2009 Eigth IEEE/ACIS International Conference on Computer and Information Science

A Novel Approach Based on PCNNs Template for Fingerprint Image Thinning

Dacheng Xu[a,c], Bailiang Li[b]
University of Soochow, China

Anton Nijholt [c]
University of Twente, The Netherlands

Abstract

A PCNNs-based square-and-triangle-template method for binary fingerprint image thinning is proposed. The algorithm is iterative, in which a combined sequential and parallel processing is employed to accelerate execution. When a neuron satisfies the square template, the pixel corresponding to this neuron will be noted during the process and be deleted until the end of the iteration; on the other hand, if a neuron meets a triangle template, it will be removed directly. In addition, this proposed algorithm can be effective for fingerprint thinning without considering the direction. The results showed that, with combined sequential and parallel conditions for border pixels removal, the algorithm could not only speed up the fingerprint thinning process, but also be applied to other common images. Furthermore, this algorithm might be applied to fingerprint identification systems to save the time for identifying and eliminating spurious minutiae.

Key words: PCNN, square template, triangle template, iteration, fingerprint image thinning.

1 Introduction

Fingerprint thinning process is now widely used to delete the pixels on the boundary of an image without disconnecting the neighboring pixels. It is generally accepted that fingerprint thinning should have the following characteristics: (1) Preserving the original connectivity; (2) Obtaining the centre skeleton of the original image; (3) Thorough thinning; (4) Keeping the skeleton intact. Image thinning is playing a more and more important role in both digital image processing and pattern recognition, such as character recognition. Besides, it can also be utilized in biomedical systems, especially for AFIS. Up to now, some thinning algorithms have been proposed [1], [2] and [3]. However, most of them are limited when being utilized to fingerprint thinning process.

According to the different iteration methods in the literature, we concluded that the present fingerprint thinning algorithms could be sorted into two categories: one is sequential algorithm [6], [7] and the other one is parallel algorithm [8], [9] and [10]. In sequential algorithm, the pixel as a boundary point is deleted directly from the image. While in parallel algorithm, if a pixel is on the edge of image, it will be flagged and not deleted until the entire image has been scanned. Based parallel conditions for thinning algorithms have been published in many literatures. In AFIS, two primary thinning algorithms are based on mathematical morphology and OPTA (one pass thinning algorithm) [12], which are parallel algorithms.

In addition to the above algorithms, some methods based on pulse-couple neural network (PCNN) [3], [4] and [5] have also been reported. However, in AFIS, some ridge spikes are often generated during thinning but [3]. In literature [3], binary fingerprint image thinning using template-based PCNNs was proposed. Although satisfied results were got in common binary images, they obtained discontinuous ridges. Moreover, the direction of fingerprint is depended on in this algorithm. In this study, PCNN based on square and triangle templates was applied to fingerprint thinning without relying on the direction of fingerprint, tests on FVC2004 database. Sequential and parallel conditions were combined for image thinning. Using template-based PCNNs, the algorithm extracts skeleton iteratively with neuron pulses. Experiment results showed that the proposed algorithm is fast in terms of thinning speed and can obtain high thinning rate. In addition, the method can depress noise and promote the robustness of the minutiae extraction algorithm. The proposed algorithm could be applied to a real fingerprint identification system to improve the recognition performance and to reduce a number of spurious minutiae, as well as the memory space required for storing image in AFIS.

The structure of the paper is organized as follows. The template-based PCNN is described in Section Ⅱ. Section Ⅲ introduces the proposed fingerprint image thinning algorithm. Experiments are given in Section Ⅳ. Finally, the conclusions are summarized in Section Ⅴ.

2 Template-based PCNN

A PCNN neuron consists of three parts [14], [15]: the receptive field, the modulation field, and the pulse generator (see Fig. 1). In [3] and [4], the proposed PCNN is a single-layer 2-D NN, and all neurons in this network are identical to each other. There exists a one-to-one relationship between a neuron and an image pixel. The Template-based PCCN model is illustrated in Fig.2. In this model, X_{ij} denotes the load image, F_{ij} denotes the feeding input, L_{ij} denotes the linking input of the neuron (i, j), β_{ij}, U_{ij}, θ_{ij},

[a] School of Applied Technology, Suzhou 215006, China
xudacheng@suda.edu.cn
[b] School of Electronic Information,Suzhou 215006, China
[c] Human Media Interaction, P.O. Box 217, 7500 AE Enschede, The Netherlands

978-0-7695-3641-5/09 $25.00 © 2009 IEEE
DOI 10.1109/ICIS.2009.132

Y_{ij} denote the linking strength, the internal activation, the neuron's threshold, and the pulse output signal of neuron, respectively. M_{kmn} is a coupled templates, and 'k' is the index of templates,1≤k≤16. W_{mn} (1≤m≤3, 1≤n≤3) denotes the linking weight between linking neuron (i, j) and neuron (i+m-2, j+n-2). We define ϕ_{mn} as each linking input.

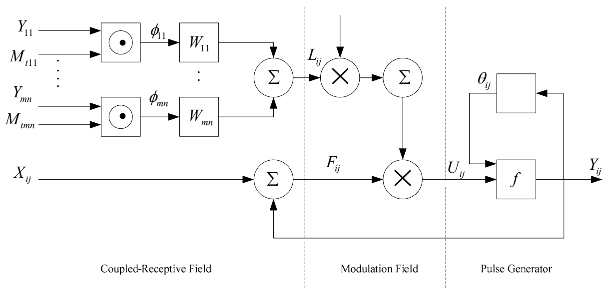


Fig.1. The template-based PCCN model

In Fig.1, feeding input $F_{ij}(t)$ is calculated as follows:

$$F_{ij}(t) = X_{ij}(t) + Y_{ij}(t) \tag{1}$$

Where at the first iteration (t=0), $Y_{ij}(t)$ is initialized zero matrix, thus $F_{ij}(t)$ equals to $X_{ij}(t)$. Then at t≥1, $X_{ij}(t)$ is closed in order to make $F_{ij}(t)$ equal to the feedback $Y_{ij}(t)$.

The internal activation $U_{ij}(t)$ is calculated by

$$U_{ij}(t) = F_{ij}(t)[1 + \sum_{m=1}^{3}\sum_{n=1}^{3} W_{mn}\phi_{mn}] \tag{2}$$

Each linking input $\phi_{mn}(t)$ (1≤m≤3, 1≤n≤3) is calculated by the following equation (3). The linking weight $W_{mn}(t)$ is represented as equation (4).

$$\phi_{mn}(t) = \begin{cases} 1, if\ M_{kmn} = Y_{mn}(t)\ and\ m \neq n \\ 0, others \end{cases} \tag{3}$$

$$W_{mn}(t) = \begin{cases} 1, if\ M_{kmn} \geq 0\ and\ m \neq n \\ 0, others \end{cases} \tag{4}$$

From equation (4), we can see $W_{mn}(t)$ is determined by M_{kmn} , whose value is '0', '1' or 'x', here 'x' denotes don't care and can assume either a '0', or a '1'. In the template matrixes, M_{kmn} =0 when m=n, that is the centre pixels of template matrixes are zero.

The neuron's threshold $\theta_{ij}(t)$ is calculated as

$$\theta_{ij}(t) = 8 - x \tag{5}$$

where 'x' is the total number of element 'x'.

The neuron output $Y_{ij}(t)$ is calculated by

$$Y_{ij}(t) = Step(U_{ij}(t) - \theta_{ij}(t)) = \begin{cases} 1, if\ U_{ij}(t) \geq \theta_{ij}(t) \\ 0, others \end{cases} \tag{6}$$

where Step(·) is the pulse generation function of neuron. When $U_{ij}(t)$ is greater than $\theta_{ij}(t)$, the neuron output $Y_{ij}(t)$ turns into 1 (namely the neuron (i, j) fires (see Eq. (6))). Then $\theta_{ij}(t)$ rises over $Y_{ij}(t)$ immediately so that $Y_{ij}(t)$ turns into '0' at the next iteration. Therefore, when $U_{ij}(t)$ is greater than or equal to $\theta_{ij}(t)$, neuron (i, j) outputs a pulse.

3 The proposed template-based PCNN thinning algorithm

To introduce the two different kinds of templates is helpful to get a better understanding about the algorithm: one is square template and the other is triangle template (Fig.2 and Fig.3). In Fig.2 and Fig.3, '0' and '1' represent object neurons and background neurons, respectively. In Fig.2 (a)-(d), templates are adopted from literatures [3], which are used to delete square border neurons. In Fig.2 (e) - (h), the templates are extended in order to avoid disconnection of images. As shown in Fig.4, one can see that four '0' (object neurons) are deleted when templates are not extended. Obviously, this is not in accordance with the rules of image thinning. Moreover, 'a', 'b' and 'c' cannot equal to '1' simultaneity. The templates in Fig.3 (a)-(d) are used to delete triangle border neurons. From Fig.4, one can see the centre neuron '0' (object pixels) are deleted when using templates proposed in literatures [3].

$$\underset{(a)}{\begin{array}{|c|c|c|}\hline \times & 1 & 1 \\ \hline 0 & 0 & 1 \\ \hline 0 & 0 & \times \\ \hline\end{array}} \quad \underset{(b)}{\begin{array}{|c|c|c|}\hline 0 & 0 & \times \\ \hline 0 & 0 & 1 \\ \hline \times & 1 & 1 \\ \hline\end{array}} \quad \underset{(c)}{\begin{array}{|c|c|c|}\hline \times & 0 & 0 \\ \hline 1 & 0 & 0 \\ \hline 1 & 1 & \times \\ \hline\end{array}} \quad \underset{(d)}{\begin{array}{|c|c|c|}\hline 1 & 1 & \times \\ \hline 1 & 0 & 0 \\ \hline \times & 0 & 0 \\ \hline\end{array}}$$

$$\underset{(e)}{\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline a & b & c \\ \hline\end{array}} \quad \underset{(f)}{\begin{array}{|c|c|c|c|}\hline a & 0 & 0 & 1 \\ \hline b & 0 & 0 & 1 \\ \hline c & 0 & 0 & 1 \\ \hline\end{array}} \quad \underset{(g)}{\begin{array}{|c|c|c|}\hline a & b & c \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline\end{array}} \quad \underset{(h)}{\begin{array}{|c|c|c|c|}\hline 1 & 0 & 0 & a \\ \hline 1 & 0 & 0 & b \\ \hline 1 & 0 & 0 & c \\ \hline\end{array}}$$

Fig.2. Square template

$$\underset{(a)}{\begin{array}{|c|c|c|}\hline \times & 1 & 1 \\ \hline 0 & 0 & 1 \\ \hline \times & 0 & \times \\ \hline\end{array}} \quad \underset{(b)}{\begin{array}{|c|c|c|}\hline 1 & 1 & \times \\ \hline 1 & 0 & 0 \\ \hline \times & 0 & \times \\ \hline\end{array}} \quad \underset{(c)}{\begin{array}{|c|c|c|}\hline \times & 0 & \times \\ \hline 1 & 0 & 0 \\ \hline 1 & 1 & \times \\ \hline\end{array}} \quad \underset{(d)}{\begin{array}{|c|c|c|}\hline \times & 0 & \times \\ \hline 0 & 0 & 1 \\ \hline \times & 1 & 1 \\ \hline\end{array}}$$

$$\underset{(e)}{\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline \times & 0 & \times \\ \hline 0 & 0 & 0 \\ \hline\end{array}} \quad \underset{(f)}{\begin{array}{|c|c|c|}\hline 1 & \times & 0 \\ \hline 1 & 0 & 0 \\ \hline 1 & \times & 0 \\ \hline\end{array}} \quad \underset{(g)}{\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline \times & 0 & \times \\ \hline 1 & 1 & 1 \\ \hline\end{array}} \quad \underset{(h)}{\begin{array}{|c|c|c|}\hline 0 & \times & 1 \\ \hline 0 & 0 & 1 \\ \hline 0 & \times & 1 \\ \hline\end{array}}$$

Fig.3. Triangle template

$$\underset{(a)}{\begin{array}{|c|c|c|c|}\hline 1 & 0 & 0 & 1 \\ \hline 1 & \boxed{0} & \boxed{0} & 1 \\ \hline 1 & \boxed{0} & \boxed{0} & 1 \\ \hline 1 & 0 & 0 & 1 \\ \hline\end{array} \quad \begin{array}{|c|c|c|c|}\hline 1 & 0 & 0 & 1 \\ \hline 1 & \boxed{1} & \boxed{1} & 1 \\ \hline 1 & \boxed{1} & \boxed{1} & 1 \\ \hline 1 & 0 & 0 & 1 \\ \hline\end{array}} \qquad \underset{(b)}{\begin{array}{|c|c|c|}\hline 1 & 0 & 1 \\ \hline 0 & \boxed{0} & 0 \\ \hline 1 & 0 & 1 \\ \hline\end{array} \quad \begin{array}{|c|c|c|}\hline 1 & 0 & 1 \\ \hline 0 & \boxed{1} & 0 \\ \hline 1 & 0 & 1 \\ \hline\end{array}}$$

Fig.4. (a) Destroy connection of image (b) Delete the centre pixel

The proposed binary image thinning algorithms are listed below.

(1) PCNN is initialized. The feeding input $F_{ij}(t)$ equals to the load signal $X_{ij}(t)$, and the threshold $\theta_{ij}(t)$ is initialized using $F_{ij}(t)$. Listing templates M_{kmn} (1≤k≤16) (seeing Fig.3 and Fig.4).Then shutting off the load signal $X_{ij}(t)$ in order to make $F_{ij}(t)$ only determined by $Y_{ij}(t)$.In the first iteration, all background neurons fire and object neurons don't fire. So the neuron output $Y_{ij}(t)$ equal to $X_{ij}(t)$.

(2) If no new neurons with pulses will be generated, thinning process is accomplished and $Y_{ij}(t)$ is the thinning result; otherwise it will shift to the next step.

(3) At t>0, according to each template M_{kmn} (1≤k≤16), $\phi_{mn}(t)$ is calculated using equation (3). Then we can obtain $U_{ij}(t)$ using equation (4). Here $\theta_{ij}(t)$ is determined by each template. The threshold $\theta_{ij}(t)$=6 in Fig.2 (a) - (d) and Fig.3 (e) - (h), and in Fig.2 (e) - (h) and Fig.3 (a) - (d) $\theta_{ij}(t)$=8, and $\theta_{ij}(t)$=5, respectively.

(4) Compare $U_{ij}(t)$ with $\theta_{ij}(t)$.If $U_{ij}(t) \geq \theta_{ij}(t)$ and the template are obtained from Fig.2, the neuron is marked with a flag of deletion. Else if $U_{ij}(t) \geq \theta_{ij}(t)$ and template matrix is obtained from Fig.3, the neuron previous output $Y_{ij}(t)$ is replaced by '1'. Moreover, the neuron feeding input $F_{ij}(t)$ must be obtained feedback from $Y_{ij}(t)$ immediately. Thus, the result is $F_{ij}(t)$=1. This method is called sequential image thinning.

(5) If all neurons are not scanned, go to step (3), else delete the neuron with flag of deletion. This method is called parallel image thinning. In addition, t=t+1, go to step (2).

4 Experimental comparison

The aim of image thinning is to obtain the skeleton of the image. In order to evaluate the thinning algorithm, the thinning rate (TR) (see equation (7)) and thinning time expense (TTE) are used.

$$TR = (1 - T_1 / T_0) * 100\% \qquad (7)$$

Where T_0 is the number of object pixel in original image, and T_1 is the number of object pixels in thinning image. Generally, large TR indicates higher thinning degree, and small TTE means faster thinning speed.

In the experiments, some common binary images and fingerprint images of FVC2004 databases are used to evaluate the thinning algorithms. The proposed thinning algorithm is compared with literature [3], [4] and [5]. Our algorithms are simulated using MATLAB 7.0, and experiments are carried out on a computer with 2.8-GHz Intel Celeron D Processor Unit and 256-MB random access memory.

Fig.5-7 showed some experimental comparison of common binary image thinning. In literature [4] and [5], the skeleton of the image were extracted. However, some new breaks are generated in literature [4], and the thinning images was not signal pixel in literature [5] (see Fig.5-7). Moreover, the thinning algorithm [3] can obtain good image skeleton, and single-pixel requirement is well satisfied. But this algorithm often deletes the centre pixel in the crossing region, and make algorithm complex because of using two steps. The method in this paper preserves the image shape as well as the connectivity, and no breaks were generated. And the method can be used to sequential image thinning and parallel image thinning to make the thinning result same thinner.

In the experiments, all the fingerprint images are preprocessed by the enhancement algorithm [13]. As shown in Fig.8-11, the method in literature [3] is better than the other PCNN thinning methods. Comparing with literature [3] (see Figs.8-11), the method in this paper adopts sixteen templates. Table 1 gives numeric thinning performance comparisons.

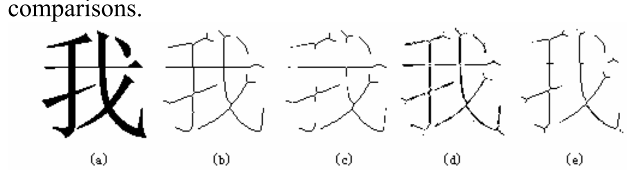


Fig.5. Chinese character thinning.(a) Original image, with(256×256),(b) thinning image of the proposed algorithm,(c) thinning image of literature [3] algorithm,(d) thinning image of literature [4] algorithm,(e) thinning image of literature [5] algorithm.

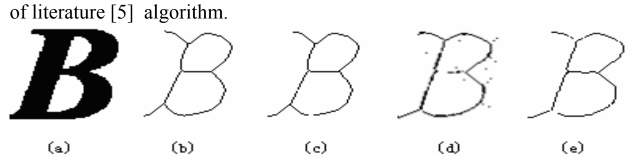


Fig.6. English letter thinning.(a) Original image, with(128×128), (b) thinning image of the proposed algorithm,(c) thinning image of literature [3] algorithm,(d) thinning image of literature [4] algorithm,(e) thinning image of literature [5] algorithm.

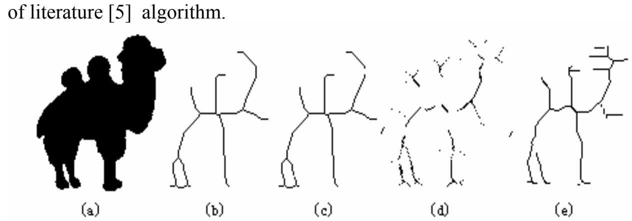


Fig.7. Animal image thinning.(a) Original image, with(140×127), (b) thinning image of the proposed algorithm,(c) thinning image of literature [3] algorithm,(d) thinning image of literature [4] algorithm,(e) thinning image of literature [5] algorithm.

Fig.8. Fingerprint image thinning. (a) Enhanced image from FVC2004 DB1, with a size of 640×480, (b) thinning image of the proposed algorithm, (c) thinning image of literature [3] algorithm.

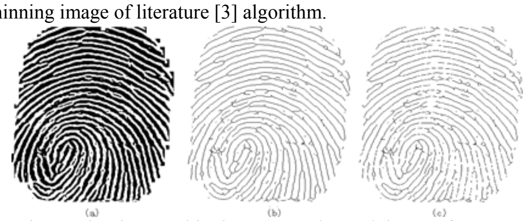


Fig.9. Fingerprint image thinning. (a) Enhanced image from FVC2004 DB2, with a size of 328×364, (b) thinning image of the proposed algorithm, (c) thinning image of literature [3] algorithm.

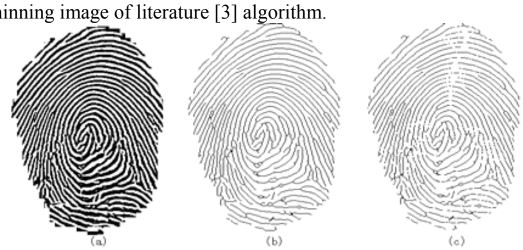


Fig.10. Fingerprint image thinning. (a) Enhanced image from FVC2004 DB3, with a size of 300×480, (b) thinning image of the proposed algorithm, (c) thinning image of literature [3] algorithm.

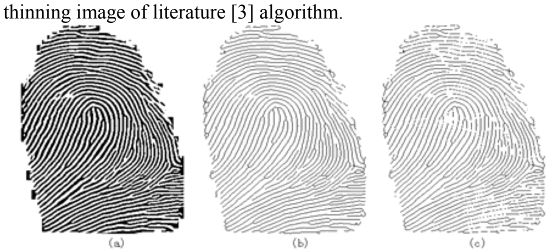


Fig.11. Fingerprint image thinning. (a) Enhanced image from FVC2004 DB4, with a size of 288×384, (b) thinning image of the proposed algorithm, (c) thinning image of literature [3] algorithm.

Table 1 Numeric comparison of thinning performance

Images in	Thinning Time Expense(Second)			Thinning Rate (%)		
	Our method	literature [3]	literature [5]	Our method	literature [3]	literature [5]
Fig.5	77.4176	82.5717	82.2529	0.0781	0.0713	1.0000
Fig.6	92.6165	92.7599	92.3297	0.0156	0.0156	0.2188
Fig.7	95.0486	95.1370	95.1901	0.0781	0.0625	0.2969
Fig.8	82.1978	83.4261	81.6933	0.0781	0.0800	0.6719
Fig.9	81.8693	83.5178	81.5345	0.3438	0.2656	4.3438
Fig.10	83.3302	84.7103	83.5845	0.1250	0.1406	1.2031
Fig.11	75.0269	76.9679	76.4175	0.0938	0.0781	0.7344

In Fig.8-11, it can be seen that the present method preserves connectivity compared with literature [3]. And from Table 1, one can also see that the algorithm in literature [3] obtains high thinning rate and faster thinning speed at the expense of the connectivity of image. Statistically analyzed, the method in this paper can also get high thinning rate and faster thinning speed in the situation of not destroying the connectivity of original image.

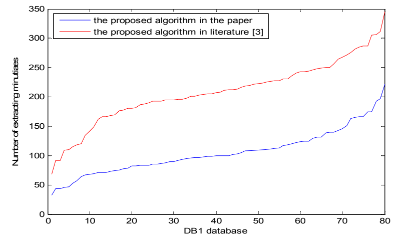


Fig.12. Number of extracting minutia comparison on DB1 of FVC2004

To evaluate the effectiveness and usefulness of thinning algorithm in fingerprint recognition, fingerprint recognition system was employed in [2] for the experiments on database of FVC2004, i.e., DB1.The system was tested by applying two different thinning algorithms: the algorithm in literature [3]. Fig.12 showed that both algorithms can extract the minutiae. However, statistically analyzed, on DB1, the method in this paper can extract truer minutia. The number of extracting minutia in literature [3] is nearly twice as much as in our method. Thus, it is obviously that the algorithm in this paper can work better than the method in literature [3]. To sum up, the proposed algorithm can reduce the spurious minutia and promote the speed of extracting minutia in AFIS.

5 Conclusion

A PCNNs template-based method for binary fingerprint image thinning was proposed. Sequential and parallel conditions are combined for image thinning. Using pulse-coupled templates, the algorithm extracted the skeleton of images iteratively by neuron pulses. It can preserve the basic image shape and the original connectivity without bringing new breaks to common binary images as well as the fingerprint ones. Experiment results showed that the proposed algorithm was fast in terms of thinning speed and could obtain high thinning rate. This algorithm could be applied to real fingerprint identification systems to improve the recognition performances.

References

[1] V. Espinosa-Duro, “Fingerprints thinning algorithm”, *IEEE Aerosp. Electron. Syst. Mag.*, vol. 18, no. 9, pp.28-30, Sept. 2003.

[2] Wahab, A., Chin, S.H.,Tan, E.C., “novel approach to automated fingerprint recognition”, *IEEE Proceedings: Vision, Image and Signal Processing*, vol. 145, no. 3 , pp. 160-166, Jun. 1998.

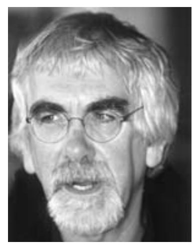
[3] L.P. Ji, Y. Zhang, L.F. Shang, and X.R. Pu, ”Binary Fingerprint Image Thinning Using Template-Based PCNNs”, *IEEE Transactions on Systems, Man, and Cybernetic, Part B: Cybernetics*, vol. 37, no. 5, pp 1407-13,Oct. 2007.

[4] X. Gu, D.Yu, and L. Zhang, ”Image thinning using pulse coupled neural network”, *Pattern Recognit. Lett.*, vol.25, no.9, pp.1075-1084, Jul. 2004.

[5] L. Shang, Z. Yi, and L. Ji, ”Binary image thinning using autowaves generated by PCNN”, Neural Process. Lett., vol.25, no.1, pp.49-62, Feb. 2007.
[6] J. Lin, and Z. Chen, ”A Chinese-character thinning algorithm base on global features and contour information”, *Pattern Recognit.*, vol.28, no.4, pp. 493-512, Apr. 1995.
[7] C. Arcelli and G. Sanniti di Baja, ”On the sequential approach to medial line transformation”, *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-8, no. 2, pp.139-144, Feb. 1978.
[8] L. Lam, and C.Y. Suen, “An evaluation of parallel thinning algorithm for character recognition”, *IEEE Trans. Pattern Anal.Mach.Intell.*, vol.17, no.9, pp. 914-919, Sep. 1995.
[9] T.Y. Zhang, and C.Y. Suen, “A fast parallel algorithm for thinning digital patterns”, *Commun. ACM*, vol.27, no.3, pp. 236-239, Mar. 1984.
[10] F. Zhang, Y.S. Wang, C.Y. Gao, S.C. Si, J.Q. Xu,”An improved parallel thinning algorithm with two subiterations”, Optoelectronics Letters, vol. 4, no. 1, pp. 69-71,January, 2008.
[11] K. Abe, F. Mizutani, and C. Wang. “Thinning of gray-scale images with combined sequential and parallel conditions for pixel removal”, *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 2, pp. 294-299, Feb. 1994.
[12] R.Y. Wu, W.H. Tsai,”A new one-pass parallel thinning algorithm for binary images”, *Pattern Recognition Letters*,vol. 13, no. 10, pp.715-723, October 1992.
[13] A.M. Tahmasebi, S. Kasaei, ”A novel adaptive approach to fingerprint enhancement filter design”, *Signal Processing: Image Communication*, vol 17, no 10, pp. 849-55, Nov. 2002.
[14] Johnson J L, Padgett M L, “PCNN Models and Applications”, *IEEE Trans.Neural Networks*, vol.10, no.3, pp.
480-498, May, 1999.
[15] B. Yu and L.M. Zhang, “Pulse-coupled neural networks for contour and motion matchings”, *IEEE Trans. Neural Netw.*, vol. 15, no.5, pp. 1186-1201, Sep. 2004.
[16] A.Jain, L.Hong, and R. Bolle, ”On-line fngerprint verification”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 302-314, Apr. 1998.
[17] Y.Y.Tang and X.G.You,”Skeletonization of ribbon-like shapes based on a new wavelet function”, *IEEE Trans.Pattern Anal.Mach.Intell.*, vol.25, no.9, pp.1118-1133, Sep. 2003.
[18] B.Yu and L.M. Zhang, ”Pulse-coupled neural networks for contour and motion matchings”, *IEEE Trans.Neural Netw.*, vol.15, no.5, pp. 1186-1201, Sep. 2004.
[19] X.Gu, D.Yu, and L.Zhang, ”Image shadow removal using pulse coupled neural network”, *IEEE Trans. Neural Netw.*, vol. 16, no.3, pp.692-698, May, 2005.

Dacheng Xu is currently working as a full professor the school of Applied Technology at Soochow University, born in Lanzhou, Gansu provinces, in 1963, graduated from University of Electron Science and Technology of China with master degree of engineering. He is an active senior member of Chinese Institute of Electronics (CIE) and a member of the IEEE Computer Society. His research focus is circuit theory, signal and information processing, and micro-computer application.

Bailiang Li received his B.S. degree in Electronic Science and Technology from Soochow University, Suzhou, China in 2006. He is currently working for the M.S. degree in School of Electronic Information at University of Soochow. His research focus is image processing.



Anton Nijholt is heading the Human Media Interaction (HMI) department of the University of Twente, the Netherlands. He studied Mathematics and Computer Science at Delft University of Technology and did his PhD at the Vrije Universiteit Amsterdam. He is active in numerous European research projects and Networks of Excellence. His research interests include artificial intelligence and human-computer interaction.