Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 25 (2004) 1075–1084
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Image thinning using pulse coupled neural network

Xiaodong Gu [a,b,*], Daoheng Yu [b], Liming Zhang [a]

[a] Department of Electronic Engineering, Fudan University, 220 Handan Road, Shanghai, 200433, PR China
[b] Department of Electronics, Peking University, Beijing 100871, PR China

Received 18 March 2003; received in revised form 15 August 2003
Available online 21 April 2004

Abstract

PCNN-pulse coupled neural network, based on the phenomena of synchronous pulse bursts in the animal visual cortex, is different from traditional artificial neural networks. This paper first introduces a new approach for binary image thinning by using the pulse parallel transmission characteristic of PCNN. The thinning result obtains when pulses emitted by background meet. The criterion of pulse meeting and the criterion of thinning completion are proposed. The computer simulation results of applying the method to thin binary image are present. Comparisons of skeleton structure and execution time with results from other thinning methods are present too. The PCNN skeleton retains more information of original binary image, such as the size of a quadrate, than the result from Zhang and Suen method. The procedure is faster than Arcelli et al. thinning method when the image resolution is from 600 to1800 dpi. Combining with PCNN restoration algorithm (namely PCNN noise-reducing algorithm), the skeletons of the objects in a noisy binary image can be obtained with the accuracy. This paper also expands the application field of PCNN.

Keywords: PCNN; Binary image thinning; Skeleton

1. Introduction

Binary image thinning plays an important role in image processing such as handwriting recognition (Tellache et al., 1993), text recognition (Abuhaiba et al., 1996), fingerprint classification (Luk et al., 1991; Fitz and Green, 1995), data compression. The results of binary image thinning are skeletons, which represent the structural shape of original images by using less data. In general, the skeleton of an object may be defined as the set of all points equally distant from two boundary points by the medial axis transformation (MAT) introduced by Blum (1967). Shin and Pu (1995) redefine the MAT as the set of local maxima in the distance transform. A good thinning method should produce skeletons including the shape information of original object with the accuracy so that these skeletons are suit for application.

Many thinning algorithms have been introduced for decade years. These thinning algorithms have

* Corresponding author. Address: Department of Electronic Engineering, Fudan University, 220 Handan Road, Shanghai 200433, PR China. Tel.: +86-21-656-43883/86-10-627-55020; fax: +86-21-656-48783/86-10-627-51762.
E-mail address: guxiaodong@263.net (X. Gu).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.03.005

advantages and disadvantages respectively. Some algorithms can obtain good quality skeletons, but they run slowly (Lam et al., 1992). Some algorithms run fast, but resulting skeletons are not good.

Many new thinning approaches have been proposed. Pavlidis (1986) and Lin and Chen (1995) introduced methods to obtain skeletons from the run-length coding of images. Some algorithms are based on polygonal approximation of the boundary of regions. Melhi et al. (2001) proposed a novel method of binary text image thinning.

PCNN (pulse coupled neural network) is different from traditional neural networks. Lately, some researchers paid more attention to it. PCNN models have biological background. According to the phenomena of synchronous pulse bursts in the cat visual cortex, Eckhorn et al. (1990) developed the linking field network. When experimental objects are monkeys, the same results of experiments are also obtained (Eckhorn et al., 1993). Johnson and Ritter (1993) introduced PCNN based on the linking model. PCNN can be applied in many fields, such as image processing (Johnson and Padgett, 1999; Gu et al., 2001) object recognition (Ranganath and Kuntimad, 1999), optimization (Caufield and Kinser, 1999).

This paper first introduces a new approach for binary image thinning by using the pulse parallel transmission characteristic of PCNN. The skeleton obtains when pulses emitted by background meet. The binary images that have different shapes can be thinned naturally by using the approach proposed in this paper.

In Section 2 of this paper, the basic model of PCNN neuron is described simply. In Section 3, PCNN image thinning algorithm is described. The criterion of pulse meeting and the criterion of thinning completion are proposed. In Section 3, it is also discussed that how different connection modes of kernel matrix $\boldsymbol{K}$ obtain different kinds of skeletons. In Section 4, the results of tests conducted using several binary images as examples are present and a comparison of the PCNN thinning algorithm with some alternative methods is shown. In Section 4, the results also show that combining with PCNN restoration algorithm (Gu et al., 2001), the skeletons of the objects in a noisy binary image can be obtained with the accuracy. The conclusions are obtained in Section 5.

2. PCNN neuron model

A PCNN neuron consists of three parts: the receptive field, the modulation field, and the pulse generator (see Fig. 1).

The neuron receives input signals from other neurons and from external sources through the receptive field. In general, the signals from other neurons are pulses; the signals from external sources are analog timing-varying signals, constants, or pulses. After inputting the receptive field, input signals are divided into two channels. One channel is feeding input (F); the other is linking input (L). In Fig. 1, I_j and J_j are inputs from external sources. In modulation field (see Fig. 1), the linking input is added a constant positive bias firstly. Then it is multiplied by the feeding input and the bias is taken to be unity (see Eq. (1)). β_j is the linking strength. The total internal activity U_j is the result of modulation and it is inputted to the pulse generator. If U_j is greater than the threshold θ_j, the neuron output Y_j turns into 1 (namely the neuron j fires (see Eq. (2))). Then Y_j feedbacks to make θ_j rises over U_j immediately so that Y_j turns into 0. Therefore, when U_j is greater than θ_j, neuron j outputs a pulse:

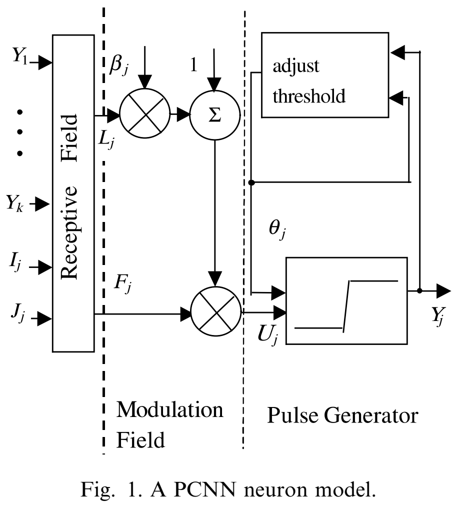


Fig. 1. A PCNN neuron model.

$$U_j = F_j(1 + \beta_j L_j), \tag{1}$$

$$Y_j = \mathrm{Step}(U_j - \theta_j) = \begin{cases} 1, & U_j > \theta_j \\ 0, & \text{else} \end{cases}. \tag{2}$$

3. Binary image thinning using PCNN

A PCNN consists of these neurons one of which is show in Fig. 1. When PCNN is applied in binary image thinning, it is a single layer two-dimensional array of laterally linked neurons. The number of neurons in the network is equal to the number of pixels in the input image. One-to-one correspondence exists between image pixels and neurons. Each pixel is connected to a unique neuron and each neuron is connected with the surrounding neurons. Each pixel's intensity is inputted to the F channel of the neuron which is connected with it (see Eq. (3)). Meanwhile, each neuron is connected with neurons by the L channel in its 4-neighbor field (see Eq. (4)). Eq. (4) indicates that $L_j = 1$ if one or more than one neuron fire in the 4-neighbor field of neuron j. Each neuron's output has two states, 1 (firing) or 0 (no firing).

$$F_j = S_j, \tag{3}$$

$$L_j = \begin{cases} 1, & \text{if } \sum_{k \in N(j)} Y_k > 0. \\ 0, & \text{else} \end{cases} \tag{4}$$

In Eq. (3), S_j is the intensity value of neuron j. In Eq. (4), $N(j)$ is the neighbor field of neuron j. Note, j does not belong to $N(j)$.

The Fig. 2 shows how one neuron is arranged in the PCNN that is used to do binary image thinning.

Eqs. (1)–(4) describe how each neuron operates and is arranged in PCNN for binary image thinning.

The value of threshold θ_j is constant A when neuron j has not fired, and is constant B when neuron j has fired. Using in binary image thinning, each neuron in PCNN has the same A and B. B is large enough so that the neuron fired will never fire again.

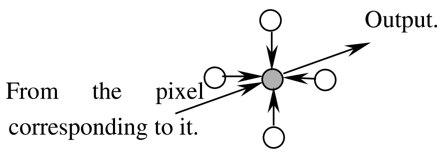


Fig. 2. The connection mode of each neuron in the PCNN for thinning.

In a binary image, in general, the bright region corresponds to the background and the dark region corresponds to the object, such as characters. Because the intensity of the bright region is larger than that of the dark region, by choosing suitable threshold (namely choosing suitable A and B), we can make all neurons corresponding to pixels in the bright region (namely background) fire firstly, and in the mean time make all neurons corresponding to pixels in the dark region (namely the object) not fire.

Suitable A should not only make all neurons corresponding to pixels in the bright region fire, make all neurons corresponding to pixels in the dark region (namely the object) not fire at the beginning, but also make the neurons corresponding to pixels in the dark region fire when they receive pulses emitted by neurons in their 4-neighbor field by modulation (see Eq. (1)). Suitable B should make every neuron fire only once. Note, each neuron only fires once.

When a neuron in bright region emits a pulse (namely fires), this pulse makes other non-fired neurons in its 4-neighbor field fire by modulation (see Eq. (1)). Pulses emitted by bright region spread all over the dark region. Therefore, the background (the bright region) emits pulses as if the boundary of the object (the dark region) emits pulses and these pulses runs all over the object (the dark region). The thinning result obtains when these pulses meet. Using this pulse parallel transmission characteristic of PCNN can thin binary image naturally and automatically.

Next, PCNN image thinning algorithm is introduced. Meanwhile, the criteria of pulse meeting is proposed, so is the criterion of thinning completion.

3.1. PCNN image thinning algorithm

In PCNN image thinning algorithm, the thinning result obtains when the pulses emitted by the background meet.

How to know the pulses emitted by the background meet? Now, the criterion of pulse meeting is present below.

If a neuron fired at previous time and no neuron in its 4-neighbor field fires at current time, pulses meet and the pixel corresponding to the neuron belongs to the set of thinning result.

How to know the completion of thinning process? It is obvious that when all neurons have fired, the thinning process is accomplished. This is the criterion of thinning completion. For a binary image with limited pixels, only has it bright region, this is guaranteed because of pulse parallel transmission characteristic of PCNN.

These two criteria are used in PCNN image thinning algorithm so that the thinning result can obtain automatically.

Before PCNN image thinning algorithm is described, we introduced the symbols that will be used in the algorithm. $\boldsymbol{F}$, an original binary image, is a feeding input matrix. Each element of $\boldsymbol{F}$ is the intensity value of the corresponding pixel in the binary image. $\boldsymbol{L}$ is a linking input matrix. $\boldsymbol{L} = \mathrm{Step}(\boldsymbol{Y} \otimes \boldsymbol{K})$, where $\boldsymbol{Y}$ is a binary output matrix recording the firing states of all neurons at current time and $\boldsymbol{K}$ is a $3*3$ kernel matrix,

$$\boldsymbol{K} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

$\boldsymbol{U}$ is an internal activity matrix. $\boldsymbol{\theta}$ is a threshold matrix and all elements of $\boldsymbol{\theta}$ are equal, $\boldsymbol{\theta} = 0.15$. $\boldsymbol{Y}_{\mathrm{pre}}$ is a binary output matrix recording the firing states of all neurons at previous time and it will be used when we decide if pulses meet. **Res** is a binary output matrix saving thinning result. $\boldsymbol{F}$, $\boldsymbol{L}$, $\boldsymbol{U}$, $\boldsymbol{Y}$ $\boldsymbol{Y}_{\mathrm{pre}}$, $\boldsymbol{\theta}$, **Res** have the same dimensions. '$.*$' indicates array multiplication and '$\otimes$' indicates two-dimensional convolution. β is the linking strength and each neuron has the same β, $\beta = 1$.

PCNN image thinning algorithm is described below.

(1) $\boldsymbol{L} = \boldsymbol{U} = \boldsymbol{0}$, $\boldsymbol{Y} = \boldsymbol{Y}_{\mathrm{pre}} = \boldsymbol{0}$, $\boldsymbol{\theta} = 0.15$, $\beta = 1$. Each element of $\boldsymbol{F}$ is adjusted to 0.1 or 1. Note, the element of $\boldsymbol{F}$ may not be 0.
(2) $\boldsymbol{L} = \mathrm{Step}(\boldsymbol{Y} \otimes \boldsymbol{K})$;
$\boldsymbol{U} = \boldsymbol{F}.*(\boldsymbol{1} + \beta\boldsymbol{L})$;
$\boldsymbol{Y} = \mathrm{Step}(\boldsymbol{U} - \boldsymbol{\theta})$.
(3) Adjust threshold, $\boldsymbol{\theta} = \boldsymbol{\theta} + 100 * \boldsymbol{Y}$. If a neuron has fired, its threshold increases so that it will never fire again.
(4) According to $\boldsymbol{Y}$, $\boldsymbol{Y}_{\mathrm{pre}}$, use the criterion of pulse meeting to obtain pixels belonging to the set of thinning result, save these pixels to **Res**.
(5) $\boldsymbol{Y}_{\mathrm{pre}} = \boldsymbol{Y}$. $\boldsymbol{Y}_{\mathrm{pre}}$ is used in the criterion of pulse meeting.
(6) Use the criterion of thinning completion. If thinning is complete, go to the next step, else go back to step (2).
(7) **Res** is the last thinning result.

Each neuron only fires once, namely if a neuron has fired, it will never fire again. In the algorithm, adjusting threshold carries it out. In this algorithm, $A = 0.15$ (see $\boldsymbol{\theta} = 0.15$ in step 1), $B = 100.15$ (see $\boldsymbol{\theta} = \boldsymbol{\theta} + 100 * \boldsymbol{Y}$ in step 3).

At the beginning, all neurons corresponding to pixels in the bright region (namely background) fire because of $1 > 0.15$, all neurons corresponding to pixels in the dark region (namely the object) not fire because of $0.1 < 1$.

The neuron in dark region that does not fire at the first time fire when one or more than one neurons in its 4-neighbor field fire so that pulses emitted by the bright region can spread all over the object (the dark region). When these pulses meet, the thinning result obtains.

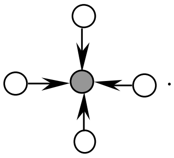
Combining the pulse parallel transmission characteristic of PCNN with two criteria (the criterion of pulse meeting and the criterion of thinning completion), we can obtain thinning results of binary images automatically.

3.2. Connection mode of kernel matrix $\boldsymbol{K}$

Connection mode of each neuron in its 8-neighbor field can be changed easily by modifying kernel matrix $\boldsymbol{K}$. In $\boldsymbol{K}$, element 1 indicates two corresponding neurons are connected, element 0 indicates two corresponding neurons are not connected. Different kernel matrixes correspond to different thinning results. $K(2,2) = 0$ in all kernel matrixes because in this paper each neuron's neighbor field does not include itself.

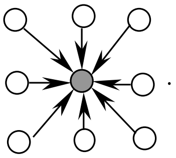
When 4-neighbor field connection mode is adopted,

$$\boldsymbol{K} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \to$$



When 8-neighbor field connection mode is adopted,

$$\boldsymbol{K} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \to$$



Why is 4-neighbor field connection mode adopted, but is not 8-neighbor field connection mode adopted for thinning? Now an example is shown in Fig. 3.

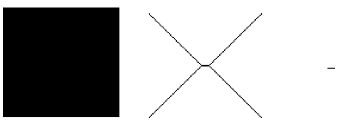
Fig. 3(a) is a rectangle. Fig. 3(b) is the 4-neighbor field PCNN thinning result of Fig. 3(a) and Fig. 3(c) is the 8-neighbor field PCNN thinning result of Fig. 3(a). These figures illustrate when 4-neighbor field connection mode is adopted, the thinning result not only retain the position information but also the size information of the original rectangle. When 8-neighbor field connection mode is adopted, the thinning result only retains the position information, but loses the size information of the original rectangle. The 4-neighbor field skeleton retains more information of original images than the 8-neighbor field skeleton.

Now, combining with the criterion of pulse meeting and the criterion of thinning completion, we analyze why this occurs. The criterion of pulse meeting and the criterion of thinning completion repeat here again. The criterion of pulse meeting: if a neuron fired at previous time and no neuron in its neighbor field fires at current time, pulses meet and the pixel corresponding to the neuron belongs to the set of thinning result. The criterion of thinning completion: when all neurons have fired, the thinning process is accomplished.

There is a $3 * 3$ gray quadrate (object) in $5 * 5$ Fig. 4(a). In Fig. 4, every grid indicates a pixel and a neuron corresponding to it. A number in a grid indicates the iteration times at which the corresponding neuron fires. When a pixel is decided to belong to the skeleton, the corresponding grid is filled with black color.

When 4-neighbor field connection mode is adopted, the thinning process of the quadrate in Fig. 4(a) is described below (see Fig. 4).

1. At the first iteration, the background neurons around the quadrate neurons fire (see Fig. 4(a)). Now according to the criterion of pulse meeting, no pixel is decided to belong to the skeleton.
2. At the second iteration, Fig. 4(b) illustrates the firing state at this time. No pixel is decided to belong to the skeleton.
3. At the third iteration, Fig. 4(c) illustrates the firing state at this time. The four apexes of the quadrate are decided to belong to the skeleton.



(a) (b) (c)

Fig. 3. A rectangle is shown in (a). The 4-neighbor field PCNN thinning result is shown in (b), and the 8-neighbor field PCNN thinning result is shown in (c).

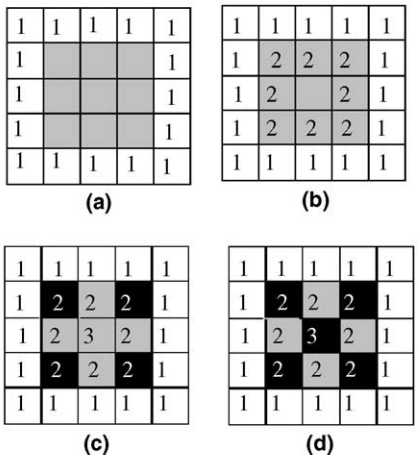


Fig. 4. When 4-neighbor field connection mode is adopted, the firing states of all neurons at first iteration, second iteration, third iteration, and fourth iteration, are shown in (a)–(d) respectively.

Therefore these four grids are filled with black color.

4. At the fourth iteration (see Fig. 4(d)), because all neurons have fired and each neuron can only fire once, no neuron fire at this time. The center pixel is decided to belong to the skeleton, so it is filled with black color. According to the criterion of thinning completion, the thinning process finishes.

Fig. 4 illustrates that the 4-neighbor field skeleton retains the position information and the size information of the original quadrate well.

When 8-neighbor field connection mode is adopted, the thinning process is described below (see Fig. 5).

1. At the first iteration, the background neurons around the quadrate neurons fire (see Fig. 5(a)). Now according to the criterion of pulse meeting, no pixel is decided to belong to the skeleton.
2. At the second iteration, Fig. 5(b) illustrates the firing state at this time. No pixel is decided to belong to the skeleton.
3. At the third iteration, Fig. 5(c) illustrates the firing state at this time. No pixel is decided to belong to the skeleton.

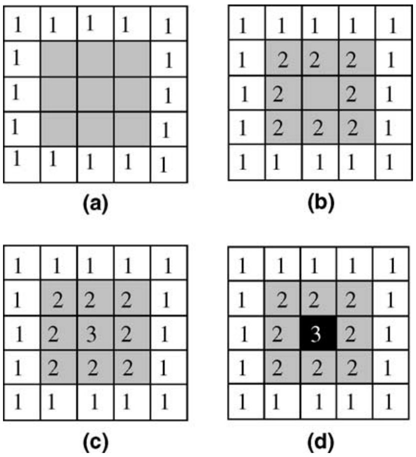


Fig. 5. When 8-neighbor field connection mode is adopted, the firing states of all neurons at first iteration, second iteration, third iteration, and fourth iteration, are shown in (a)–(d) respectively.

4. At the fourth iteration, see Fig. 5(d), because all neurons have fired, no neuron fire at this time. The center pixel is decided to belong to the skeleton, so it is filled with black color. According to the criterion of thinning completion, the thinning process finishes.

The 8-neighbor field skeleton of the quadrate is a point, and only retains the position information, but loses the size information of the original quadrate. No matter how large the quadrate is, the same result is obtained.

Therefore, in Fig. 3, the 4-neighbor field skeleton retains the position information and the size information of the original rectangle well, and the 8-neighbor field skeleton is only a short line. When 8-neighbor field connection mode is adopted, the larger the difference between the rectangle width and the rectangle height, the longer the skeleton line. In order to retain more information of original binary image, 4-neighbor field connection mode is adopted for thinning in this paper.

There are other connection modes except two connection modes mentioned above, such as horizontal connection mode, vertical connection mode.

When horizontal connection mode is adopted,

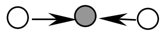
$$\boldsymbol{K} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \rightarrow$$  .

Fig. 6(a) is the PCNN horizontal thinning result of Fig. 3(a).

When vertical connection mode is adopted,

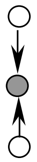
$$\boldsymbol{K} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \rightarrow$$  .

Fig. 6. The PCNN horizontal thinning result of Fig. 3(a) is shown in (a), and the PCNN vertical thinning result of Fig. 3(a) is shown in (b).

Fig. 6(b) is the PCNN vertical thinning result of Fig. 3(a).

In PCNN image thinning algorithm, we can easily obtain different thinning results by change kernel matrix $\boldsymbol{K}$. Because 4-neighbor field skeleton retains the information of original binary well, 4-neighbor field connection mode is adopted for thinning in this paper.

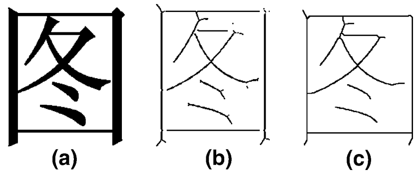


Fig. 9. A binary Chinese character is shown in (a). The 4-neighbor field PCNN thinning result is shown in (b), and the thinning result of Zhang and Suen algorithm is shown in (c).

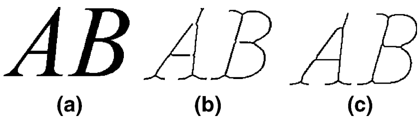


Fig. 10. Two binary English characters are shown in (a). The 4-neighbor field PCNN thinning result is shown in (b), and the thinning result of Zhang and Suen is shown in (c).

4. Results of computer simulations

The PCNN image thinning algorithm described in Section 3 was coded in C and applied to 500 binary images, including English words, Chinese words, and other images.

Figs. 7–10 show samples of the results for several binary images thinned by using PCNN image thinning algorithm (4-neighbor field connection mode), a traditional parallel algorithm (Zhang and Suen, 1984).

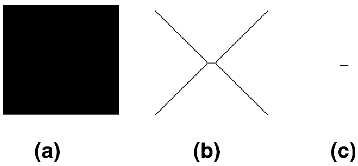


Fig. 7. A rectangle is shown in (a). The 4-neighbor field PCNN thinning result is shown in (b), and the thinning result of Zhang and Suen method is shown in (c).

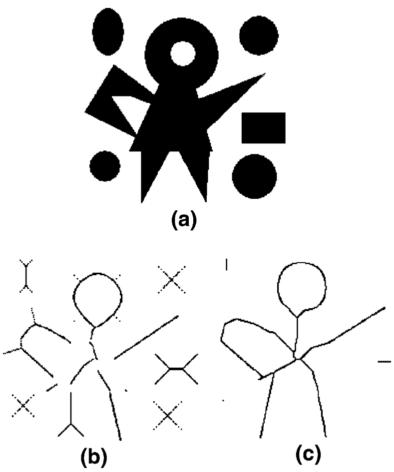


Fig. 8. A binary image is shown in (a). The 4-neighbor field PCNN thinning result is shown in (b), and the thinning result of Zhang and Suen algorithm is shown in (c).

Fig. 7(b) illustrates that the PCNN thinning result not only retain the position information but also the size information of the original rectangle in Fig. 7(a). PCNN thinning result retains the shape information of original object with the accuracy. The thinning result of Zhang and Suen method only retains the position information, but loses the size information of the original rectangle (see Fig. 7(c)). The PCNN skeleton retains more information of original images than the thinning result of Zhang and Suen method.

Fig. 8(b) illustrates that the PCNN thinning result retain the information of all parts of the object in Fig. 8(a) well. Fig. 8(c) for the thinning result of Zhang and Suen method loses all information of three circles in Fig. 8(a). In Fig. 8(c), the skeletons of a rectangle and an ellipse are two short lines, and the other information of the rectangle and the ellipse is lost.

By using PCNN image thinning algorithm, Zhang and Suen thinning algorithm, Arcelli et al. thinning algorithm (Arcelli et al., 1975) respectively, the times taken to obtain skeletons of characters in Fig. 10(a) are shown in Table 1 for image resolutions range from 300 dpi to 1800 dpi. All skeletons were calculated using a PIII 600

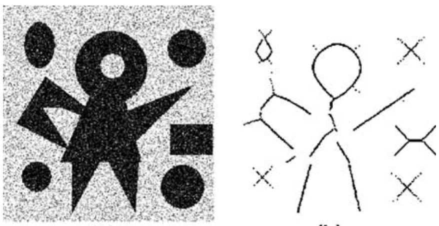
Table 1
Times in milliseconds taken to thin the characters in Fig. 10(a) using three different thinning methods

Resolution (dpi)	PCNN	Zhang and Suen	Arcelli et al.
300	770	130	387
600	3014	810	3534
900	7784	2905	8532
1200	12,388	6018	17,454
1500	23,244	11,247	32,645
1800	34,940	19,018	60,234

MHz computer. Zhang and Suen thinning algorithm is faster than other algorithms. At the lowest resolution of 300 dpi, Arcelli et al. thinning algorithm are faster than PCNN image thinning algorithm. From 600 to 1800 dpi, PCNN image thinning algorithm is faster than Arcelli et al. thinning algorithm. These trends were exhibited by all 500 binary images.

Noise has been a challenge for tinning. PCNN also can be used to reduce noise. Combining with PCNN restoration algorithm (namely PCNN noise-reducing algorithm, Gu et al., 2001), the skeletons of the objects in a noisy binary image can be obtained. For noisy binary image, first using PCNN noise-reducing algorithm reduces the noise. Next using PCNN thinning algorithm obtains the thinning result. Reducing image noise by using PCNN noise-reducing algorithm, according to each neuron's output and its neighbors' outputs, the intensity of each noisy neuron is adjusted (Gu et al., 2001). PCNN thinning algorithm can include PCNN noise-reducing algorithm.

Fig. 11(a) is a white Gaussian noisy image (SNR = 3 dB), Fig. 11(b) is the thinning result using PCNN thinning algorithm including PCNN noise-reducing algorithm. Compare Fig. 11(b) with Fig. 8(b), the skeleton of the ellipse in Fig. 11(b) is different from that in Fig. 8(b), and the skeletons of other parts in Fig. 11(b) are almost as the same as those in Fig. 8(b).

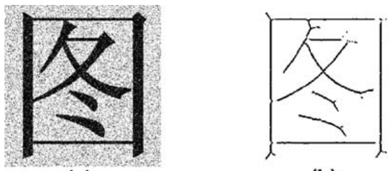


(a) (b)

Fig. 11. A noisy binary image is shown in (a) and the thinning result using PCNN thinning algorithm including PCNN noise-reducing algorithm is shown in (b).

Fig. 12(a) is a white Gaussian noisy Chinese character (SNR = 3 dB), Fig. 12(b) is the thinning result using PCNN thinning algorithm including PCNN noise-reducing algorithm. Figs. 11 and 12 illustrate that using PCNN thinning algorithm including PCNN noise-reducing algorithm, the skeletons of the objects retain the shape information of original noisy objects with the accuracy.

For noisy binary image, using PCNN thinning algorithm excluding PCNN noise-reducing algorithm, some small solid or dashed loops appear in resulting skeletons. Fig. 13(a) is a white Gaussian noisy image (SNR = 9 dB). Fig. 13(b) is the thinning result using PCNN thinning algorithm excluding PCNN noise-reducing algorithm, and Fig. 13(c) is the thinning result using Zhang and Suen algorithm. In Fig. 13(b), PCNN thinning algorithm excluding PCNN noise-reducing algorithm produces several solid or dashed loops because of noise. In Fig. 13(c) Zhang and Suen



(a) (b)

Fig. 12. A noisy binary Chinese character is shown in (a) and the thinning result using PCNN thinning algorithm including PCNN noise-reducing algorithm is shown in (b).

(a) (b) (c)

Fig. 13. Two noisy binary English characters are shown in (a). The thinning result using PCNN thinning algorithm excluding PCNN noise-reducing algorithm is shown in (b), and the thinning result of Zhang and Suen is shown in (c).

algorithm produces several solid loops adhering to the resulting skeleton because of noise too.

Therefore, in order to obtain good skeletons of noisy objects with the accuracy, PCNN noise-reducing algorithm should be adopted in PCNN thinning algorithm.

5. Conclusion

This paper first introduced the PCNN image-thinning algorithm based on the pulse parallel transmission characteristic of PCNN.

The results of computer simulations show that using this algorithm can thin binary image naturally and produce skeletons with the accuracy. Pulses emitted by PCNN neurons spread over networks in the parallel mode, so thinning process is fast. Meanwhile, the criterion of pulse meeting and the criterion of thinning completion are introduced so that using PCNN can thin binary image automatically. The PCNN thinning result retains the shape information of original objects with the accuracy. The PCNN skeleton retains more information of original binary image, such as the size of a quadrate, than the result from Zhang and Suen method. The procedure is faster than Arcelli et al. thinning method when the image resolution is from 600 to 1800 dpi.

For noisy binary image, using PCNN thinning algorithm including PCNN noise-reducing process, the skeletons of the objects in a noisy binary image can be obtained with the accuracy.

For grey level image, first we can use PCNN to segment the original grey level image into binary image (Kuntimad and Ranganath, 1999 and Gu et al., 2002). Then using PCNN thinning algorithm obtains the thinning result.

In addition, this paper also expands the application range of PCNN.

Acknowledgements

This work was supported by China Postdoctoral Science Foundation (No. 2003034282) and National 863 Foundation (No. 2002AA783060) respectively.

References

Abuhaiba, I., Holt, M., Datta, S., 1996. Processing of binary images of handwritten text documents. Pattern Recogn. 29, 1161–1177.

Arcelli, C., Cordella, L., Levialdi, S., 1975. Parallel thinning of binary pictures. Electron. Lett. 11, 148–149.

Blum, H., 1967. A transformation for extracting new descriptors of shape. In: Wathen-Dunn, W. (Ed.), Models for the Perception of Speech and Visual Form. MIT Press, Cambridge, MA.

Caufield, H.J., Kinser, J.M., 1999. Finding shortest path in the shortest time using PCNN's. IEEE Trans. Neural Networks 10, 604–606.

Eckhorn, R., Frien, A., Bauer, R., et al., 1993. High frequency oscillations in primary visual cortex of awake monkey. Neuro. Rep. 4, 243–246.

Eckhorn, R., Reitboeck, H.J., Arndt, M., et al., 1990. Feature linking via synchronization among distributed assemblies: Simulation of results from cat cortex. Neural Comput. 2, 293–307.

Fitz, A., Green, R., 1995. Fingerprint Pre-Processing on a hexagonal grid. In: Proc. Eur. Convention on Security and Detection, 1995. Brighton, UK, pp. 257–260.

Gu, X.D., Wang, H.M., Yu, D.H., 2001. Binary image restoration using pulse coupled neural network. In: The 8th Internat. Conf. Neural on Information Processing, Shanghai, China, 922–927.

Gu, X.D., Guo, Sh.D, Yu, D.H., 2002. A New approach for automated image segmentation based on unit-linking PCNN. The First Internat. Conf. on Machine learning and Cybernetics, Beijing, China, pp. 175–178.

Johnson, J.L., Padgett, M.L., 1999. PCNN models and applications. IEEE Trans. Neural Networks 10, 480–498.

Johnson, J.L., Ritter, D., 1993. Observation of periodic waves in a pulse-coupled neural network. Opt. Lett. 18 (15), 1253–1255.

Kuntimad, G., Ranganath, H.S., 1999. Perfect image segmentation using pulse coupled neural networks. IEEE Trans. Neural Networks 10 (3), 591–598.

Lam, L., Lee, S., Suen, C.Y., 1992. Thinning methodologies—a comprehensive survey. IEEE Trans. Pattern Anal. Machine Intell. 14, 869–885.

Lin, J., Chen, Z., 1995. A Chinese-character thinning algorithm based on global features and contour information. Pattern Recogn. 28, 493–512.

Luk, A., Leung, S., Lee, C., Lau, W., 1991. A two-level classifier for fingerprint recognition. In: Proc. IEEE Internat. Symp.on Circuits and systeme, Singapore, pp. 2625–2628.

Melhi, M., Ipson, S.S., et al., 2001. A noveltriangulation procedure for thinning hand-written text. Pattern Recogn. Lett. 22, 1059–1071.

Pavlidis, T., 1986. A vectorizer and feature extractor for document recognition. Comput. Vision Graphics Image Process. 35, 111–127.

Ranganath, H.S., Kuntimad, G., 1999. Object detection using pulse coupled neural networks. IEEE Trans. Neural Networks 10, 615–620.

Shin, F.Y., Pu, C.C., 1995. A skeletonization algorithm by maxima tracking on Euclidiean distance transform. Pattern Recogn. 28, 331–341.

Tellache, M., Sid-Ahmed, M., Anaza, B., 1993. Thinning algorithms for Arabic OCR. Commum. Comput. Signal Process. IEEE Pacific Rim, 248–251.

Zhang, T.Y., Suen, C.Y., 1984. A fast parallel algorithm for thinning digital patterns. Communications of the ACM 27 (3), 236–239.