + G_e \cdot (E_e - E) + G_i \cdot (E_i - E) + SCN}{T_{mem}}, \quad (1.4)$$

where  $SC$  is the current injected to cell.  $T_{G_K}$  is the decay of  $GK$  time constant,  $T_h$  the threshold value,  $T_{h0}$  the resting value of threshold,  $T_{th}$  decay of threshold time constant,  $T_{mem}$  the membrane time constant,  $T_{mem} = C/G$ ,  $C$ —for the second training phase capacitance,  $c$ —determines rise of threshold  $c \in [0, 1]$  and  $B$ —postfiring potassium increment.

*Transmembrane potentials:*

$$E = V - V_r \quad E_K = V_K - V_r,$$

$$E = V_i - V_r \quad E_e = V_e - V_r,$$

where  $V$  is the membrane potential,  $V_r$  the membrane resting potential,  $V_K$  the potassium resting potential,  $V_i$  the inhibitory resting potential and  $V_e$  the excitatory resting potential.

*Transmembrane conductances are:*

$$G_K = g_K/G,$$

$$G_i = g_{si}/G,$$

$$G_e = g_{se}/G,$$

where  $g_K$  is the potassium resting conductance,  $g_{si}$  the inhibitory resting conductance,  $g_{se}$  the excitatory resting conductance and  $G$  the membrane resting conductance.

*Current through a membrane is described by:*

$$SCN = SC/G$$

The neuron's membrane potential changes according to the input signals (incoming spikes). The spikes enter the neuron through synaptic connections, thereby, increasing the synaptic conductances. This results in postsynaptic potential changes.

There are two types of synaptic connections in this model: excitatory and inhibitory. The type of synaptic connection depends upon the presynaptic neuron.

The weighted sum of all excitatory and inhibitory synaptic conductances yields the excitatory and inhibitory stimulus values, respectively. If the excitatory stimulus is too weak or inhibitory stimulus too strong, the membrane potential cannot reach the threshold and the neuron does not fire. If the stimulus is strong enough for the membrane potential to reach the threshold, the neuron fires (generates a spike train traveling along the axon). For a short time, immediately after the spike generation, the neuron is incapable of responding to any additional stimulation. This time interval is referred to as the absolute refractory period. Following the absolute refractory period is an interval known as the relative refractory period. During the relative refractory period the neuron can only respond to very strong stimulation. Depending on the neuron properties, it may accommodate to the stimulating signals, requiring a stronger stimulus for the generation of consecutive spikes. Figs. 1a–d, show examples of the model responses to stimulations (a step current and a series of spikes), for both weak and strong stimulus. Figs. 2a–d, show the model responses to stimulation (series of spikes with different frequencies), and the impact of the inhibitory input.

## 1.2. Postsynaptic potentials

The spike's arrival at the neuron's synapse stimulates synaptic conduction. An increased synaptic conductance raises the synaptic current, and results in a change in

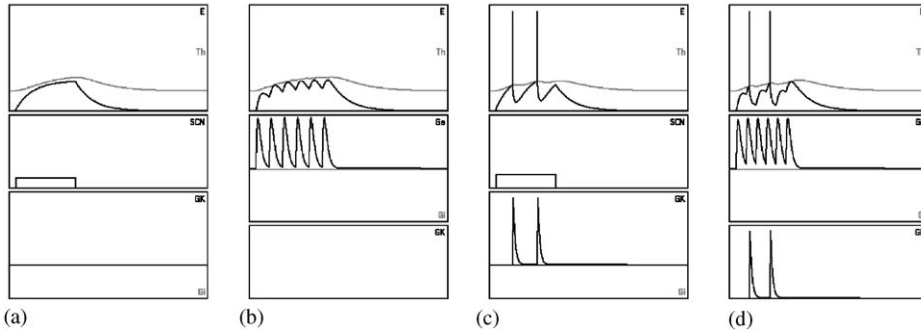


Fig. 1. Neuron model responses to different stimulations: (a) weak step current, (b) weak spikes train stimulation, (c) stronger step current, (d) stronger spikes train stimulation.

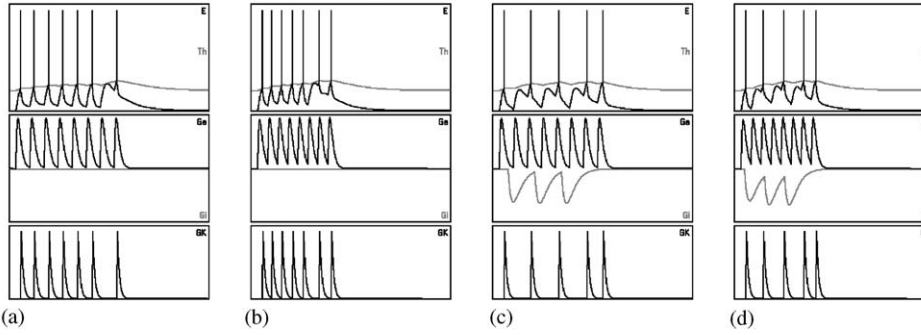


Fig. 2. Neuron responses to different frequency spike trains. (a) lower frequency, (b) higher frequency, (c) lower frequency and inhibitory input, (d) higher frequency and inhibitory input.

the postsynaptic potential (PSP) of the neuron. Depending on whether the synapse is excitatory or inhibitory, the PSP can be either increased or decreased. Increasing the excitatory conductance raises the PSP (excitatory PSP, EPSP), while increasing the inhibitory conductance reduces the PSP (inhibitory PSP, IPSP). The changes in the PSP directly affect the neuron's membrane potential and are proportional to the conductance changes. The synaptic conductance between the neurons  $i$  and  $j$  is obtained by using Eq. (1.5). Fig. 3 illustrates normalized shapes of the EPSP and IPSP. The IPSP is negative because it decreases the membrane potential.

$$g_{ij}(t) = \frac{t - \Delta t_{ij}}{T_{ij}} \exp\left(1 - \frac{t - \Delta t_{ij}}{T_{ij}}\right), \quad (1.5)$$

where  $\Delta t_{ij}$  is the propagation time,  $T_{ij}$  the synapse time constant.

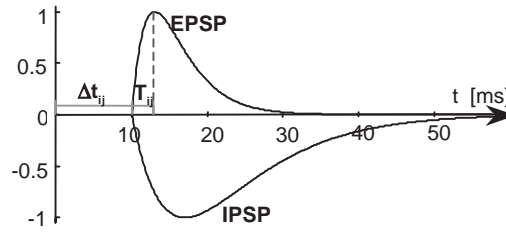


Fig. 3. EPSP and IPSP normalized shapes.

The total synaptic excitatory and inhibitory conductances are obtained by calculating the weighted sum of all excitatory and inhibitory conductances, respectively [5]. Eqs. (1.6) and (1.7) describe total excitatory ( $G_e$ ) and inhibitory ( $G_i$ ) conductances for neuron  $j$ :

$$G_e = \sum_k w_{kj} g_{kj}^e, \quad (1.6)$$

$$G_i = \sum_k w_{kj} g_{kj}^i, \quad (1.7)$$

where  $w_{kj}$  is the synaptic weight between neuron  $k$  and  $j$ ,  $g_{kj}^e$  the excitatory synaptic conductance between neuron  $k$  and  $j$ ,  $g_{kj}^i$  the inhibitory synaptic conductance between neuron  $k$  and  $j$ .

### 1.3. Synaptic plasticity

In our network model we use a Hebbian-type learning rule [9] with spike time-dependent plasticity (STDP) [17]. This model is used for learning strength of the synaptic weights. Synaptic weight values are updated according to

$$w_{ij}(t+1) = w_{ij}(t) + \alpha c_{\text{STDP}}(t_{ij}), \quad (1.8)$$

where  $t_{ij}$  is the time between firing of the neurons  $i$  and  $j$ ,  $\alpha$  the learning rate,  $c_{\text{STDP}}$  the STDP function.

Modification of the synaptic weights, between pre- and postsynaptic neurons, takes place every time the postsynaptic neuron fires. According to the STDP function, if the excitatory presynaptic neuron spike arrives shortly before the postsynaptic neuron fires, the excitatory synapse is rewarded (synaptic strength increases). On the other hand, when excitatory presynaptic neuron's spike arrives shortly after the postsynaptic neuron fires, the excitatory synapse is punished (synaptic strength decreases).

We also implemented learning mechanism for inhibitory synapses. If the inhibitory presynaptic neuron's spike arrives shortly before the postsynaptic neuron firing, the inhibitory synapse strength decreases. In a situation when the inhibitory presynaptic neuron's spike arrives and the postsynaptic neuron does not fire, the inhibitory

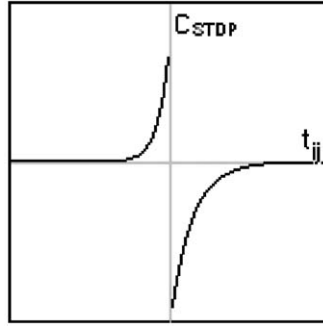


Fig. 4. STDP shape for  $\alpha_+ = 0.8, \alpha_- = 1, \tau_+ = 10, \tau_- = 20$ .

synapse strength is increased. Eq. (1.9) describes the STDP function shown in Fig. 4.

$$c_{\text{STDP}}(t_{ij}) = \begin{cases} \alpha_+ \exp\left(\frac{t_{ij}}{\tau_+}\right) & \text{if } t_{ij} < 0, \\ -\alpha_- \exp\left(\frac{-t_{ij}}{\tau_-}\right) & \text{if } t_{ij} \geq 0, \end{cases} \quad (1.9)$$

where  $t_{ij}$  is the time between the pre-synaptic spike arrival and the post-synaptic neuron firing,  $\tau_+, \tau_-$  the time constants, which determine the time intervals significant for synapse rewarding or punishment,  $\alpha_+, \alpha_-$  the amount of modification rate.

STDP function reflects both the long-term potentiation (LTP), and the long-term depression (LTD). Both LTP and LTD are observed in biological networks and are associated with learning and memory. Under the assumption that the synaptic weight values are all positive and constrained by some maximum value it has been proven that use of the STDP function leads to stable networks [18].

#### 1.4. Neuron stimulation

Our implementation of the network of spiking neurons is object-oriented. Each network layer, neuron, and synaptic connection is an autonomous object. This allows for dynamic creation and modification of the network. Furthermore, all of the parameters can be easily adjusted. Each layer is a two-dimensional matrix of neurons. Each synaptic connection is unidirectional and connects exactly two neurons, either within a layer or between two layers, and has a synaptic strength associated with it. The auxiliary object used in the model is the neuron stimulator. The network is stimulated by changing the SCN value in Eq. (1.4) which mimics “injecting current” into the neurons during biological experiments. Currently we use these shapes of the SCN to stimulate the network: step, trapezoid and the sinus function. Fig. 5 shows the different types of stimulation and the associated neuron responses.

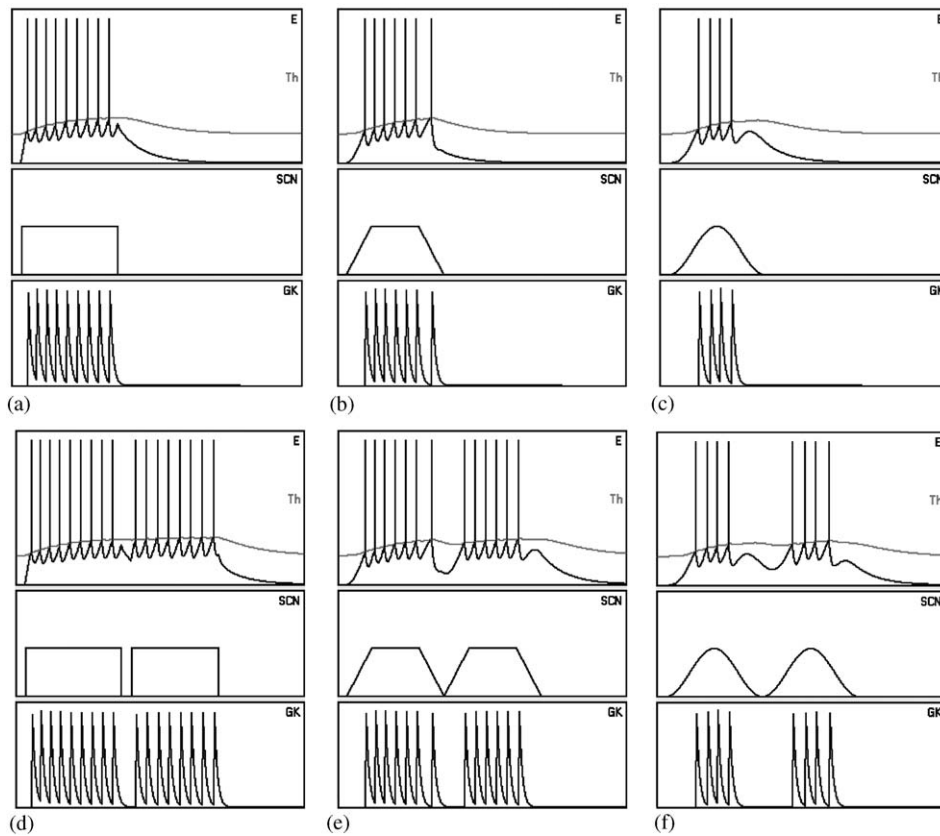


Fig. 5. Neuron responses to stimulation: (a) single step, (b) single trapezoid, (c) single sinus, (d) multiple step, (e) multiple trapezoid, (f) multiple sinus.

Each stimulator can influence anywhere from one to several neurons at a time. The program can stimulate each neuron once or the stimulation can be repeated a specified number of times. The stimulator can activate the entire set of neurons simultaneously, or a randomly selected subset of neurons. Stimulators can be also grouped into a 2D ensemble to provide differentiation of the stimulation strength (i.e. to simulate pressing touch sensory area). The center of the area generates the strongest stimulation while the stimulation strength of the remaining stimulators is inversely proportional to the distance from the center. Gaussian distribution is used to calculate the magnitude of the decrease in stimulation strength.

### 1.5. Noise

The implemented neuron model also allows for simulation of noise of the membrane potential [8]. The noise is applied only to the membrane potential equation, which

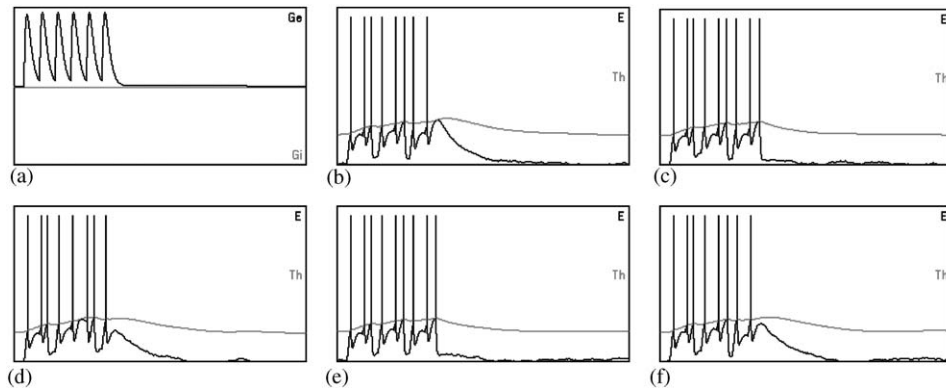


Fig. 6. Noisy neuron response (a) stimulation, (b)–(f) responses to the same stimulation.

accounts for all noise as it is the main equation of the model. The membrane potential can either be influenced by inputs from other neurons, or it can send its output to them. The amount of noise is controlled by a parameter that is set before computations as a percentage of the maximum current strength. In Fig. 6 we show neuron responses to the same synaptic stimulation, with the noise applied to the membrane potential.

## 2. Modeling of the somatosensory system

We based our somatosensory system model on the neurophysiological description of the somatosensory pathways that process the tactile inputs from the hand to the cerebral cortex in primates. The modeling described below is partially a repetition of one originally performed by Cios and Sala [5]. The difference is that the previously used learning rule did not take into account the LTD phenomenon and only the excitatory synapse learning was used. The current model takes into account both LTP and LTD, and employs both excitatory and inhibitory synapse learning. The network of spiking neurons implemented in the experiment imitates four levels of information processing: sensory afferents, spinal cord—brainstem, thalamus, and cortex. To develop this somatosensory system model we used McGregor's neuron model and the Hebbian-type plasticity rule described above.

### 2.1. Network topology

In order to simulate the four levels of information processing from the hand, our model also consists of four layers: sensory, sub-cortical 1, sub-cortical 2, and cortical. The connections between the neurons used in modeling are: local excitatory, local inhibitory and afferent excitatory. Connections' lengths are calculated as Euclidean distances between connected neurons. Separate layers are considered to be located on separate planes. Thus, the interlayer (local) connections use a 2D distance, while



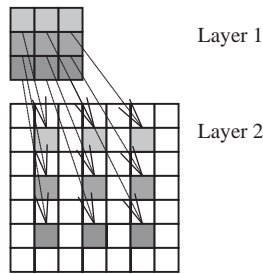


Fig. 7. Projection of smaller layer neurons on larger layer neurons.

the intralayer (afferent) connections use a 3D distance measure. The initial values of synaptic connection strengths are set to a random value in range of 0.9–1.1, i.e. about 33% of the maximum synaptic strength. To avoid edge effects, the connections wrap along the opposite edges.

#### 2.1.1. Sensory layer

The sensory layer models tactile receptors on the skin. It consists of  $9 \times 9$  matrix of excitatory neurons innervated by 9 groups of stimulators. Each stimulator group stimulates a  $3 \times 3$  sub-matrix of neurons, and represents one touch-sensitive area (e.g. digit/finger). The input signal for each area is of Gaussian shape. The central neuron stimulation is the strongest and signal strength decreases along the radius. There are no local connections between neurons. The stimulation is transformed into spikes and passed into the sub-cortical layer 1. The afferent connections to sub-cortical layer 1 are one-to-one, with the neighborhood radius of 0.

#### 2.1.2. Sub-cortical layer 1

The sub-cortical layer 1 models the spinal cord and the brainstem. It consists of  $9 \times 9$  matrix of excitatory neurons. There are no local connections between neurons. The inputs come from the sensory layer and are passed to the sub-cortical layer 2 through the afferent connections. Since the sub-cortical layer 1 dimensions are smaller than sub-cortical layer 2, the sub-cortical layer 1 neurons are evenly projected onto sub-cortical layer 2 excitatory neurons; see Fig. 7. The afferent connections to the excitatory and inhibitory neurons of the sub-cortical layer 2 connect within a neighborhood of radius 2. Fig. 8 shows the neighborhoods with three different radii.

#### 2.1.3. Sub-cortical layer 2

The sub-cortical layer 2 models the thalamus. It consists of a  $21 \times 21$  matrix of the excitatory neurons and a “blended in,”  $9 \times 9$  matrix of the inhibitory neurons evenly distributed among excitatory neurons; see Fig. 9. Stimulation comes from the sub-cortical layer 1 neurons. There are local connections within the sub-cortical layer 2. The excitatory neurons facilitate inhibitory neurons and the inhibitory neurons depress

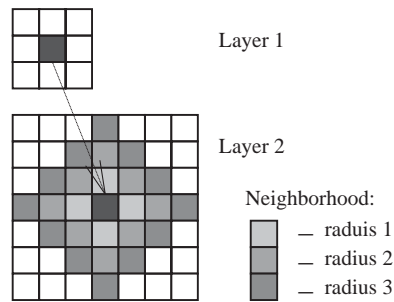


Fig. 8. Neighborhood of the neuron.

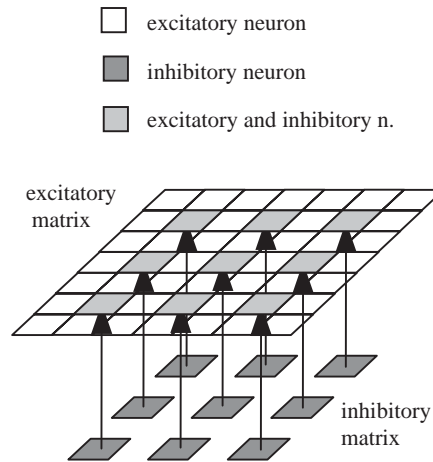


Fig. 9. Blended excitatory and inhibitory matrices.

excitatory neurons, both with the neighborhood radius of 2. Since the number of excitatory neurons is bigger than the number of inhibitory neurons, synaptic connections between them have to be evenly projected in the same way as explained earlier for the afferent connections. Also the excitatory connections to inhibitory neurons are evenly projected, but in an inverted fashion. The afferent connections only come out from the layer's excitatory neurons. They facilitate the cortical layer's excitatory and inhibitory neurons, within a neighborhood of radius 2. Connections to excitatory neurons are one to one; while the connections to the inhibitory neurons are evenly projected in the manner described above.

#### 2.1.4. Cortical layer

The cortical layer models the cortex. It consists of a  $21 \times 21$  matrix of the excitatory neurons and a blended in  $9 \times 9$  matrix of the inhibitory neurons evenly distributed among the excitatory neurons; see Fig. 9. The local connections between excitatory

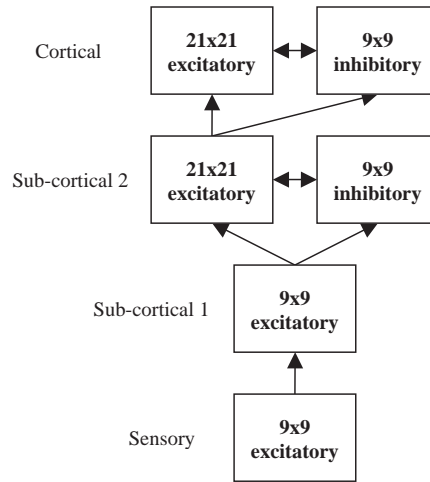


Fig. 10. Schema of the network topology.

and inhibitory neurons are implemented in the same way as in the sub-cortical layer 2. Fig. 10 shows the entire network topology.

## 2.2. Experiments

We performed two experiments, using the same network topology and stimulation procedures. The only difference was that one used only excitatory synapses learning, while the other used both excitatory and inhibitory synapses learning. The sensory layer was divided into 9 areas, each of size  $3 \times 3$ , representing separate sensory fields, corresponding to nine “digits/fingers”. Each field was innervated by a separate stimulator. Each stimulator could “inject” current into specified neurons. The stimulation signal is a step function that lasts for 50 ms, followed by the 50 ms break. The stimulation signal is distributed on each of the  $3 \times 3$  areas using Gaussian distribution. The network training consisted of a number of stimulation cycles with the learning rule using the STDP function. In every stimulation cycle all areas are randomly stimulated; each area separately from all other areas. The network testing was performed by stimulating each area, with no learning rule used, and then the cortical layer’s response was read. First, the untrained network was tested to obtain the initial response for later comparison (see Fig. 11). Then the network was trained. There were two phases of training. The first phase consisted of 5 stimulation cycles. Parameters of the learning rule for the first phase were:  $\alpha_+ = 0.5$   $\alpha_- = 0.6$   $\tau_+ = 10$   $\tau_- = 20$  for excitatory learning, and  $\alpha_+ = 0.6$   $\alpha_- = 0.7$   $\tau_+ = 9$   $\tau_- = 23$  for both excitatory and inhibitory learning. In this phase there was a tendency to increase the synaptic connections strength. Some of the strengths rose to 70% of the maximum value (see Figs. 12 and 14). The second phase consisted of 8 stimulation cycles. The parameters of the learning rule for the second

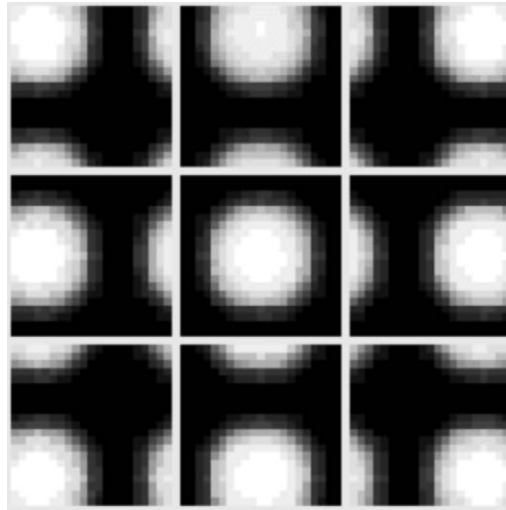


Fig. 11. Untrained network response.

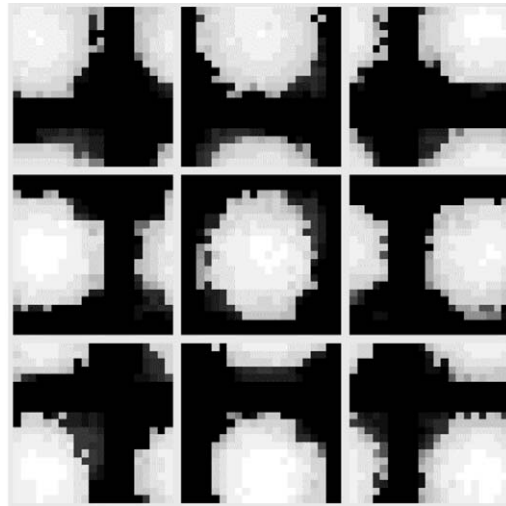


Fig. 12. Response after first phase; with excitatory synapses training.

phase were:  $\alpha_+ = 0.4$   $\alpha_- = 0.7$   $\tau_+ = 9$   $\tau_- = 22$  for excitatory learning, and  $\alpha_+ = 0.4$   $\alpha_- = 0.9$   $\tau_+ = 7$   $\tau_- = 25$  for both excitatory/inhibitory learning. In this phase only the strongest connections remained, while the weaker ones either decreased significantly or vanished altogether. Next, the trained network was tested in order to compare its response to

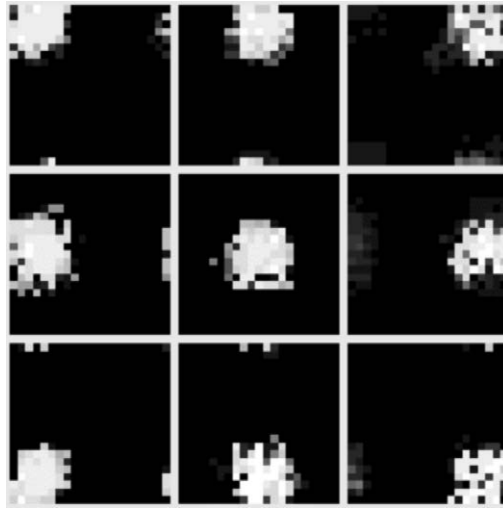


Fig. 13. Response after complete training; with excitatory synapses training.

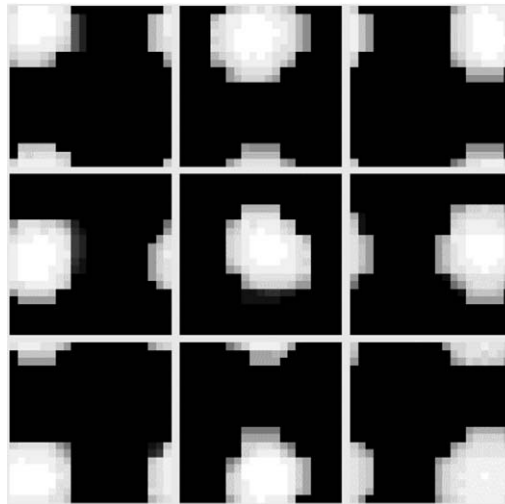


Fig. 14. Response after first phase; with excitatory/inhibitory synapses training.

the untrained network (Figs. 13 and 15). The values of the training parameters were established experimentally. The stimulation cycle numbers were fixed to 5 and 8 for the first and second phases, respectively, since further computations were just slightly oscillating around the shown values.

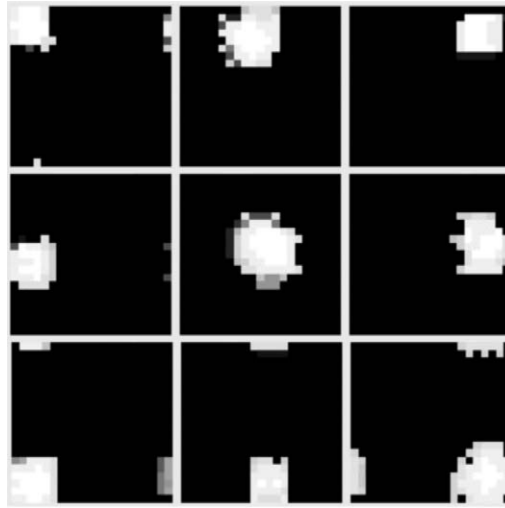


Fig. 15. Response after complete training; with excitatory/inhibitory synapses training.

### 2.3. Results

The results are presented in the form of graphic images as shown in Figs. 12–15. For each test there are 9 images, each corresponding to one stimulation area. The images represent the cortical layer's excitatory neurons responses (number of spikes generated). Black represents a lack of response, while white represents the strongest response. Fig. 10 shows the response of the untrained network. The results of the excitatory learning with excitatory learning are shown in Fig. 11 for the first training phase and in Fig. 12 for the second training phase. The results for both the excitatory and the inhibitory learning are shown in Figs. 13 and 14 for the first and second training phases, respectively.

The results illustrate the network's, using the STDP learning ability to self-organize into maps via the process of competition between neurons. The competition between synapses [1] takes place in the following way. The synaptic connections strengths between neurons belonging to the same stimulation area tend to increase, while connection strengths between neurons from distinct areas tend to decrease and vanish. The results for the experiment including inhibitory synapses learning are better than the results without it; areas corresponding to separate stimulation regions are better separated and more uniform.

## 3. Diabetic retinopathy data

Diabetic retinopathy (DR) is caused by diabetes and is one of the major causes of blindness. Elevated blood sugar in diabetics causes the blood vessels in the retina

to leak, close, and proliferate, damaging the retina. Nearly all patients with diabetes will eventually develop DR but they will not know that they have it until the later stages of the disease [2]. If not properly treated, DR can cause blindness, but with appropriate treatment over 90% of visual loss can be prevented. Thus diagnosis of early stages of DR is very important. The reason is that the patient will not develop severe DR before being detected by the next examination, and appropriately treated. Unfortunately, about 50% of diabetic patients are never examined and only 30% of them are examined regularly [19]. Automatic computer recognition of DR through image analysis may provide for such early detection. More patients can be examined since only a technician will be required to do the screening. Only patients diagnosed positively for sight threatening DR would be required to see the ophthalmologist.

### 3.1. DR images and their preprocessing

Digital images of the fundus are used to detect neovascularization, exudates, hemorrhages, and microaneurysms that indicate positive diagnosis of DR. The severity level of the disease is diagnosed depending on the pattern of abnormalities. The fundus images used are  $2160 \times 1140$  pixels in a 24-bit color. Since the green layer of the image contains the most information (red and blue layers contain almost no information) only the green layer was used. The color, contrast, and brightness of each image are distinct for each patient, as well as non-uniformity of the image intensity. To diminish the influence of those factors, two filters are used: the Center Weighted Median (CWM) filter to remove noise [3] and the Flat Fielding (FF) filter which is based on a Fast Fourier Transform (FFT) to make intensity of the images uniform [15].

The CWM filter is very effective in removing noise from the images. The CWM works almost like the Median Filter (MF). However, it has an advantage over the simple (MF) in that it preserves image details and allows for good noise removal. The difference is that all MF mask pixels, except for the center one, are assigned a weight value of 1. The central pixel of the MF mask is assigned a weight bigger than one. The value of that weight specifies the strength of the filtering. After sorting all the pixels in the window, the selection of median value takes place. Beginning with the pixel with the highest value the corresponding weights are added until the sum of the weights is bigger than or equal to the average of all the weights. The value of the pixel for which summation has stopped is returned as a median value. The FFT filter helps in reducing non-uniform intensity of the image. First, the original image is filtered using a low-pass filter based on the FFT. Then, the low-pass filtered image is subtracted from the original image. This operation leaves only the high and the medium frequency features on the image. After applying the filters to the image, its histogram is expanded to increase the contrast. Preprocessed images are then used as input for clustering using a network of spiking neurons.

However, the large dimensions of the images, cause difficulties in performing the computations on the entire image using the network of spiking neurons. To address this problem we decided to process sub-images of the size  $32 \times 32$  pixels. Fig. 16

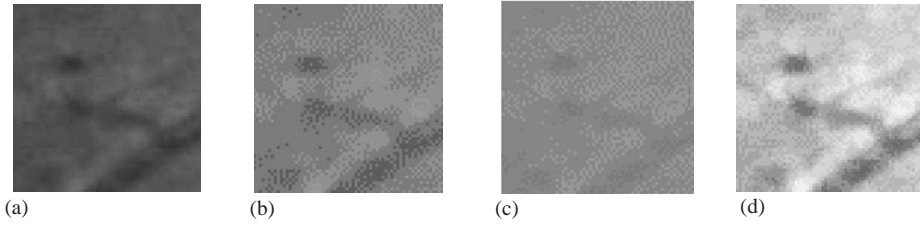


Fig. 16. Image preprocessing steps. (a) Green layer; (b) CWM filter; (c) flat fielding; (d) expanding.

illustrates the four steps of image preprocessing on the sample image containing part of a blood vessel and a small hemorrhage.

### 3.2. Synaptic plasticity rule

Image objects (i.e. microaneurysms and hemorrhages) can be clustered by using the same neuron model that we used for somatosensory system modeling. The distinction is that we use a different synaptic plasticity rule, and the appropriate network topology. To adjust the synaptic connections strengths we use the temporal correlation learning (TCL) rule of Sala and Cios [6], which is defined by Eqs. (2.0) and (2.1).

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \cdot c_{\text{corr}}(t_{ij}), \quad (2.0)$$

where  $t_{ij}$  is the time between firing of neurons  $i$  and  $j$ ,  $\alpha$  the learning rate and  $c_{\text{corr}}$  the temporal correlation function.

Modification of the synaptic weights, between pre- and postsynaptic neurons, takes place every time the postsynaptic neuron fires. The correlation function, used within the TCL rule, evaluates whether a particular synapse is to be rewarded (synapse had influence on postsynaptic neuron firing) or punished (synapse did not have influence on postsynaptic neuron firing). The correlation function is defined, after [14], by Eq. (2.1), and is illustrated in Fig. 17

$$c_{\text{corr}} = (1 + y) \exp\left(-k \frac{t_{ij}^2}{t_{\text{corr}}^2}\right) - y, \quad (2.1)$$

where  $t_{ij}$  is the time elapsed between firing of  $j$  and  $i$ ,  $t_{\text{corr}}$  the time window size,  $y$  the max value for the decay,  $y \in (0, 1]$ ;  $k = \ln(1 + 1/y)$ —assures that  $c_{\text{corr}}(t_{\text{corr}}) = 0$ .

The following example illustrates how the TCL rule works. Let us consider two presynaptic neurons  $N_2$  and  $N_3$  and one postsynaptic neuron  $N_1$ , as shown in Fig. 18. The  $t_{\text{corr}}$  value is set to 5 ms. Let us assume that a postsynaptic neuron  $N_1$  fires at time  $t_1 = 0$ . Presynaptic neuron  $N_2$  fired 2 ms before, and presynaptic neuron  $N_3$  fired 7 ms before. Since the time window size is 5 ms, only neuron  $N_2$  falls inside the window, so only the weight of the synapse between  $N_2$  and  $N_1$  will increase. At the same time the weight of synapse between  $N_3$  and  $N_1$  will decrease. Fig. 19 shows the activation times and the time window for this example.



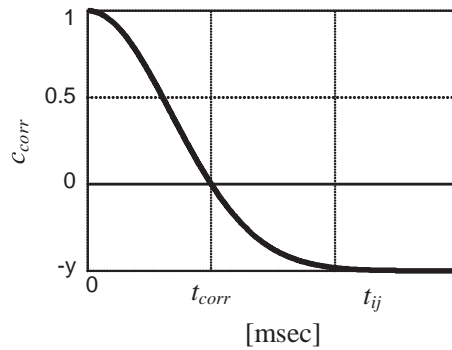
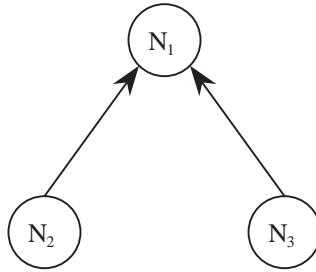


Fig. 17. Temporal correlation function.

Fig. 18. Postsynaptic neuron  $N_1$ , and presynaptic neurons  $N_2$  and  $N_3$ .

### 3.3. Image clustering network topology

The network consists of only one layer of excitatory neurons. The dimensions of the layer are the same as the dimensions of the image so the number of neurons within the layer corresponds to the number of pixels in the image [4]. The intensity difference between the neighboring pixels is used as a connection length between the respective neurons in the network. Only connections between neighboring pixels are computed. All remaining values are set to infinity. We use two neighborhoods in the experiments: one of dimension  $3 \times 3$  and the other  $5 \times 5$ , see Fig. 20. The black circle represents the neuron under consideration. The gray circles represent adjacent neighboring neurons, and the white circles represent non-adjacent neurons.

For example, if the intensity difference between two neighboring pixels is 50, the time delay between the corresponding neurons is set to 50 ms. All synaptic connection strengths are initially set to value in the range (1.4–1.6); about 50% of the maximum value of 3.

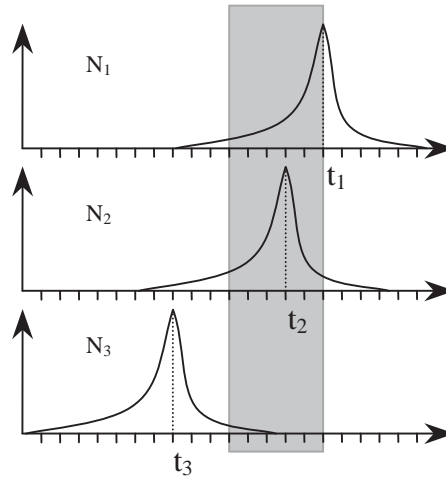
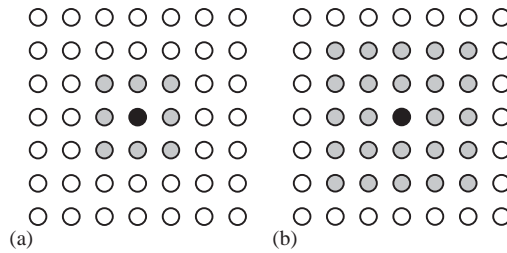


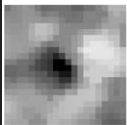


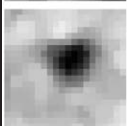


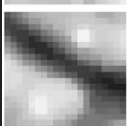


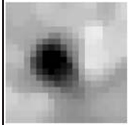
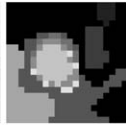
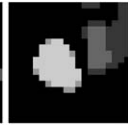
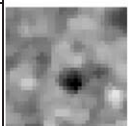


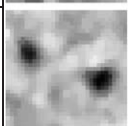


Fig. 19. Activation times and time window.

Fig. 20. Neighborhood types: (a) dimension  $3 \times 3$ , (b) dimension  $5 \times 5$ .

### 3.4. Experiments

Experiments were performed separately for each neighborhood type. The value of the  $t_{\text{corr}}$  was initially set to a relatively big value of 5.3, which is 70% of the maximum value. In each iteration the neurons were stimulated in a random manner. The weights of synapses between neurons were adjusted (rewarded or punished) according to the TCL rule. After each iteration the value of the  $t_{\text{corr}}$  was reduced, thus narrowing the correlation time window. This resulted in decreasing the number of rewarded synapses in the following iteration, which caused many synapses to vanish. At the end of the computations only part of the original synapses remained. The remaining connections defined the clusters of neurons, corresponding to objects of interest within the image. The network of spiking neurons finds clusters automatically without the need to a priori specification of the number of clusters. A similar type of network was originally used by Sala and Cios [16] for solving graph problems and was shown to find correct number of clusters in the data automatically (without specifying their number).

Table 1  
Results

	Original	Neighborhood	
		3x3	5x5
Image1			
Image2			
Image3			
Image4			
Image5			
Image6			


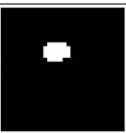

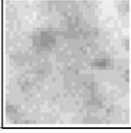


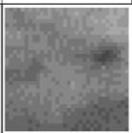
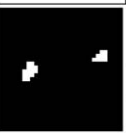

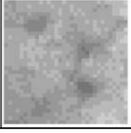
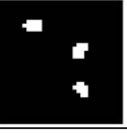
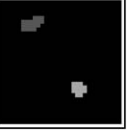
### 3.5. Results

The goal of the experiment was to identify and distinguish between important objects of interest, against an image background. Table 1 shows some of the results in a graphical form. Column named ‘Original’ contains subimages of size  $32 \times 32$  pixels, preprocessed in the manner described above. Several subimages containing important objects were selected for the experiment. Images 1, 2 and 4, contain hemorrhages, Image 3 part of a blood vessel, and Images 5 and 6 contain microaneurysms. The images after clustering are shown in two columns named Neighborhood  $3 \times 3$  and Neighborhood  $5 \times 5$ , respectively. The size of the neighborhood influences the number and size of objects detected by the network. The Neighborhood  $5 \times 5$  seems to produce results with more uniform objects, but detects fewer objects than Neighborhood  $3 \times 3$ . For instance on Image 3 the blood vessel was not detected by the algorithm when using the larger neighborhood. Although Neighborhood  $3 \times 3$  did detect the blood vessel, the results are noisier and contain more artifacts (images 1, 4, and 5). Overall the results are satisfactory, especially for the Neighborhood  $5 \times 5$ , because the network distinguishes well between the image background and the objects of interest. The resulting output images can be used as input to, say, Image Recognition Neural Network [7] to improve performance of DR recognition [11].

Results of detection, using the network of spiking neurons, were compared with the results of algorithm that uses the watershed transform to detect abnormalities in DR images [13]. The watershed transform results are presented after masking (removal) of blood vessels, and after pixel value checking (which removed all non-relevant objects that have been detected), so the comparison is not quite fair.

Because we used subimages taken from [13] that were of lower quality, the neighborhood used in all computations was  $3 \times 3$ , regardless of the previous results. Table 2

Table 2  
Results comparison

	Original	Watershed	Spiking
Image 1			
Image 2			
Image 3			
Image 4			

compares the results achieved by the two algorithms. As can be seen, the results of the network of spiking neurons are comparable with those of the watershed algorithm. It needs to be stressed, however, that they were obtained without masking blood vessels and without pixel value checking. In some cases, a very low image contrast may cause detection of artifacts, as seen in Image 3 in Table 2. The solution to this problem could be to preprocess the images by another algorithm that is able to remove the artifacts.

#### 4. Conclusions

The network of spiking neurons performed very well in modeling of the somatosensory system and clustering of the objects of interest in diabetic retinopathy images. The network with spike time-dependent plasticity learning rule for both excitatory and inhibitory synapses learning possesses self-organizing capabilities. It correctly mapped stimulation regions onto cortical area.

The DR image clustering is an original application of the network of spiking neurons to image segmentation. The clustering results were achieved in a fully automatic way; the number of clusters returned was not a priori specified. Comparison with the watershed algorithm revealed that the network of spiking neurons algorithm generates good results. The resulting images can be used as input to other image recognition algorithms like the IRNN, which was previously used for DR detection from raw DR images. The advantage of the network of spiking neurons is that it removes irrelevant information from images. As a result its output can be used as input to many other classification algorithms. The drawback is its computational complexity that makes it difficult to use it for processing large images. To address the problem we plan to implement the STDP learning rule for image clustering, use pixel values as stimulation strengths and Euclidean distance as connections' lengths.

## References

- [1] L.F. Abbot, S.B. Nelson, Synaptic plasticity: taming the beast, *Nature Neurosci. Suppl.* 3 (2000) 1178–1183.
- [2] L.P. Aiello, et al., Diabetic retinopathy, *Diab. Care* 21 (1) (1998) 143–156.
- [3] A. Bovik, *Image and Video Processing*, Academic Press, New York, 2000.
- [4] K.J. Cios, W. Jackson, W. Swiercz, L. Springhetti, Spiking neurons in clustering of diabetic retinopathy data, *Second International Conference On Hybrid Intelligent Systems*, Santiago, Chile, 2002, pp. 84–91.
- [5] K.J. Cios, D.M. Sala, Advances in applications of spiking neuron networks, *SPIE AeroScience 14th International Symposium, Applications and Science of Computational Intelligence III*, Orlando, FL, 2000, pp. 324–336.
- [6] K.J. Cios, D.M. Sala, Networks of spiking neurons in data mining in: S.K. Pal, A. Pal (Eds.), *Pattern Recognition: From Classical to Modern approaches*, World Scientific, Singapore, 2001, pp. 329–346.
- [7] K.J. Cios, I. Shin, Image recognition neural network: IRNN, *Neurocomputing*, 7 (2) (1995) 159–185.
- [8] W. Gerstner, Integrate and fire neurons and networks, in: M.A. Arbib (Ed.), *The Handbook of Brain Theory and Neural Networks*, 2nd edition, The MIT Press, Cambridge, MA, 2002, pp. 577–581.
- [9] W. Gerstner, W. Kistler, Mathematical formulations of Hebbian learning, *Biological Cybernetics* 87 (2002) 404–415.
- [10] <http://neuron.duke.edu>.
- [11] W. Jackson, K.J. Cios, W. Swiercz, L. Springhetti, Computer Analysis of Diabetic Retinopathy, *J. Am. Diabetes Assoc.* 51 (Suppl. 2) (2002) A208.
- [12] E.R. Kandel, et al., *Principles of Neural Science*, McGraw-Hill Companies, New York, 2000.
- [13] G. Luo, C. Opas, et al., Abnormality detection in automated mass screening system of diabetic retinopathy, *Computer-based Medical System 2001, Proceedings of the 14th IEEE Symposium*, 2001.
- [14] R.J. MacGregor, *Theoretical Mechanics of Biological Neural Networks*, Academic Press Inc., New York, 1993.
- [15] S. Michael, et al., *Practical Algorithms for Image Analysis*, Cambridge University Press, Cambridge, 2000.
- [16] D.M. Sala, K.J. Cios, Solving graph algorithms with networks of spiking neurons, *IEEE Trans. Neural Networks*, 10 (4) (1999) 953–957.
- [17] S. Song, L.F. Abbot, Cortical development and remapping through spike timing-dependent plasticity, *Neuron*, 32 (2001) 339–350.
- [18] S. Song, K.D. Miller, L.F. Abbot, Competitive Hebbian learning through spike timing-dependent synaptic plasticity, *Nature Neurosci.* 3 (9) (2000) 919–926.
- [19] H.R. Taylor, Diabetic retinopathy: a public health challenge, *Am. J. Ophthalmol.* 123 (4) (1997) 543–545.