TR-92-003

STROKE-BASED HANDWRITTEN CHINESE CHARACTER RECOGNITION USING NEURAL NETWORKS

Hong-Yuan Liao, Jun-Shon Huang, and Shih-Ta Huang

Institute of Information Science
Academia Sinica, Nan-Kang
Taipei, Taiwan, ROC

TEL: 886-2-788-3799
FAX: 886-2-782-4814

STROKE-BASED HANDWRITTEN CHINESE CHARACTER RECOGNITION USING NEURAL NETWORKS

Hong-Yuan Liao, Jun-Shon Huang, and Shih-Ta Huang

Institute of Information Science
Academia Sinica, Nan-Kang
Taipei, Taiwan, ROC

TEL: 886-2-788-3799
FAX: 886-2-782-4814

1. INTRODUCTION

Chinese characters represent one kind of delicate arts. Each character has its structural meaning. Basically they are composed of certain number of writing strokes, which can be roughly classified into 33 categories. There are about 5401 daily-used characters, of which many are very similar or too complex that the efficient machine recognition of them by conventional methods seems not feasible. The conventional approaches include extracting statistical features for classification [1-4] and segmenting strokes for structure matching [5-8]. The first approach is suitable for machine-printed characters and constrained handwritten characters, whereas the second approach is suitable for general handwritten characters. In our study here, we try an innovative method, the neural network approach, to recognize the general handwritten Chinese characters.

The neural network approach for Chinese character recognition has been studied by Kimura [9], where a modified back-propagation neural net is used and only four similar characters with various distortions and noises are tested. Yong [10] has also studied handprinted Chinese character recognition by using a content addressable associative memory where the recognition scheme consists of two phases: a stroke recognizing phase in which a handprinted character is encoded, and a character recognizing phase in which the handwritten character is matched with a standard character whose code is the closest to that of handwritten character. However, no experiment is done in this study. Wang *et al.* [11] study multilayer backpropagation neural network for handwritten Chinese character recognition where each character is transformed into polar coordinate; the contour of the transformed character is coded and input to the neural net, and experiments have been conducted on a set of 48 radicals.

In this paper, we extract strokes of Chinese characters by conventional methods [12-14] where each stroke is restricted to be a straight line segment. Then, we derive a set of features from each stroke of a character. For all model characters, these features are fed into columns a two-dimensional Hopfield neural network, and for an unknown character, its features are fed into rows. Finally, through an optimization process executed in the neural net, a quantitative similarity measure between the unknown character and each of the model characters is derived. Compared with the conventional matching schemes, the proposed technique provides a more general and compact formulation of the problem and a solution more suitable for parallel implementation. In the next section, we briefly mention about some preprocessing tasks and stroke extraction procedure. Then, the basic concept of the continuous Hopfield network model for matching is introduced in Section 3. In Section 4, the procedure of mapping the character recognition scheme into neural network is described. This is followed by the presentation of experimental results in Section 5. Finally, a conclusion is made in Section 6.

2. PREPROCESSING AND STROKE EXTRACTION

In this paper, we consider strokes as basic constituents of a Chinese character. Therefore, the first step toward Chinese character recognition is to segment every single character into a set of its constituent strokes. In this section, we will describe the procedure of preprocessing and stroke extraction.

2.1. Preprocessing

Generally, a digitized image with Chinese character may have some unwanted holes and noises. Therefore, the first step of preprocessing is to smooth the raw image. Then, for the sake of reducing subsequent processing time and storage space, a thinning algorithm is used to minimize the information of each character. After the thinning process, the resultant skeletons are more typical to the original patterns and more feasible for further analysis.

In the thinning process, we adopt a stripping method which strips off the character from the outermost layer (each time one layer) until its skeleton is reached. After the thinning process, it is possible to have some spurious branches in the image. Therefore, the resultant skeletons are traced and the set of n-fork points ($n \geq 3$) are located. We then use a threshold to determine whether a branch should be deleted or not. An example showing the result of preprocessing is in Fig.1.

2.2. Stroke Extraction

The purpose of stroke extraction is to extract the set of constituent strokes from a Chinese character. In this process, we use end points in the skeleton (Fig.2) as starting points. An iterative tracing process starting from each end point is performed. Then, we classify each segment as either a curve or a straight line segment. In this paper, we require that all strokes be straight line segments. It is well known that a curve segment can be divided into several straight line segments from its turning points. We apply k-curvature method to locate the set of turning points in a curve segment. Then, a set of criteria is used to determine whether two adjacent line segments should be concatenated into one. Finally, the set of extracted strokes is labeled with a top to bottom, left to right order. An example demonstrating the result after performing the stroke extraction process together with the labeled strokes is shown in Fig.3.

3. TWO-DIMENSIONAL HOPFIELD NETWORKS FOR MATCHING

In this section, a two-dimensional Hopfield neural array is introduced. This neural net will be used as a tool for handwritten Chinese character recognition. A Hopfield net [15] is composed of numerous highly parallel computing units. It is built from a single layer of neurons, with feedback connections from each unit to every other unit (but not to itself). The weights on these connections are constrained to be symmetrical. Generally, we first characterize a problem to be solved by an energy function E.

Through a constraint satisfaction process performed in the Hopfield net, the matching result is ultimately reflected in the states of the neurons.

In general, the rows of the net are arranged to represent the features of an unknown handwritten character, and the columns represent the features of a model character. The state of a neuron reflects the degree of similarity between two nodes, one from the unknown character and the other from the model character. The matching process can be characterized as minimizing the following energy function [17]:

$$E = -\sum_i\sum_k\sum_j\sum_l C_{ikjl} V_{ik} V_{jl} + \sum_i (1 - \sum_k V_{ik})^2 + \sum_k (1 - \sum_i V_{ik})^2 \qquad (1)$$

where V_{ik} is a state variable which converges to 1.0 if the ith node in the unknown character matches the kth node in the model character ; otherwise, it converges to 0.0. The first term in (1) is a compatibility constraint. The second and the third terms are the uniqueness constraints so that the summation of the states of the neurons in each row or column is equal to 1.0. The major component of the compatibility measure C_{ikjl} (or strength of interconnection) between a neuron in row i column k and a neuron in row j column l is expressed in terms of a function F defined as follows:

$$F(x,y) = \begin{cases} 1 & , \text{if } |x-y| < \theta \\ -1 & , \textit{otherwise} \end{cases} \qquad (2)$$

where θ is a threshold value, and x and y are features pertaining to row and column nodes, respectively. C_{ikjl} can be expressed as

$$C_{ikjl} = \sum_n W_n \times F(x_n, y_n) \qquad (3)$$

where x_n is the nth feature of the node in row i column k, y_n is the nth feature of the node in row j column l, and the summation of the weighting functions W_n's is equal to 1. Equation (1) can be simplified and fit into the energy function form of a

Hopfield network [16] as follows:

$$E = -\frac{1}{2}\sum_i\sum_k\sum_j\sum_l T_{ikjl} V_{ik} V_{jl} - \sum_i\sum_k I_{ik} V_{ik} \tag{4}$$

where $I_{ik} = 2$, and

$$T_{ikjl} = C_{ikjl} - \delta_{ij} - \delta_{kl} \tag{5}$$

where $\delta_{ij} = 1$, if $i = j$, and $\delta_{ij} = 0$ otherwise.

Matching can be considered as a constraint satisfaction process [18-20]. According to Hopfield and Tank [15], the strength of the connection between each neuron pair can be derived from the energy function. Based on these connections, the equation of motion for input u_{ik} of a neuron at position (i,k) can be derived as follows:

$$\frac{du_{ik}}{dt} = \sum_j\sum_l C_{ikjl} V_{jl} - \sum_l V_{il} - \sum_j V_{jk} - \frac{u_{ik}}{\tau} + I_{ik} \tag{6}$$

where

$$V_{ik} = g(\, u_{ik}\,) = \frac{1}{1 + e^{\frac{-2u_{ik}}{u_0}}} \tag{7}$$

Since the sum of V_{ik} on all the neurons at initialization is constrained to be equal to the number of the final desired output, i.e.,

$$\sum_i\sum_k V_{ik} = N \tag{8}$$

where N is either the number of rows or the number of columns of the array depending on which one is smaller. We can derive the initial condition for u_{ik} from equations (7) and (8) as follows:

$$u_{init} = -\frac{u_0}{2}\ln(N - 1) \tag{9}$$

In order to prevent the system from being trapped in a local minimum, a certain amount of noise must be added to this initial value. We can rewrite the initial condition as follows:

$$u_{ik}{}^{0} = u_{init} + \delta \tag{10}$$

and

$$V_{ik}{}^{0} = g(\, u_{ik}{}^{0}\,) \tag{11}$$

where δ is a random number uniformly distributed between $-0.1u_{init}$ and $+0.1u_{init}$.

The algorithm for matching is summerized as follows:

Input: A set of neurons arranged in a two-dimensional array with initial values V_{ik}^0, where $0 \le i \le row\_max - 1$, $0 \le k \le column\_max - 1$, and *row_max* and *column_max* are the numbers of rows and columns in the array, respectively.

Output: A set of stabilized neurons with state values V_{ik}, where $0.0 \le V_{ik} \le 1.0$ for $0 \le i \le row\_max$ and $0 \le k \le column\_max$.

Method:

(1) Set the initial conditions using equations (10) and (11).
(2) Set index = 1 and limit = n.
(3) Randomly pick up a node (i,k).
(4) Update the value of u_{ik}. In order to simulate equation (6), the 4*th* order Runge-Kutta method is used. That is,

$$u_{ik}^{t+1} = u_{ik}^{t} + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

where

$$k_1 = hf(u_{ik}^t),$$

$$k_2 = hf(u_{ik}^t + \frac{1}{2}k_1),$$

$$k_3 = hf(u_{ik}^t + \frac{1}{2}k_2),$$

$$k_4 = hf(u_{ik}^t + k_3),$$

$f(\,.\,)$ is the right hand side of equation (6), and h is a constant.
(5) Calculate the new state of neuron (i,k) as follows:

$$V_{ik} = g(u_{ik})$$

(6) Increment index by 1.
(7) If index $< n$, then go to step (3), else stop and output the final states of all the neurons based on the following rule:

$$V_{ik} = \begin{cases} 1.0 & , \text{ if } V_{ik} > \theta 1 \\ 0.0 & , \textit{ otherwise} \end{cases}$$

where $\theta 1$ is a threshold value.

In the real implementation, the coefficient τ is set to 1 and u_0 is 0.002. The constant h for the Runge-Kutta approximation is 0.0001. The instability problem of the Hopfield neural network has been extensively studied [24-26]. It is well known that the optimal solution is not always found. The time-out strategy adopted in the algorithm forces the system to converge to a stable state and thus implicitly solve the instability problem.

4. CHARACTER RECOGNITION VIA HOPFIELD NET

In this section, the aforementioned two-dimensional Hopfield net is used to perform character matching. The first step of the procedure is to select the set of features. This is followed by the introduction of row-column assignment. Then, the strength of interconnection C_{ikjl} is defined. Finally, a method for quantitatively evaluating the degree of match between the unknown character and the model character is presented.

4.1. Feature Selection and Row-Column Assignments

Before the character matching can be performed, each stroke of the character has to be labeled. This is equivalent to assign each stroke with a specific number. Fortunately, this work has been done in the stroke extraction process (see Fig.3). In fact, during the stroke extraction process, we also calculate the length of each stroke and derive the coordinate of its centroid. For each stroke, we extract two features for character matching. One is a local feature which is the length of the stroke. The other is a relational feature which is defined as a set of distances originated from the stroke's centroid to all the centroids on other strokes of the character. An example demonstrating both features of a stroke is shown in Fig.4. The proposed features are able to implicitly solve both translational and rotational invariance problems. However, they are not helpful to scaling problem. In this paper, we assume the size of the unknown character is similar to that of the character models. In fact, a simple normalization process by conventional methods will resolve the aforementioned problem easily.

In order to perform matching, each stroke of the unknown character is assigned a row index and each stroke of the model character is assigned a column index. An example illustrating this arrangement is shown in Fig.5.

4.2. C_{ikjl} for Character Matching

To perform character matching, C_{ikjl} is expressed as follows:

$$C_{ikjl} = W_1 \times F(I_i, M_k) + W_2 \times F(I_j, M_l) + W_3 \times F(d_{I_i I_j}, d_{M_k M_l}) \tag{12}$$

where I_x represents the length of the xth stroke in the unknown character, M_y the length of the yth stroke in the character model, $d_{I_i I_j}$ the distance between the centroids of the ith and the jth strokes in the unknown character, and $d_{M_k M_l}$ the distance between the centroids of the kth and the lth strokes in the model character.

For having better performance, we put a constraint in the relational feature comparison process. The reason why this constraint is used is to save the amount of comparisons. According to the definition, the relational feature is the distance between the centroids of two arbitrary strokes in a character. In other words, if there are n strokes in a character, then the number of relational distances is C_2^n. In the matching process, an unknown character (with n strokes) and a model character (with m strokes) will require $C_2^n \times C_2^m$ comparisons. In order to reduce the number of comparisons, we identify each stroke with either LONG, MEDIUM, or SHORT, based on its length. In the relational distance comparison process, if the types of the two strokes in the unknown character are different from those in the model character, then we simply neglect it without any comparison. An example showing this case is in Fig.6.

4.3. Similarity Measure for Character Matching

After the states of the Hopfield net are stabilized, we can count the number of active neurons in the network and use it to measure the degree of match (or similarity) between the character model and the unknown character. Since the lengths of the set of strokes in a character are of different sizes, an importance measure I is defined based on each stroke's length. The importance measure of stroke j is

$$I_j = \frac{l_j}{\sum_{i=1}^{n} l_i} \tag{13}$$

where l_j is the length of stroke j and n the total number of strokes in a character. The procedure for similarity measure consists of four steps:

Step 1: Initialize *row_match*, *column_match*, *row_zero* and *column_zero* to be 0.

Step 2: Count the number of 1's in each row. If there is no 1 in row i, increment *row_zero* by 1, skip to the next row and leave *row_match* unchanged. If there is only one 1 in row i, add I_i to *row_match*. If there are n 1s ($n > 1$) in row i, then add $\frac{I_i}{n}$ to *row_match*. Repeat this for all the rows.

Step 3: Do the same calculation for all the columns and update *column_match* and *column_zero*.

Step 4: Update *row_match* and *column_match* by the following equations:

$$row\_match = row\_match \times \frac{number\ of\ rows - row\ zero}{number\ of\ rows}$$

$$column\_match = column\_match \times \frac{number\ of\ columns - column\ zero}{number\ of\ columns}$$

Step 5: Pick up the smaller one from *row_match* and *column_match*, and take the result as the similarity measure.

Given an unknown character and a large number of model characters, the degree of match can be derived by comparing the unknown character and each of the model characters in the final state of the Hopfield net. Ideally, there should be at most one active neuron in each row or column. However, due to the nature of the constraint satisfaction process, it is possible to have more than one candidate in the same row or column in the final state of the network. This situation should be considered as unfavorable and hence decreases the degree of match. This is the reason why $\frac{I_i}{n}$ is added to the degree of match (row or column) instead of I_i when there are n 1s simultaneouly existing in the same row or column. When there is no 1 in a row (or column), it means a stroke of the unknown character (or model character) does not have a corresponding stroke in the model character (or unknown character). Under the circumstances, we define a penalty function to adjust the similarity measure. The definition of the penalty function is described in step 4 of the similarity measure process. It is obvious that when the number of empty rows (or columns) is larger, the penalty is heavier. For a model-based Chinese character recognition system, a set of model characters are stored in the model database. To derive their degrees of match with the unknown character in the Hopfield net, we associate the unknown character with row indexes and model characters column indexes. This arrangement allows us to compare all the model characters simultaneouly with the unknown character provided that the dimension of the neuron array is large enough.

5. EXPERIMENTAL RESULTS

A series of experiments have been conducted to corroborate the proposed method. In order to demonstrate the powerfulness of the proposed scheme, we first select a set of similar Chinese characters as samples (Fig.7). Although these characters are structurally similar, they are quite different semantically. Fig.7(d) shows the correct counterpart of the given unknown character in the character database. It reflects that the model character in Fig.7(d) has the highest matching rate with the input character. The other model characters, though also have high matching rates with the unknown character, are automatically discarded due to the relatively low matching rates in comparison with the correct model. In Fig.8(a), we use a set of model characters containing different number of strokes (7, 8, or 9 strokes) to test the effectiveness of the inexact matching performed in the Hopfield net. Fig.8(b) is an example showing the comparison result between two characters having different number of strokes (7 and 9 strokes, respectively). It is obvious that the proposed scheme is able to deal with inexact matching problem. Fig.9(a) shows a synthetic input character which is a model character undergoes translational and rotational displacements with respect to the image center. For simplicity, we use the same set of model characters as is shown in

Fig.8(a). Figs.9(b)-(f) is a set of model characters selected from the sample set in Fig.8(a). The value below each model is the matching rate between the corresponding model and the unknown character. It is obvious that the correct model character can still be identified no matter how the unknown character is rotated or translated. Fig.10 shows the final states of the stabilized Hopfield net which reflects the comparison result between the unknown character and the best matched model character in Fig.9.

Another important feature of the proposed scheme is its distortion-tolerant capability. Fig.11(a) shows a model character. Fig.11(b) shows the same character written by twenty different individuals. The word below each handwritten character reflects its matching status with the model character shown in Fig.11(a). Although the handwriting style of each individual has great variation, the proposed method can still recognize 90 percents of the set of handwritten characters shown in Fig.11(b). The experimental results reflect that although the distortion will cause some influence on the matching rate but the proposed method is still robust.

6. CONCLUDING REMARKS

In this paper, we proposed a Hopfield neural network approach to solve the handwritten Chinese character matching problem. The proposed Hopfield net for character matching has a flexible structure and is able to solve the problem even if the number of strokes in the unknown character and the model character are different. In other words, the two-dimensional neuron array may have different number of rows and columns and it allows inexact matching [27] to be performed in this net. Furthermore, based on the states of the neurons in the network, a similarity measure between any model character and the unknown character can be derived even if they contain different number of strokes. Besides, the proposed mechanism is able to implicitly resolve the problems caused by translational and rotational displacements. These problems are normally hard to deal with in traditional optical character recognition process.

Another advantage of the proposed scheme is its relatively less heuristics in comparison with other methods. Furthermore, the proposed Hopfield net can deal with matching problems in an elegant manner and is able to quantitatively reflects the degree of similarity in the final states of the neurons.

In this research, the basic primitive used to perform matching is stroke. In fact, there is no ample information could be used in a stroke due to its simple structure. In the future research, we will extend the definition of primitive to a more structural form, i.e., to use radical instead of stroke. Since a radical is composed of several strokes, it has much more information than a stroke does. With more available features, the matching could be even better and reliable.

REFERENCES

[1] R. Casey and G. Nagy, "Recognition of Printed Chinese Characters," *IEEE Trans. on Elect. Comput.*, Vol. 15, pp.91-101, 1966.

[2] Q.-R. Wang, C.-Y. Suen and Y.-Y. Tang, "Application of A Statistical Equivalent Block Classifier in the Recognition of Chinese Characters Printed in Different Fonts," *Int. Conf. on Chinese Computing*, San Francisco, Calif., pp.H2.1-2.13, 1985.

[3] G.-H. Chang, B.-S. Jeng, K.-H. Shyu, T.-S. Liu and J.-K. Lin, "A PC-based Optical Chinese Multifont Recognition System," TL Technical Journal, pp.59-75 (in Chinese), 1990.

[4] J. Tsukumo and K. Asai, "Machine Printed Chinese and Japanese Character Recognition Method and Experiments for Reading Japanese Pocket Books," *IEEE Proc. of Computer Vision and Pattern Recognition*, pp.162-167, 1986.

[5] C.-H. Leung, Y.-S. Cheung, and Y.-L. Wong, "A Knowledge-based Stroke Matching Method for Chinese Character Recognition," *IEEE Trans. on Systems, Man, Cybernetics,* Vol. SMC-17, No.6, Nov./Dec. 1987.

[6] M.-C. Kuo and W.-H. Hsu, "Recognition on Handprinted Chinese Characters by Stroke Relaxation Algorithm," Master Thesis, Department of Electrical Engineering, National Tsing Hua University, May 1989.

[7] C.-W. Liao and J.-S. Huang, "A Transformation Invariant Matching Algorithm for Handwritten Chinese Character Recognition," *Pattern Recognition,* Vol. 23, No. 11, pp.1167-1188, 1990.

[8] S.-L. Chou and W.-H. Tsai, "Recognizing Handwritten Chinese Characters by Stroke Matching," *Proc. 1991 First National Workshop on Character Recognition,* Taiwan, ROC, pp.43-54, 1991.

[9] Y. Kimura, "Distorted Handwritten Kanji Character Pattern Recognition by a Learning Algorithm Minimizing Output Variation," in *Proc. International Joint Conf. on Neural Networks,* July 8-12, Seattle, Washington, pp.I103-I106, 1991.

[10] Y. Yong, "Handprinted Chinese Character Recognition via Neural Networks," *Pattern Recognition Letters,* 7, pp.19-25, 1988.

[11] J.-F. Wang, H.-D. Chang and J.-H. Tseng, "Handwritten Chinese Radical Recognition via Neural Networks," *Proc. 1991 Int. Conf. Computer Process. Chinese and Oriental Languages,* pp.92-97, 1991.

[12] F.-F. Cheng, W.-H. Hsu and M.-Y. Chen, "Recognition os Handwritten Chinese Characters by Modified Hough Transform Techniques," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol.PAMI-11, pp.429-439, 1989.

[13] A. Shio and K. Komori, "A Stroke Extraction Method for Handprinted Chinese Character Recognition," *Trans. IECE Japan,* IE80-14, pp.83-90, 1980.

[14] C.-W. Liao and J.-S. Huang, ''Stroke Segmentation by Bernstein-Bezier Curve Fitting,'' *Pattern Recognition,* Vol. 23, No. 5, pp.475-484, 1990.

[15] J. J. Hopfield, ''Neural Networks and Physical Systems with Emergent Collective Computational Abilities,'' *Proc. Natl. Acad. Sci. USA,* Vol. 79, pp.2554-2558, April 1982.

[16] J. J. Hopfield, ''Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons,'' *Proc. Natl. Acad. Sci. USA,* Vol. 81, pp.3088-3092, May 1984.

[17] W. Li and M. Nasrabadi, ''Object Recognition Based on Graph Matching Implemented by a Hopfield-Style Neural Network,'' in *Proc. Int. J. Conf. on Neural Networks,* II, pp.287-290, Washington D. C., June 18-22, 1989.

[18] B. Parvin and G. Medioni, ''A Constraint Satisfaction Network for Matching 3D Objects,'' in *Proc. Int. J. Conf. on Neural Networks,* II, pp.281-286, Washington D. C., June 18-22, 1989.

[19] E. Mjolsness, G. Gindi, and P. Anandan, ''Optimization in Model Matching and Perceptual Organization,'' *Neural Computation,* 1, pp.218-229, 1989.

[20] R. Mohan, ''Application of Neural Constraint Satisfaction Networks to Vision,'' in *Proc. Int. J. Conf. on Neural Networks,* II, pp.619-620, Washington D. C., June 18-22, 1989.

[21] R. P. Lippmann, ''An Introduction to Computing with Neural Nets,'' *IEEE ASSP Magazine,* pp.4-22, April 1987.

[22] W.-C. Lin, F.-Y. Liao, C.-K. Tsao, and T. Lingutla, ''A Hierarchical Multiple-view Approach to Three-dimensional Object Recognition,'' *IEEE Trans. on Neural Networks,* Vol. 2, No. 1, pp.84-92, 1991.

[23] C.-C. Liang, F.-Y. Liao and W.-C. Lin, ''ART-1 Neural Network for 3-D Object Recognition Using Multiple-Views,'' in *Proceedings of International Joint Conf.*

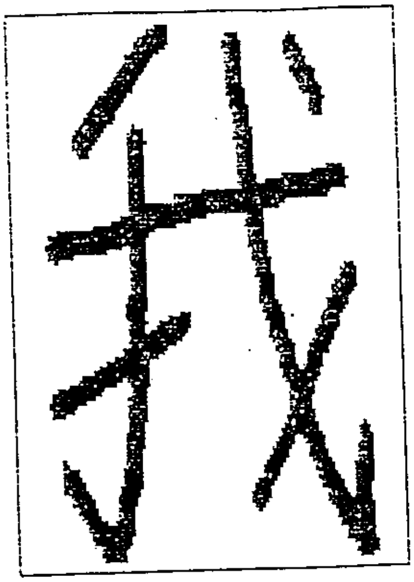
on Neural Networks, San Diego, June 17-21, II, pp.931-936, 1990.

[24] G. V. Wilson and G. S. Pawley, "On the Stability of the Travelling Salesman Problem Algorithm of Hopfield and Tank," *Biological Cybernetics,* 58, pp.63-70, 1988.

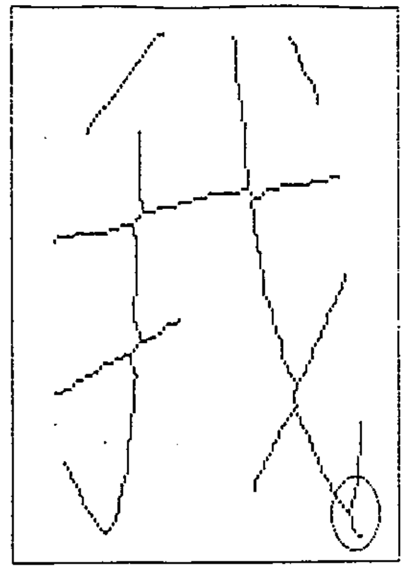
[25] G. W. Davis, "Sensitivity Analysis in Neural Net Solutions," *IEEE Trans. on Systems, Man, and Cybernetics,* Vol.19, No.5, pp.1078-1082, Sept./Oct. 1989.

[26] P. Floreen and P. Orponen, "Counting Stable States and Sizes of Attraction Domains in Hopfield Nets Is Hard," in *Proc. Int. J. Conf. on Neural Networks,* I, pp.395-399, Washington D. C., June 18-22, 1989.

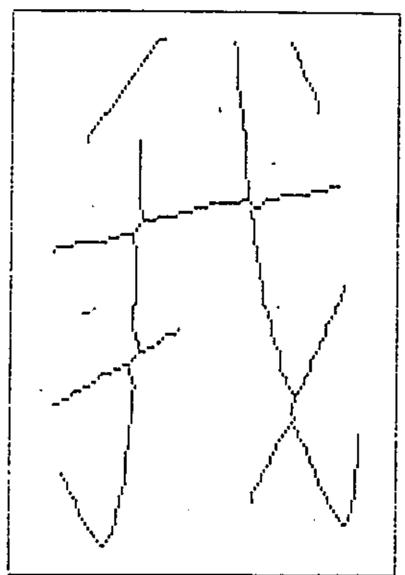
[27] L. G. Shapiro and R. M. Haralick, "Structural Description and Inexact Matching," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol.PAMI-3, pp.504-515, 1981.



(a) original input image



(b) image after thinning process



(c) result after spurious branches has been removed.

Fig.1 Result after preprocessing

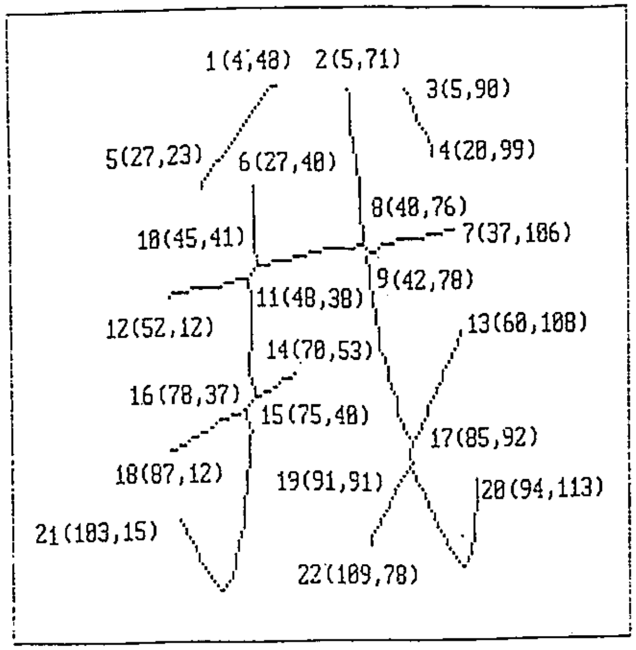


Fig.2 The set of end points and their coordinates in a Chinese character.

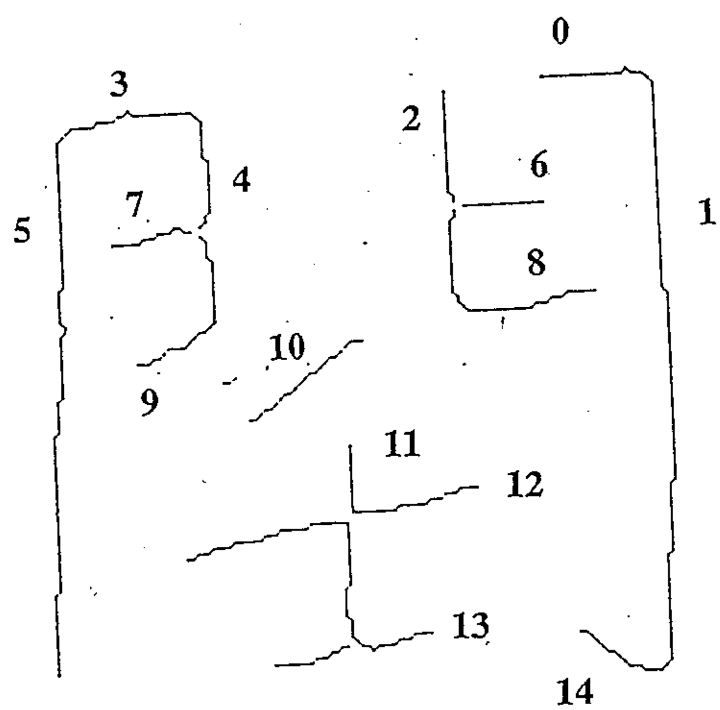
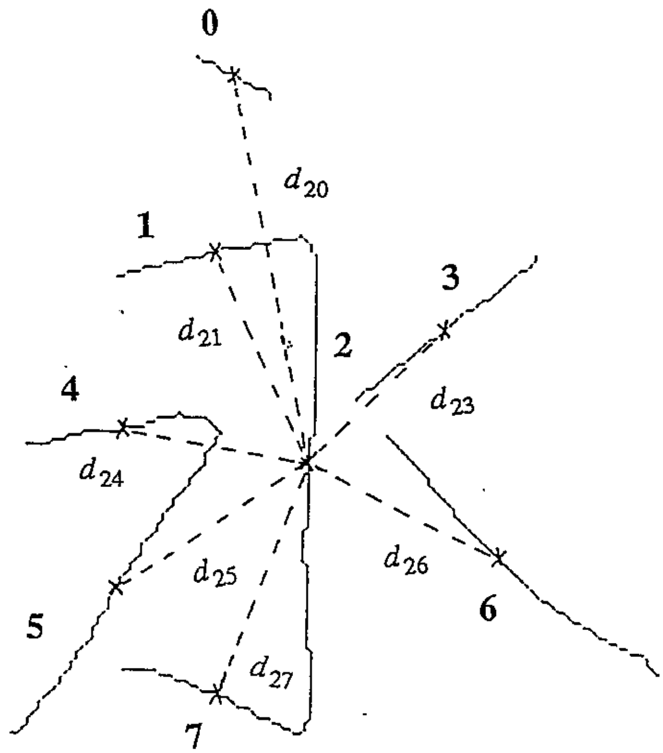


Fig.3 The set of labeled strokes extracted from a Chinese character after stroke extraction process is performed.



local feature: length of stroke 2

relational feature: $\{d_{20}, d_{21}, d_{23}, d_{24}, d_{25}, d_{26}, d_{27}\}$

Fig.4 The feature set of stroke 2 in a Chinese character.

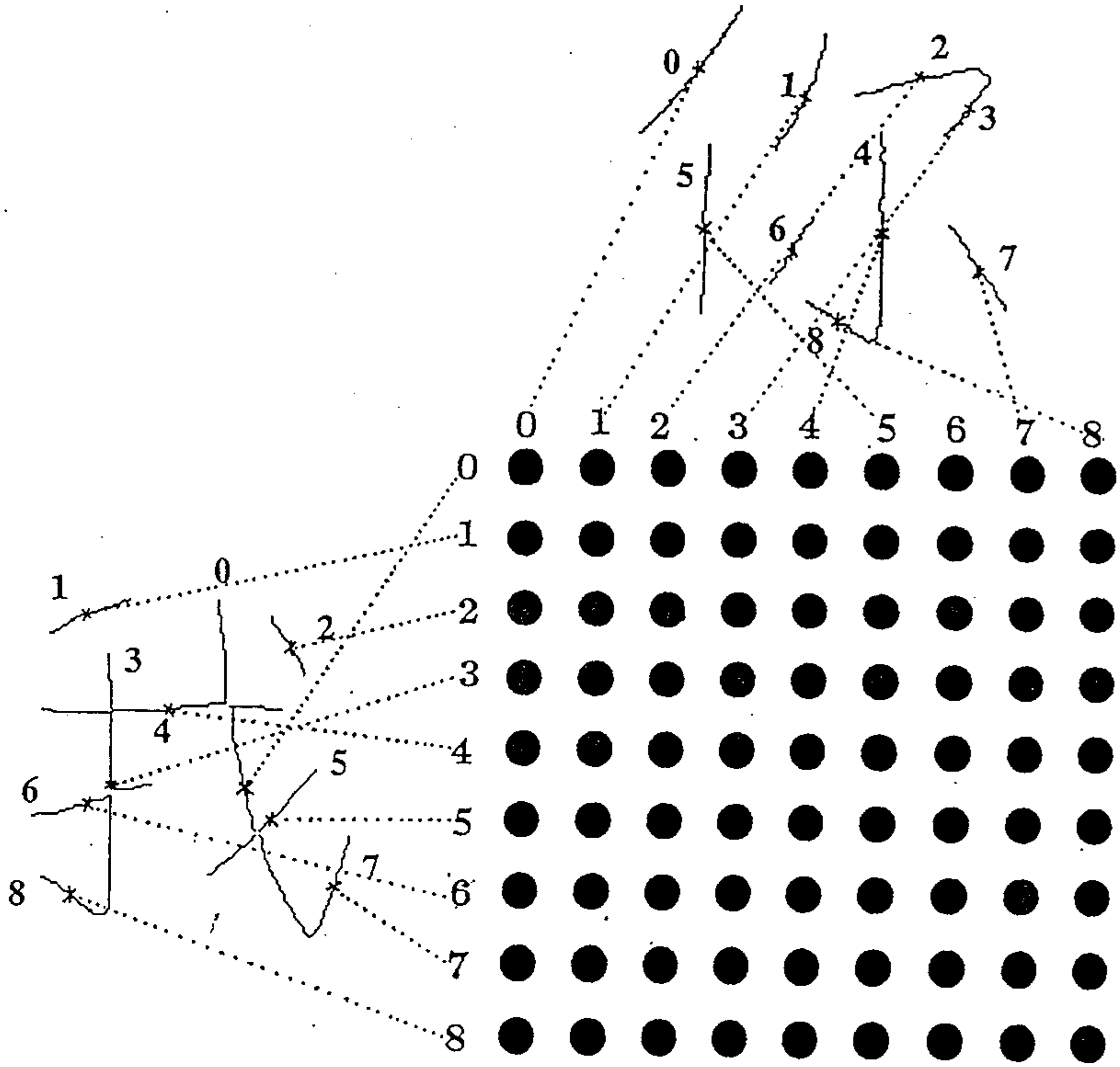


Fig.5 Row-column assignment for Chinese character matching

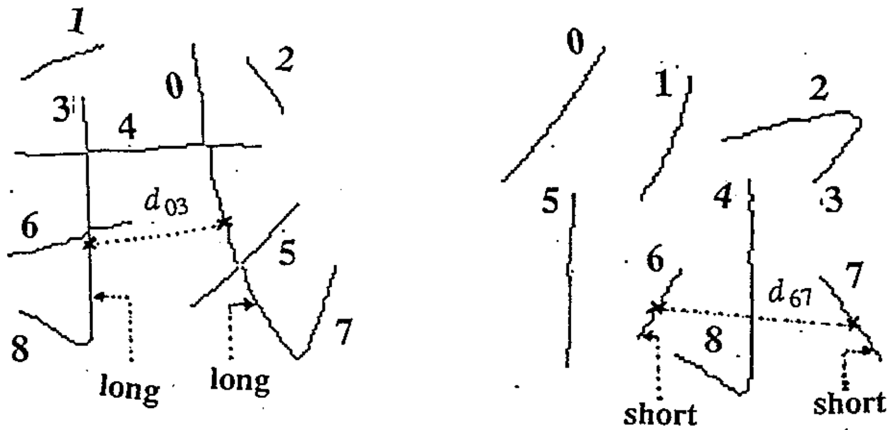
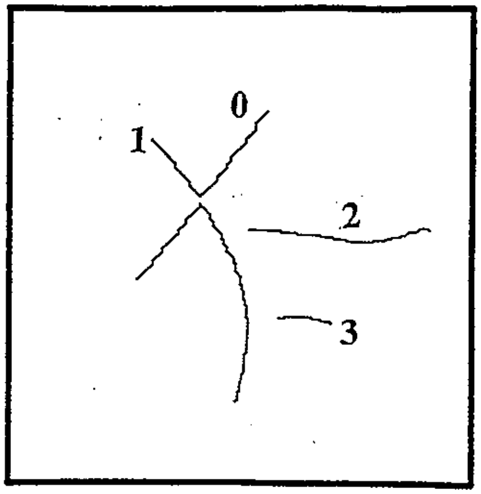
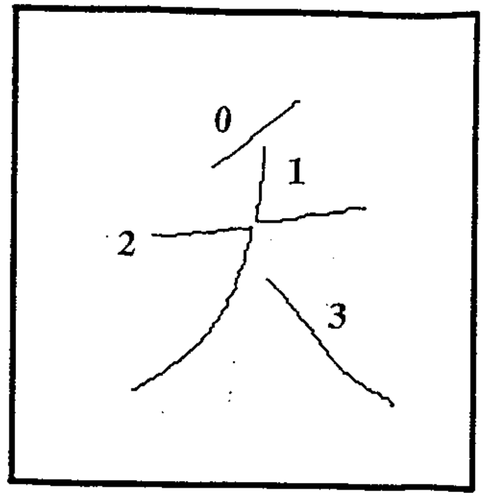


Fig.6 The comparison between d_{03} in unknown character and d_{67} in model character can be omitted due to the type differences of the strokes they connect.



(a)

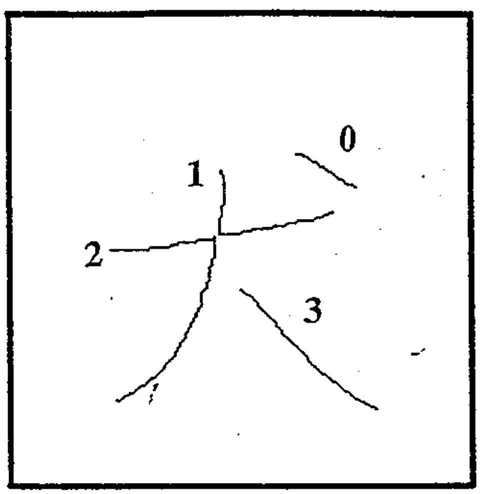


	CL#0	CL#1	CL#2	CL#3
RW#0:	0.00	0.00	1.00	0.00
RW#1:	0.00	1.00	0.00	0.00
RW#2:	0.00	0.00	0.00	1.00
RW#3:	0.00	0.00	0.00	0.00

****

matching rate = 0.690000

(b)

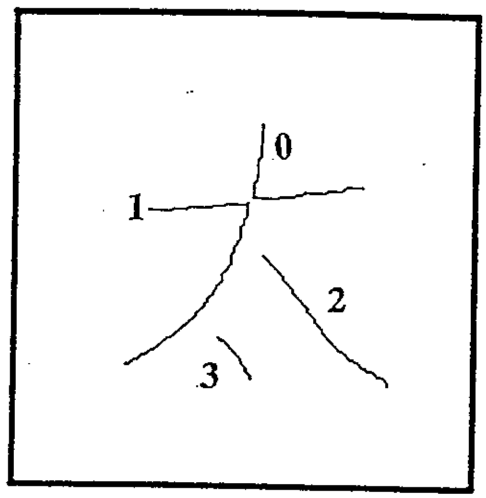


	CL#0	CL#1	CL#2	CL#3
RW#0:	0.00	0.00	1.00	0.00
RW#1:	0.00	1.00	0.00	0.00
RW#2:	0.00	0.00	0.00	1.00
RW#3:	0.00	0.00	0.00	0.00

****

matching rate = 0.690000

(c)



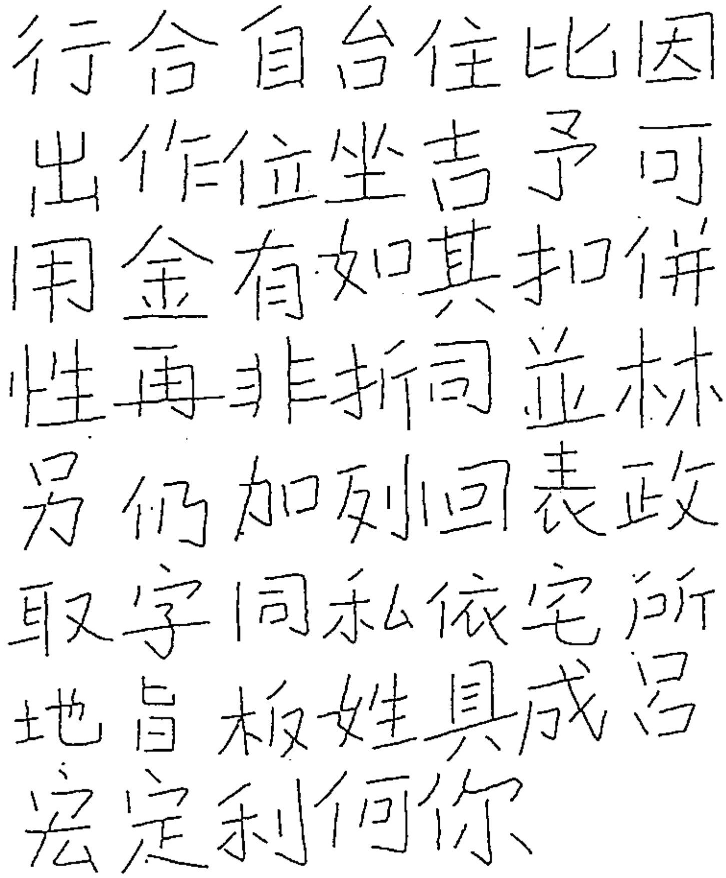
	CL#0	CL#1	CL#2	CL#3
RW#0:	0.00	1.00	0.00	0.00
RW#1:	1.00	0.00	0.00	0.00
RW#2:	0.00	0.00	1.00	0.00
RW#3:	0.00	0.00	0.00	1.00

****

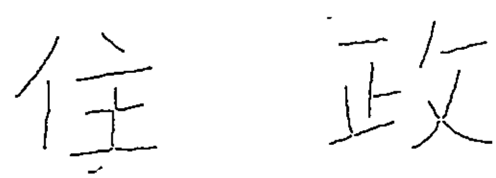
matching rate = 1.000000

(d)

Fig.7 (a) A synthetic input character. (b)-(d) is a set of structurally similar Chinese characters. The proposed method is able to identify (d) as the best matched model character.



(a)



```
       CL#0 CL#1 CL#2 CL#3 CL#4 CL#5 CL#6 CL#7 CL#8
RW#0:  0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00
RW#1:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
RW#2:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
RW#3:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
RW#4:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00
RW#5:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
RW#6:  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
************************
matching rate = 0.043554
```

(b)

Fig.8 (a) A set of model characters containing 7, 8, or 9 strokes. (b) The comparison result of two characters that have different number of strokes.

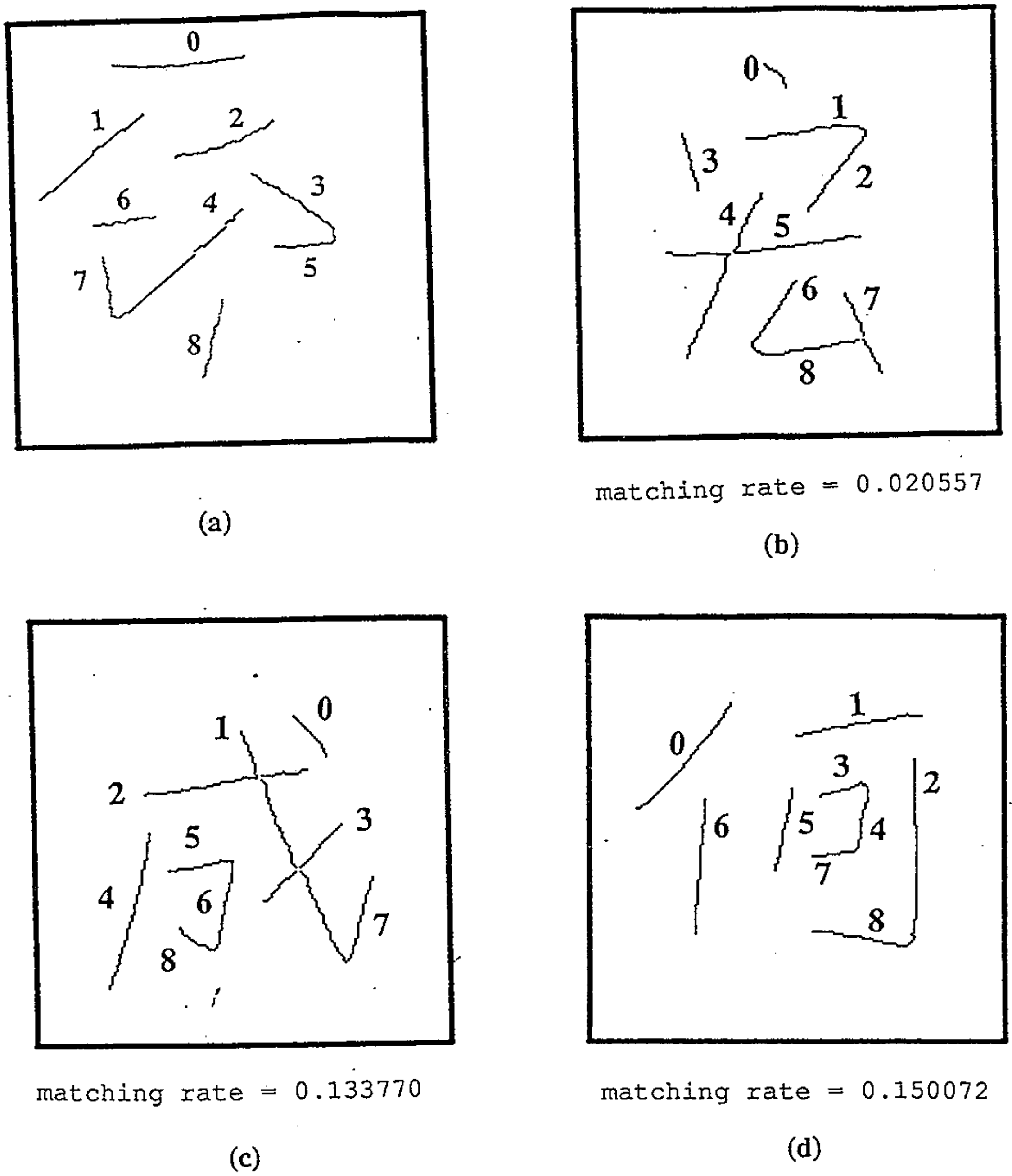
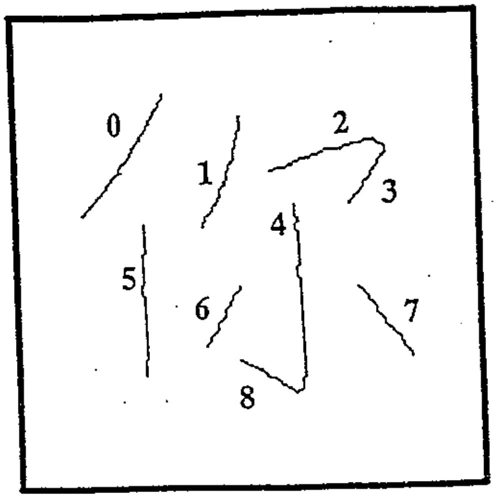
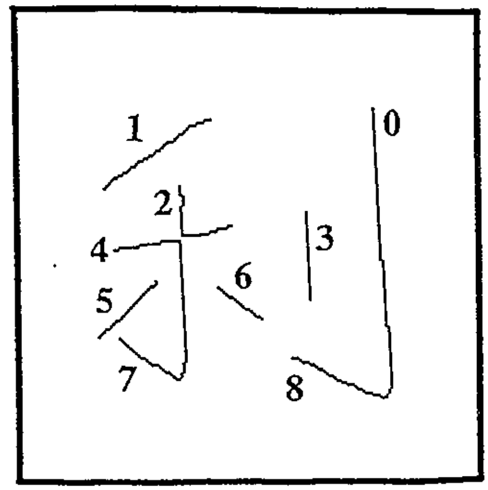


Fig.9 (a) A synthetic input character. (b)-(f) A set of model characters retrieved from database. The numerical value below each model reflects its similarity rate with the unknown character.



matching rate = 1.000000

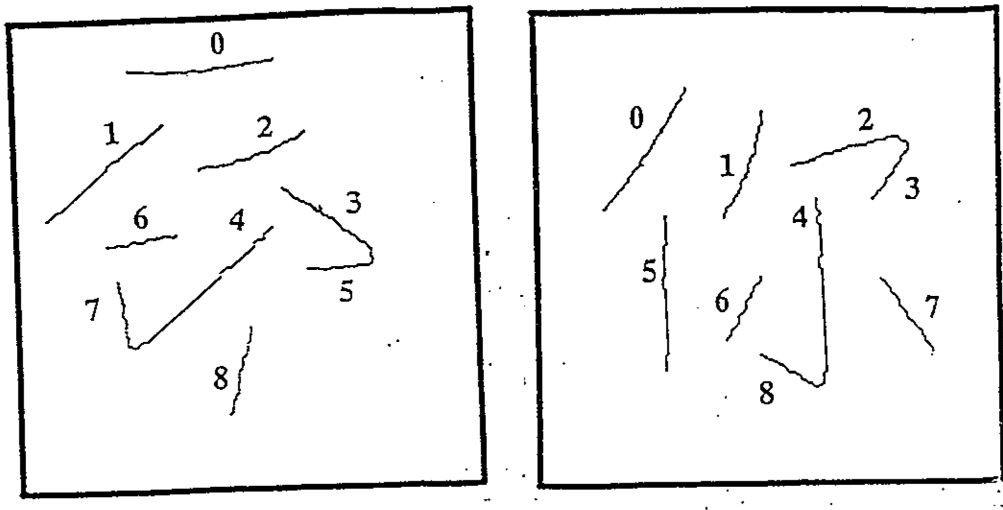
(e)



matching rate = 0.074003

(f)

Fig.9 Continued.



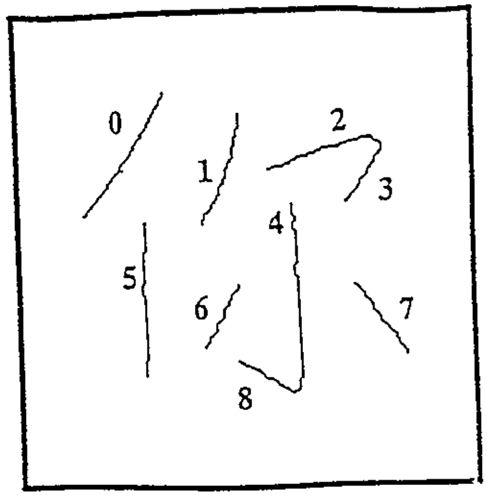
(a) (b)

	CL#0	CL#1	CL#2	CL#3	CL#4	CL#5	CL#6	CL#7	CL#8
RW#0:	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RW#1:	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
RW#2:	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RW#3:	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
RW#4:	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
RW#5:	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
RW#6:	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
RW#7:	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
RW#8:	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00

```
************************
matching rate = 1.000000
```

(c)

Fig.10 (a) A synthetic input character. (b) The model character that has the highest matching rate with the input character. (c) The final neuron matrix that reflects the stroke correspondences between (a) and (b).



(a)

(b)

Fig.11 (a) A model character. (b) A set of handwritten characters (the same character as (a)) written by twenty different individuals. The word below each character represents whether this character is matched with the model character in (a).