

Image Information Extraction Based on Spiking Neural Networks ^{*}

Xu Wang, Zhi-Qiang Cao, Long Cheng, Chao Zhou, Min Tan,
Zeng-Guang Hou

*Key Laboratory of Complex Systems and Intelligence Science,
Institute of Automation, Chinese Academy of Sciences, Beijing,
100190, China (e-mail: xu.wang@ia.ac.cn).*

Abstract: Inspired by the excellent performance of the biological visual system and relative experimental results, a three-layer spiking neural network is designed to extract the image information and it maps the information into the output spike space. The input neuron in the first layer that detects the image edges are direction-selective. Furthermore, these neurons are assumed to activate the neurons in their preferred directions, such that the information content of spikes can be increased. The information obtained is encoded in the temporal and spacial structure of the output spikes, which will be utilized for the further image processing.

Keywords: Image processing, Neural network, Encoding.

1. INTRODUCTION

Nowadays, image processing plays a very important role in the engineering, and it is widely applied to many fields, such as multimedia, biometrics, biomedical imaging, remote sensing, optical character reading (Camastra (2007)), etc. It is well known that the biological visual system can detect and recognize the objects in a very short time (Delorme and Thorpe (2001)). However, the mechanism of the cortex to process image and the way by which the image is encoded is still beyond our knowledge. Many scientists have engaged in this field and some important characteristics of the visual cortex have been obtained, for example, the direction selection of the visual cortex (G.H. Henry and Bishop (1974)).

On the other hand, the spiking neural network (SNN), considered as the third generation of the artificial neural network, has been developed based on the characteristics obtained from the real biological neurons (Maass and Bishop (1999)). Different from other neural networks, the information processed is encoded by the spikes, which is the essential characteristic of SNN. Many spiking neuron models have been proposed to design new suitable controllers, such as the leaky integrate and fire (LIF) neuron (Tal and Schwart (1997)), spike response model (SRM), Hodgkin-Huxley Model (Maass and Bishop (1999)), etc. Furthermore, besides the usual rate coding, some coding methods are proposed, for example, time-to-first-spike coding, phase-coding, correlation coding and so on (Maass and Bishop (1999)). At present, SNN has been applied to solve many problems, such as the control of the mobile robot (Floreato and Mattiussi (2001), H. Qu and Yi

(2009)), phase/frequency correlations recognizing (Kiselev (2009)), movement prediction from realworld images (H. Burgsteiner and Steinbauer (2005)), movement generation of the robot arm (Joshi and Maass (2005)), etc.

By noting that SNN has many advantages over other neural networks, some SNN based methods have been proposed for image information extraction. Banks et al. have presented an adaptive, current-mode analogue neuron circuit to implement image processing by adapting the spiking output frequency and pulse width according to the magnitude and vector of the input (D. Banks and Toumazou (2007)). Bodyanskiy and Dolotov propose a self-learning fuzzy spiking neural network to separate overlapping classes (Bodyanskiy and Dolotov (2008)) and Zhan et al. design a local-connected neural network model for image segmentation and edge detection (K. Zhan and Ma (2009)).

In this paper, the image is processed by a three-layer spiking neural network, which extracts the information of the outlines in the image and encodes the information by the temporal and spacial structure of output spikes. The spiking neural network is recurrent and its neurons are locally connected. The input neurons in layer 1 are direction-selective with the input image. Also, the outputs of these neurons are assumed to be direction-selective. Compared with the existing works, the output spikes of the proposed method can provide the information of the outlines' range and skeleton directly. The rest of the paper is organized as follows: the proposed spiking neural network is described in Section 2, including the neuron model, the structure of SNN, the neurons' direction selection and the information obtained. The simulation is given in Section 3 and Section 4 concludes the paper.

^{*} This work is supported in part by National Natural Science Foundation of China under Grants 60805038, 61004099, 60725309, the Natural Science Foundation of Hebei Province Grants F2010000437 and the start-up fund for the recipient of Presidential Award of Chinese Academy of Sciences .

2. IMAGE INFORMATION EXTRACTION BY SPIKING NEURAL NETWORK

A three-layer recurrent spiking neuron network is used to extract the information of the outlines in the image. The spiking neurons are locally connected and the temporal and spacial structure of its output spikes contains the information extracted from the object in the picture, based on which the digital image operations, such as object recognition etc, can be carried out.

2.1 The Spiking neurons

The neurons follow SRM model (spike response model). For every neuron $N_{i,j}$ with position (i, j) , its potential $U_{i,j}$ at time t can be calculated as

$$U_{i,j}(t) = \sum_{u,v} w_{ij}(u,v)\epsilon(t - t_{u,v}). \quad (1)$$

where $w_{ij}(u,v)$ is the weight for the input with subscript (u,v) and $t_{u,v}$ is the time when the input spike arrives. ϵ is the corresponding PSP (post synaptic potential) kernel function, which has two forms according to the characteristic of the input:

- If the inputs come from the image, $\epsilon(t) = I(u,v)$, here $I(u,v)$ is the gray scale value of the pixel (u,v) .
- If the input is the input from the neuron $N_{u,v}$ in other layer, $\epsilon(t) = \exp(-\frac{t}{\tau_m}) - \exp(-\frac{t}{\tau_s})$. By considering t being discrete (counted by cycles) and $\exp(\cdot)$ decreasing fast, $\epsilon(t - t_{u,v})$ could be approximated as

$$\epsilon(t - t_{u,v}) = \begin{cases} 1, & t_{u,v} \in (t - 1, t] \\ 0, & \text{otherwise} \end{cases} = O_{u,v}(t - 1) \quad (2)$$

where $O_{u,v}(t)$ is the output of the neuron $N_{u,v}$.

Thus, the potential $U_{i,j}$ can be approximated by using spacial filter, i.e.

$$U_{i,j}(t) = \sum_{u,v} A_{u,v}\epsilon(t - t_{u,v}). \quad (3)$$

where the element $A_{u,v}$ of the spacial filter A is the corresponding weight.

When the potential exceeds the constant threshold, the neuron fires and generates one spike to the next layer. Then, the neuron turns into refractory period. Furthermore, the temporal structure of output spike contains the information of the image, which means it can be seen as a kind of temporal coding.

2.2 The Structure of Spiking Neural Network

The proposed spiking neural network contains three layers, as shown in Fig. 1, where the solid lines denote the connections from one layer to the next layer while dashed lines are the feedback connections from one layer to its previous layer. In every layer, the neuron $N_{i,j}$ only connect to its corresponding neurons in the next layer, where the ‘‘corresponding’’ neurons are those neurons in the position (i_1, j_1) s.t. $\sqrt{(i_1 - i)^2 + (j_1 - j)^2} \leq \delta_r$ with small positive constant δ_r . Then, the neurons only need to deal with local information and can be implemented more easily in the real applications. There are no connections between the

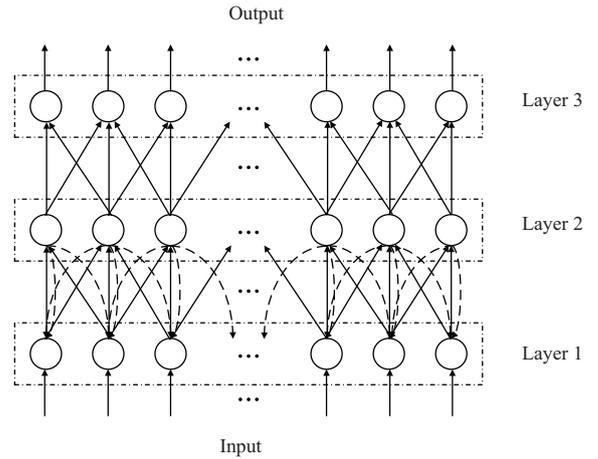


Fig. 1. The structure of SNN.

neurons in the same layer. The sizes of layer 2 and 3 are the same, denoted by $n \times m$, while the size of layer 1 is $n \times m \times l$ with l being the number of the selective directions.

The first layer detects the direction of the edge of the object in the image and the $n \times m \times l$ neurons which are direction-selective according to the biological results (Dayan and Abbott (2001)). Every input pixel $I(i, j)$, $i = 1, \dots, n$; $j = 1, \dots, m$, corresponding to one pixel neuron group $l_1(i, j)$ composing $l = 4$ neurons, which are sensitive to the line with slope $0, 1, \infty, -1$, respectively.

The neurons in the second layer receive the output spikes from the previous layer and fire spikes to generate the output to layer 3. Also, these spikes are fed back into the first layer and the first layer detects the edge of the result of the layer 2. This process will cycle until the image information extraction finishes. The layer 3 generates the output spikes, which contain the information extracted from the object in the image.

2.3 The Selection of the Direction about the Neurons

As mentioned above, for every direction-selective neuron, the space filters are used to imitate the function of direction-selection as follows:

$$A_1^1 = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ 4 & 4 & 4 & 4 & 4 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}, A_2^1 = \begin{bmatrix} -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \end{bmatrix},$$

$$A_3^1 = \begin{bmatrix} 0 & 0 & -1 & -1 & 3 \\ 0 & -1 & -1 & 4 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & 4 & -1 & -1 & 0 \\ 3 & -1 & -1 & 0 & 0 \end{bmatrix}, A_4^1 = \begin{bmatrix} 3 & -1 & -1 & 0 & 0 \\ -1 & 4 & -1 & -1 & 0 \\ -1 & -1 & 4 & -1 & -1 \\ 0 & -1 & -1 & 4 & -1 \\ 0 & 0 & -1 & -1 & 3 \end{bmatrix},$$

where $A_k^1, k = 1, 2, 3, 4$ are the space filters corresponding to the 4 directions, respectively and the elements of A_k^1 can be considered as the weight from the input to the neurons. Denote $l = [0, 1, \infty, -1]$ the vector containing four kinds of sensitive directions.

Then, the potential $U_1(i, j, k)$ of the neuron with sensitive direction $l(k)$ in the position (i, j) is obtained by convolv-

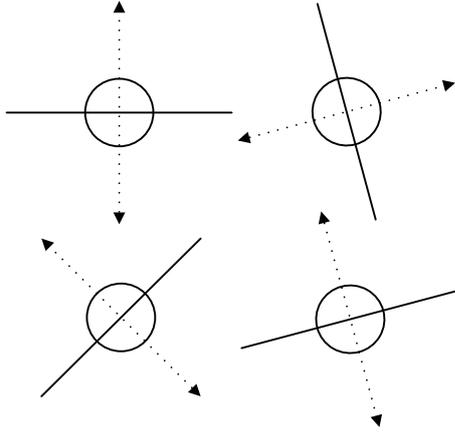


Fig. 2. The direction selection of the neuron in layer 1.

ing the inputs with the neuron's corresponding space filter as

$$U_1(i, j, k) = \sum_{u=-2}^2 \sum_{v=-2}^2 I(i+u, j+v) A_k^1(3+u, 3+v) \quad (4)$$

where I is the input from the image. When its potential exceeds the constant threshold $thred_1$, the neuron fires and generates one spike to the next layer.

Furthermore, the outputs of the neurons are assumed to be direction-selective, i.e., for the next layer, only the neurons in the preferred direction can receive the excitatory spike. By using output direction selection, the information of the neuron with direction selection can be transmitted to the neurons in the preferred direction by propagating spikes. Using the preferred direction will increase the information content of the spikes while decreasing the fire rate of the network, which means the power consumed can be reduced. Thus, we assume that there is one preferred direction for very neuron in the first layer, and to make the neuron symmetric, this direction is chosen to be vertical to the neuron's sensitive direction, as follows:

Assumption 1. The direction-selective neuron outputs its spike in a preferred direction.

In Fig. 2, four neurons with different direction selections are shown. For each neuron, the solid line represents the most sensitive direction that the neuron can response maximally to and the dotted arrows are the preferred direction for the output spikes to propagate. Different kinds of direction-selective neurons have different preferred directions.

Also, the 9×9 space filters are used to implement the direction selection of the outputs. In this paper, denote the sensitive angle of the line for a neuron in the first layer by $\theta \in [0, 180^\circ)$, then the preferred direction is perpendicular to the sensitive line, i.e., $\theta + 90^\circ$ (here, the directions θ and $\theta + 180^\circ$ are considered as the same direction). The corresponding space filters are

$$A_1^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -6 & -6 & -6 & -6 & -6 & -6 & -6 & -6 & -6 \\ -6 & -6 & -6 & -6 & -6 & -6 & -6 & -6 & -6 \\ -6 & -6 & -6 & -6 & -6 & -6 & -6 & -6 & -6 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A_2^2 = \begin{bmatrix} 0 & 0 & 0 & -6 & -6 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & -6 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & -6 & -6 & 0 & 0 & 0 \\ 0 & 1 & 0 & -6 & -6 & -6 & 0 & 1 & 0 \\ 2 & 2 & 1 & -6 & -6 & -6 & 1 & 2 & 2 \\ 0 & 1 & 0 & -6 & -6 & -6 & 0 & 1 & 0 \\ 0 & 0 & 0 & -6 & -6 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & -6 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 & -6 & -6 & 0 & 0 & 0 \end{bmatrix},$$

$$A_3^2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & -6 & -6 \\ 1 & 2 & 1 & 0 & 0 & 0 & -6 & -6 & -6 \\ 1 & 1 & 1 & -1 & -6 & -6 & -6 & -6 & 0 \\ 0 & 0 & -1 & -6 & -6 & -6 & -6 & 0 & 0 \\ 0 & 0 & -6 & -6 & -6 & -6 & -1 & 0 & 0 \\ 0 & -6 & -6 & -6 & -6 & -1 & 1 & 1 & 1 \\ -6 & -6 & -6 & 0 & 0 & 0 & 1 & 2 & 1 \\ -6 & -6 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix},$$

$$A_4^2 = \begin{bmatrix} -6 & -6 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ -6 & -6 & -6 & 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & -6 & -6 & -6 & -6 & -1 & 1 & 1 & 1 \\ 0 & 0 & -6 & -6 & -6 & -6 & -1 & 0 & 0 \\ 0 & 0 & -6 & -6 & -6 & -6 & -6 & 0 & 0 \\ 0 & 0 & -1 & -6 & -6 & -6 & -6 & 0 & 0 \\ 1 & 1 & 1 & -1 & -6 & -6 & -6 & -6 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 & -6 & -6 & -6 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & -6 & -6 \end{bmatrix},$$

where the elements of A_k^2 , $k = 1, 2, 3, 4$ are the corresponding weights of the fired neuron in layer 1 to its 9×9 corresponding neurons in layer 2. The potential $U_2(i, j)$ of the neuron of the layer 2 in the position (i, j) is the sum of the effects from the previous layer, i.e.

$$U_2(i, j) = \sum_{k=1}^4 \sum_{u=-4}^4 \sum_{v=-4}^4 O_1(i+u, j+v, k) A_k^2(5+u, 5+v) \quad (5)$$

where $O_1(i, j, k)$ is the output spike generated by the neuron with sensitive direction $l(k)$ in the position (i, j) of the layer 1. When $U(i, j)$ exceeds the threshold $thred_2 = 3$ and the neuron is not in its refractory period, the neuron fires a spike and turns into a refractory period lasting about $t_{re} = 4$. If its potential is less than 0, the neuron is also turned into the refractory period. Note that the size of the filter is bigger than that in Equ.(4), which make the process more fast. This is because the number of the cycles required is related to the speed of the information propagation, which is determined by the size of the spacial filter.

The weights of the feedback from the layer 2 to layer 1 are nearly same as $A_1^1, A_2^1, A_3^1, A_4^1$ except some small differences.

2.4 The Output Spikes of SNN

Layer 3 is the output layer. The neuron in the position (i, j) of the layer 3 receives the output of the layer 2 and its potential can be calculated by

$$U_3(i, j) = \sum_{v=-2}^2 \sum_{u=-2}^2 O_2(i+u, j+v) \quad (6)$$

with the output $O_2(i, j)$ comes from the neuron in the position (i, j) of the layer 2 with weights being 1. The neuron will fire when there are more than $thred_3(t)$ spikes generated by its corresponding neurons 5×5 in the layer 2, i.e. $U_3(i, j) > thred_3(t)$. Although U_3 is also a function of the time t , use $U_3(i, j)$ instead of $U_3(i, j, t)$ will not cause any misunderstanding. The threshold $thred_3(t)$ decrease with the discrete time t , because the number of neurons which fire decreases with t . Thus, at the beginning, the threshold should be high enough to filter out some noise, while the threshold should be low enough to obtain the information at the end. In the simulations, $thred_3(t)$ is chosen as

$$thred_3(t) = 20 - 2 \min(t, 4).$$

Obviously, smaller $thred_3(t)$ can retain more information. But small $thred_3(t)$ will not satisfy the requirement of information extraction and may make some noise survive.

2.5 The Recovery of the Image

The output spikes's temporal and spacial structure contains the extracted object information. In this section, one extra layer is used to recover the image from the information extracted by SNN. For further image processing, it is not a necessary to recover the image, which is just a test about the extracted information.

This extra layer is also $n \times m$ and the input of the recovery process is from t_{end} to $t = 1$, where t_{end} is the time when the image information extraction stops. At every time t , each neuron in the extra layer checks that whether there is a spike generated by the corresponding neuron in the output layer at time t . If there exists such spike or the neuron of the extra layer fires at time $t - 1$, then the potential $U_{rec}(i, j, t)$ of the neuron $N_{i,j}^{rec}$ is calculated by (notice that the input of the recovery process is from the end time to the beginning)

$$U_{rec}(i, j, t) = \sum_{u=-3}^3 \sum_{v=-3}^3 (O_{rec}(i, j, t-1) | O_{snn}(i, j, t_{end} - t + 1)) A_{rec}(4+u, 4+v) \quad (7)$$

with the space filter

$$A_{rec} = \begin{bmatrix} 0 & 2 & 2 & 2 & 2 & 2 & 0 \\ 2 & -2 & -2 & -2 & -2 & -2 & 2 \\ 2 & -2 & -6 & -6 & -6 & -2 & 2 \\ 2 & -2 & -6 & -6 & -6 & -2 & 2 \\ 2 & -2 & -6 & -6 & -6 & -2 & 2 \\ 2 & -2 & -2 & -2 & -2 & -2 & 2 \\ 0 & 2 & 2 & 2 & 2 & 2 & 0 \end{bmatrix}$$

where $|$ is the union operation and O_{rec}, O_{snn} are the outputs of the extra layer and SNN, respectively. If $U_{rec}(i, j, t)$

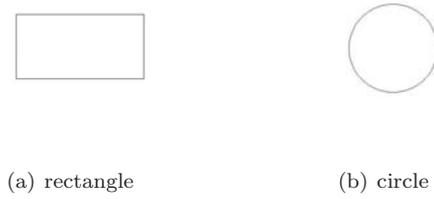


Fig. 3. The original images: rectangle and circle.

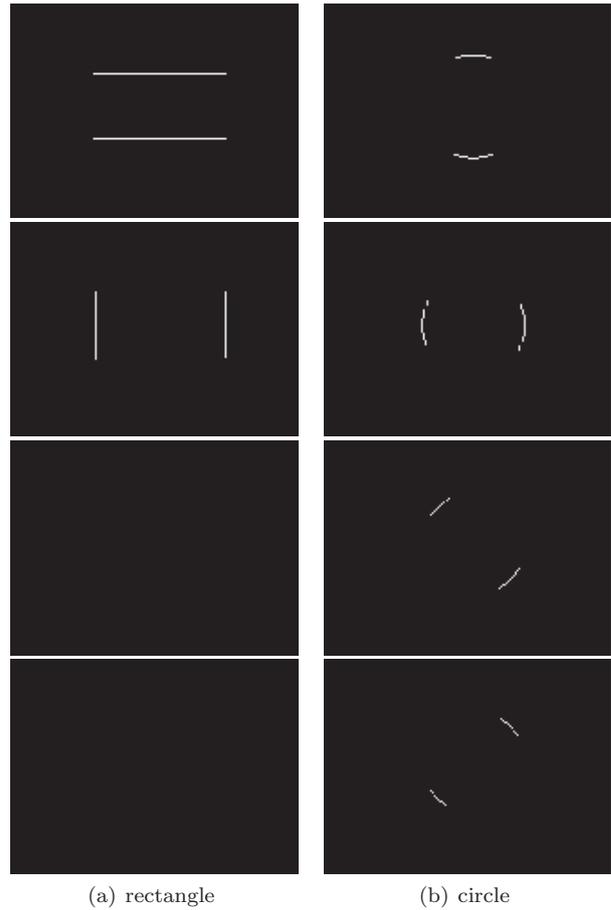


Fig. 4. The responses of neurons in layer 1 with direction selection to $0^\circ, 90^\circ, 45^\circ, 135^\circ$, respectively

exceeds the threshold, the neuron $N_{i,j}^{rec}$ fires. Also, the neuron turns into refractory period after it fires or its potential is below zero.

3. NUMERICAL SIMULATION

In this section, some simulations are used to demonstrate the image process and testify the proposed method. First, one rectangle and one circle are chosen as original image (shown in Fig. 3). The four figures of Fig. 4(a) and 4(b) show the outputs of the neurons in layer 1 at time $t = 1$ with sensitive directions $(0, 1, \infty, -1)$, respectively. In the figures, the white point represents the neuron fires while the black point means that the neuron dose not fire.

The neurons, which fire, propagate their spikes to the layer 2 in their preferred directions and the response of the layer

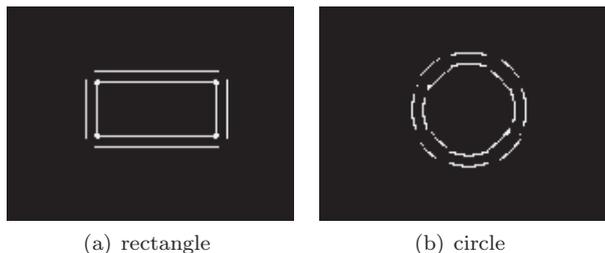


Fig. 5. The responses of neurons in layer 2

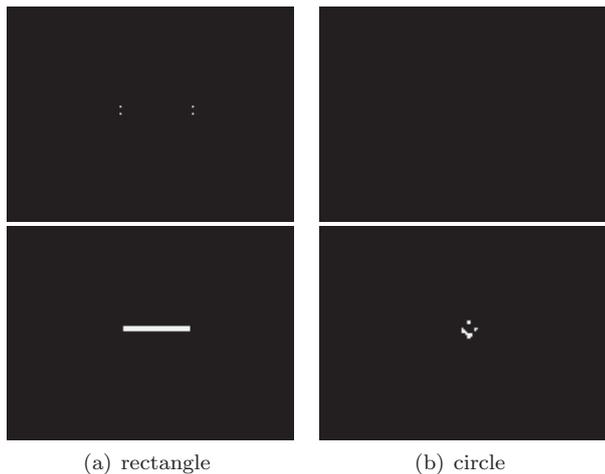


Fig. 6. The responses of neurons in layer 3 at $t = 5, 6$

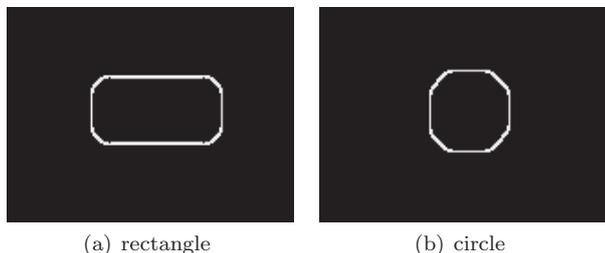


Fig. 7. The responses of neurons in the extra layer at time $t = t_{end}$.

2 is shown in Fig. 5. Also, the layer 2 feeds its output back to the layer 1, and this process continues until the information extraction is finished.

The response of the layer 3 at time $t = 5, 6$ is shown in Fig. 6. At the other time, there is no neuron firing.

Remark 2. The output of layer 3 is similar to the skeleton of graph. However, there is an essential difference between them: the time when the spike are generated contains the information about the scale of the object, while skeleton of graph has no such temporal structure.

The states of the neurons at time $t = t_{end}$, i.e. the results of the recovery, are shown in Fig. 7, compared to the original images.

Then, a more complex curve shown in Fig. 8 is used to test the results gained above. This closed curve contains not only the convex parts, but also the concave parts. The output of SNN at time $t = 3, 4, 5, 6, 7$ (at the other time

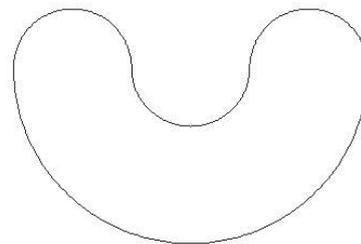


Fig. 8. The original image of the curve.

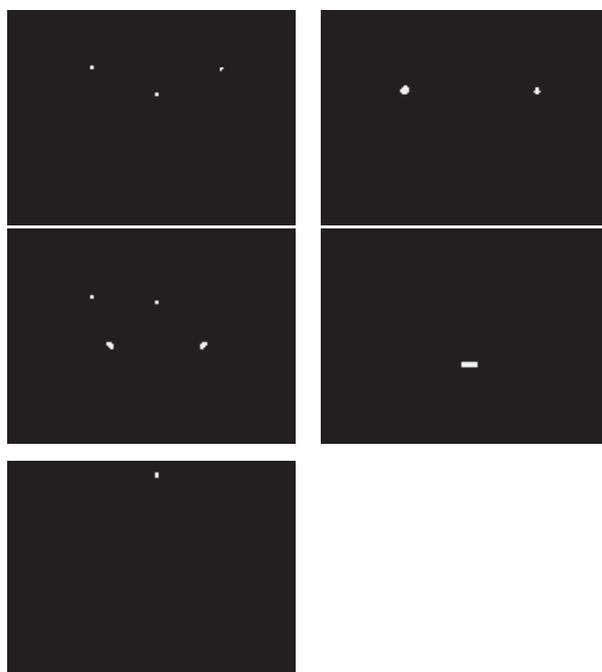


Fig. 9. The responses of neurons in the output layer at time $t = 3, 4, 5, 6, 7$, respectively. The order of the pictures is from left to right and top to down.



Fig. 10. The recovery of the curve.

there are no spikes generated at the output layer of SNN and the output is zero) is given in Fig. 9. Also, the recovery of the image is given in Fig.10.

Compared to the rectangle and the circle, the recovery of the curve is different from the original image, which is caused by the concave parts of the closed curve. However, the closed part of the curve in Fig. 10 includes the information needed to represent the curve, which can be done by simply deleting the unclosed parts of the curve.

The differences between the results of Fig. 7, 10 and the original images come from the following reasons:

- (1) the information reduction during the processing.
- (2) the inaccurate space filter such as A_{rec} .
- (3) only 4 kinds of direction-selective neurons are used.

From the above results, it can be found that some important characteristics of the original object have been extracted and encoded in the temporal and spacial structure of SNN's output. Although the recovery seems unperfect, it is not necessary to recover the image for the further image processing. The recovery of the image is just a test about the information extracted from the image, which is encoded in the temporal and spacial structure of the output, as shown in Fig. 6 and 9.

Above all, the information extracted from the rectangle, the circle and the curve by SNN is significantly different such that the three objects can be distinguished from each other easily. The results show that the extraction processing by SNN provides enough information for the future utility.

4. CONCLUSION

In this paper, a SNN based image information extraction approach is proposed to extract the image information. Not only the direction-selective the input neurons are, but also the outputs of these neurons are assumed to have a preferred direction. By using SNN, the information of the object is encoded in the temporal and spacial structure of the output spikes. In the future, the further image processing, such as object recognition, will be implemented based on the information extracted by SNN. One possible application would be the racket recognition for ping-pong robot, as the rough size of the racket is known such that the temporal information provided by the proposed SNN can be used to recognize the racket directly.

REFERENCES

- Bodyanskiy, Y. and Dolotov, A. (2008). Image processing using self-learning fuzzy spiking neural network in the presence of overlapping classes. *International Biennial Baltic Electronics Conference*, 213–216.
- Camastra, F. (2007). Image processing: principles and applications (book review). *IEEE Transactions on Neural Networks*, 18, 610–610.
- D. Banks, P.D. and Toulmazou, C. (2007). A bio-inspired adaptive retinal processing neuron with multiplexed spiking outputs. *IEEE International Symposium on Digital Object Identifier*, 2375–2378.
- Dayan, P. and Abbott, L. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press, Massachusetts.
- Delorme, A. and Thorpe, S. (2001). Face identification using one spike per neuron: resistance to image degradations. *Neural Networks*, 14, 795–803.
- Floreano, D. and Mattiussi, C. (2001). Evolution of spiking neural controllers for autonomous vision-based robots. *Evolution Robotics, Lecture Notes in Computer Science*, 2217, 38–61.
- G.H. Henry, B.D. and Bishop, P. (1974). Orientation specificity of cells in cat striate cortex. *Journal of Neurophysiology*, 34, 1394–1409.
- H. Burgsteiner, M. Kroll, A.L. and Steinbauer, G. (2005). Movement prediction from real-world images using a liquid state machine. *Proc. Lecture Notes In Computer Science, Proceedings of the 18th international conference on Innovations in Applied Artificial Intelligence*, 121–130.
- H. Qu, S.X. Yang, A.W. and Yi, Z. (2009). Real-time robot path planning based on a modified pulse-coupled neural network model. *IEEE Transactions on Neural Networks*, 20, 1724–1739.
- Joshi, P. and Maass, W. (2005). Movement generation with circuits of spiking neurons. *Neural Computation*, 17, 1715–1738.
- K. Zhan, H.Z. and Ma, Y. (2009). New spiking cortical model for invariant texture retrieval and image processing. *IEEE Transactions on Neural Networks*, 20, 1980–1986.
- Kiselev, M. (2009). Self-organized spiking neural network recognizing phase/frequency correlations. *International Joint Conference on Neural Networks*, 1633–1639.
- Maass, W. and Bishop, C. (1999). *Pulsed Neural Networks*. The MIT Press, Cambridge.
- Tal, D. and Schwartz, E. (1997). Computing with the leaky integrate and fire neuron: logarithmic computation and multiplication. *Neural Computation*, 9, 305–318.