Self-Organization of High-order Receptive Fields in Recognition of Handprinted Characters

Cheng-Yuan Liou[1], Hsin-Chang Yang[2]

[1]Dept. of Comp. Sci. Info. Eng., National Taiwan University, Taipei, Taiwan, email: cyliou@csie.ntu.edu.tw
[2]Dept. of Info. Management, Chang-Jung University, Tainan, Taiwan, email: hcyang@mail.cju.edu.tw
This work was supported by National Science Council under project number NSC 84-2213-E-002-012.

Abstract

The printed areas of a handprinted character with thick strokes were replaced by a frame formed by bended ellipses to represent the character efficiently and emulate high order receptive fields in a visual system. To afford topology preservation during adaptive matching of this frame with a template frame, we employ a devised self-organization model. This model uses these bended ellipses as training patterns in searching, measuring, and updating their corresponding ellipses in the template frame. The neighborhood of a corresponding ellipse is also weighted by the appearance of the training bended-ellipse. With this method, each handprinted character can effectively evolve into its template character with predetermined training parameters. Each template has a different number of training cycles. Within this controlled number of cycles, the model can flex a handprinted character into a correct template.

1 Introduction

The recognition process in the brain could be briefly described as follows. Visual neurons which extract features are distributed in different layers in a hierarchical manner. The neurons in lower layers respond to simple features and those in higher layers respond to more complex features. The detected features of an image are then classified according to certain high cortical functions. A similar process is devised in this work. We develop a kind of high-order receptive field to extract various features from a pattern. The features are then classified in a topology-preserving manner by a self-organizing network [1]. This network emulates the high cortical function. This method is inspired by biological evidence and is able to solve the cursive handprinted character recognition problem.

The classification process is directed by the self-organizing network which can preserve the topology of a pattern in a neuron support with reduced dimensions. The training parameters in the network can be predetermined and controlled such that patterns which have the same topology can quickly evolve toward their object template within a fixed number of training cycles. In other words, a template on the support is stable for patterns with the same topology. It may not be stable for other patterns within the given number of training cycles. A new feature representation has also been devised for characters with thick strokes to aid correct convergence on this support. We will explore the above idea and describe a pattern recognition process for handprinted characters with thick strokes.

Most handprinted character recognition systems extract features. There are insufficiencies in the feature extraction methods. These features are extracted from the skeleton which is obtained after applying a thinning process to the raw pattern [2, 3]. As a result, the skeleton is oversimplified compared to the original pattern. This simplification often causes distortion and confusion. So far, no work has provided a foundation for applying such a thinning process. Another reason for not using a thinning process is that the human eye perceives edges rather than skeletons in many perceptual aspects [4]. These facts suggest that we should not ignore the information found within the whole printed area.

We propose a higher-order feature representation which contains sufficient structure information. The new features of a standard pattern can be represented by a set of 5-dimensional feature vectors, $\mathbf{S}^i = \{\mathbf{s}_n^i | 1 \leq n \leq N^i\}$, where N^i is the total number of feature points for the ith character. $\mathbf{s}_n^i$ is the 5-dimensional vector for the nth feature point in this feature set. Basically, the 5 elements in each vector represent geometrically a bended ellipse which is maximally fitted into a local printed area of a pattern. This bended ellipse is used to represent the receptive field of a visual neuron. Two methods have been devised to generate these bended ellipses.

The widely used classification methods fall into the various categories of correlation matching [2]. Correlation matching measures the distance between a handprinted pattern and every standard pattern. The distance is defined on the fitness of alignment across the two sets of the feature points for the two patterns. Alignment is obtained by applying an elastic matching across these two feature sets. Many one-dimensional elastic matching methods have been devised to accomplish this alignment [5, 6, 7]. Recently, several methods have been devised to match two-dimensional images and include such as the self-organizing map [8], backpropagation network [9], elastic beads [10] etc. So far, few methods include topological relations in the matching. In this work, a devised self-organization network is used to perform elastic matching.

2 Bended-Ellipse Receptive Fields

We will present two methods to generate bended-ellipse receptive fields. All the characters used in this work must be normalized in advance. This normalization task is accomplished by shifting and scaling the characters. Let the features of an unknown handprinted pattern be represented by a set of feature vectors $\mathbf{H} = \{\mathbf{h}_m | 1 \leq m \leq M\}$, where M is the total number of feature vectors of the handprinted pattern. The feature vector $\mathbf{h}_m$ can be considered as a receptive field filling a local printed area of the handprinted pattern. Each field represents the structure information of that area. In the following two subsections, we will describe the methods.

0-7803-5871-6/99/$10.00 © 1999 IEEE

2.1 Method 1

We sample seeds p_n^i, $p_n^i = (x_n^i, y_n^i)$ for $1 \leq n \leq N^i$ from the pixels on the skeleton of the ith standard character pattern. p_n^i is usually obtained by uniformly sampling the skeleton. We define a line section, A_n^i, for each seed p_n^i, where A_n^i is almost perpendicular to the local skeleton and passes p_n^i. This line can be built by growing circles from p_n^i using the Voronoi method until the outermost circle reaches the outer boundary of the local stroke. Since this seed is on the skeleton, the outermost circle reaches the outer boundary at two points simultaneously. These two points are on opposite sides of p_n^i. Let these two points be q_1 and q_2. A_n^i is defined as $\overline{q_1 q_2}$ as depicted in Fig. 1(a). We define a set of line segments $B_{n,k}^i$ for each p_n^i. $B_{n,k}^i$ connects seed p_n^i and another pixel k on the skeleton, i.e., $B_{n,k}^i = \overline{p_n^i k}$. $B_{n,k}^i$ can not pass the blank(unprinted) area of the character pattern, i.e., $B_{n,k}^i \subset U^i$, where U^i is the set of all pixels in the printed area. We let the orientation of $B_{n,k}^i$ be $\theta_{n,k}^i$ and the length of $B_{n,k}^i$ be $r_{n,k}^i$. For each seed p_n^i, we calculate two different average orientations for the two opposite sides of A_n^i. Let the set of all skeleton pixels which satisfy $B_{n,k}^i \subset U^i$ be denoted by K_n^i. Let the orientation of A_n^i be denoted by θ_n^i; K_n^i can be divided into two different subsets $K_n^{i\,\prime}$ and $K_n^{i\,\prime\prime}$ as follows:

$$\begin{aligned} K_n^{i\,\prime} &= \{k \mid 0 \leq \theta_{n,k}^i - \theta_n^i < \pi, k \in K_n^i\} \\ K_n^{i\,\prime\prime} &= \{k \mid \theta_{n,k}^i - \theta_n^i < 0 \text{ or } \pi \leq \theta_{n,k}^i - \theta_n^i, \\ &\qquad k \in K_n^i\}, \end{aligned} \tag{1}$$

where $K_n^{i\,\prime} \cup K_n^{i\,\prime\prime} = K_n^i$ and $K_n^{i\,\prime} \cap K_n^{i\,\prime\prime} = \emptyset$. Let $|K_n^{i\,\prime}|$ and $|K_n^{i\,\prime\prime}|$ be the numbers of pixels in $K_n^{i\,\prime}$ and $K_n^{i\,\prime\prime}$, respectively. We obtain the average orientations of $B_{n,k}^i$ for the sets $K_n^{i\,\prime}$ and $K_n^{i\,\prime\prime}$, respectively, as follows:

$$\begin{aligned} \overline{\theta_n^{i\,\prime}} &= \frac{1}{|K_n^{i\,\prime}|} \sum_{k \in K_n^{i\,\prime}} \theta_{n,k}^i r_{n,k}^i \quad \text{and} \\ \overline{\theta_n^{i\,\prime\prime}} &= \frac{1}{|K_n^{i\,\prime\prime}|} \sum_{k \in K_n^{i\,\prime\prime}} \theta_{n,k}^i r_{n,k}^i. \end{aligned} \tag{2}$$

Note that the sets $K_n^i, K_n^{i\,\prime}$, and $K_n^{i\,\prime\prime}$ may contain other seeds. For each p_n^i, we extract two values, $\overline{r_n^{i\,\prime}}$ and $\overline{r_n^{i\,\prime\prime}}$, which are the average lengths of $B_{n,k}^i$ for the two sets $K_n^{i\,\prime}$ and $K_n^{i\,\prime\prime}$, respectively. $\overline{r_n^{i\,\prime}}$ and $\overline{r_n^{i\,\prime\prime}}$ are obtained by:

$$\begin{aligned} \overline{r_n^{i\,\prime}} &= \frac{1}{|K_n^{i\,\prime}|} \sum_{k \in K_n^{i\,\prime}} r_{n,k}^i \\ \overline{r_n^{i\,\prime\prime}} &= \frac{1}{|K_n^{i\,\prime\prime}|} \sum_{k \in K_n^{i\,\prime\prime}} r_{n,k}^i. \end{aligned} \tag{3}$$

The 5 elements of vector $\mathbf{s}_n^i$ are obtained as follows. Let x_n^i and y_n^i be the coordinates of the seed p_n^i. We define two lines(arms) which extend from p_n^i in the directions $\overline{\theta_n^{i\,\prime}}$ and $\overline{\theta_n^{i\,\prime\prime}}$ with lengths $\overline{r_n^{i\,\prime}}$ and $\overline{r_n^{i\,\prime\prime}}$, respectively. Let the end points of the two arms be t_1 and t_2. Let $\widetilde{\phi_n^i}$ be the angle between the two arms $\overline{p_n^i t_1}$ and $\overline{p_n^i t_2}$, where $0 \leq \widetilde{\phi_n^i} \leq \pi$. Let $\phi_n^i = \frac{\pi - \widetilde{\phi_n^i}}{\pi}$. Let $(\hat{x_j^i}, \hat{y_j^i})$ be a unit vector which extends from p_n^i and equally divides the angle $\widetilde{\phi_n^i}$. Let $u_n^i = (\overline{r_n^{i\,\prime}} + \overline{r_n^{i\,\prime\prime}})/2$. We define $\mathbf{s}_n^i = \left[x_n^i, y_n^i, u_n^i, \phi_n^i \hat{x_n^i}, \phi_n^i \hat{y_n^i}\right]^T$. These element parameters are depicted in Fig. 1(a). Fig. 1(b) and (c) show the representation for the character 'a'.

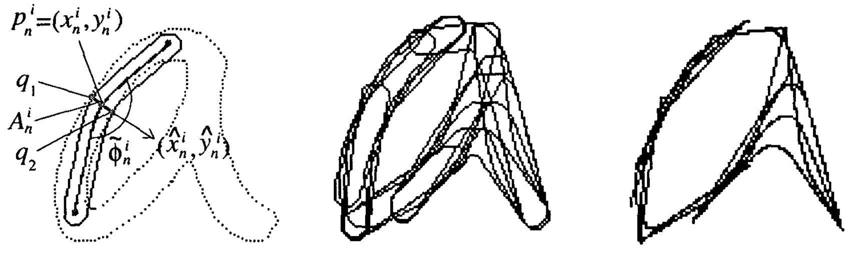
We will use the ellipses which satisfy $\{0.8 \leq \overline{r_n^{i\,\prime}} / \overline{r_n^{i\,\prime\prime}} \leq 1.2\}$ and $\{\overline{r_n^{i\,\prime}} > r^*, \overline{r_n^{i\,\prime\prime}} > r^*\}$, where r^* is a predetermined small positive constant. This means that we discard the deformity ellipses and ellipses with small $\overline{r_n^{i\,\prime}}$ or $\overline{r_n^{i\,\prime\prime}}$. Including the various deformity ellipses will increase the system complexity in the recognition process. We may sample enough seeds to obtain acceptably representative ellipses. Using few large ellipses to reduce the number of features is not feasible in our method. The seeds are sampled uniformly from the skeleton pixels. The same process is also applied to the unknown handprinted pattern to obtain the mth feature vector $\mathbf{h}_m = [x_m, y_m, u_m, \phi_m \hat{x_m}, \phi_m \hat{y_m}]^T$. The elements of $\mathbf{h}_m$ are defined in a way similar to that for the elements of $\mathbf{s}_n^i$, where the subscript n is replaced by m.

2.2 Method 2

A different method can be used to generate the bended-ellipse receptive fields. We sample seeds $p_n^i, 1 \leq n \leq N^i$, of the ith standard character pattern as in the above subsection. We then select the most significant directions (MSDs) for each seed. An MSD is defined as the direction in which the length of the line extending from the seed in this direction is a local maximum. The line must not pass blank(unprinted) area. All the directions in $[0, 2\pi)$ are candidates of the MSDs. We may subsample all the directions within $[0, 2\pi)$ to decrease the computation load without loss of accuracy. Typically, the direction is subsampled by every angle $\frac{2\pi}{72}$, which results in a total of 72 directions. We then grow lines from the seed along these directions. The lines stop growing when they reach the outer boundary of the local stroke. This is shown in Fig. 2. Let $L_n^i(l)$ denote the line segment growing from p_n^i in direction l and terminating at the outer boundary. The MSDs are determined by finding the directions which satisfy the following rule:

Direction l is a MSD if

$$\begin{aligned} &\frac{1}{2v+1} \sum_{j=-v}^{v} |L_n^i(l + j \cdot \frac{\pi}{36})| > \\ &\frac{1}{2v+1} \sum_{j=-v}^{v} |L_n^i(l + (j+o) \cdot \frac{\pi}{36})| \quad \text{and} \\ &\frac{1}{2v+1} \sum_{j=-v}^{v} |L_n^i(l + j \cdot \frac{\pi}{36})| > \end{aligned}$$



(a) (b) (c)

Figure 1: The bended ellipses fitted to the stroke. (a) A fitted bended ellipse and its parameters. (b) All the bended ellipses in the character 'a'. (c) The feature representations for 'a'.

$$\frac{1}{2v+1}\sum_{j=-v}^{v}|L_n^i(l+(j-o)\cdot\frac{\pi}{36})| \quad \text{for } o=1,\ldots,O \tag{4}$$

where v is a constant which denotes the window size for averaging and O is the neighborhood range from which the maximum is chosen. $|L_n^i|$ denotes the length of the line L_n^i. Usually, we define $O=1$. In Eq. (4), we calculate the average length of $L_n^i(l)$ by averaging over the lengths within the window. To find the local maximum, we compare the average length around $L_n^i(l)$ to the average lengths around its $2 \times O$ neighboring directions. The direction l is an MSD if the average length around $L_n^i(l)$ is a local maximum. We call $L_n^i(l)$ an MSD line if l is an MSD. In Fig. 2, the two longest lines are the MSD lines. We use $v=1$ and $O=1$ in this work.

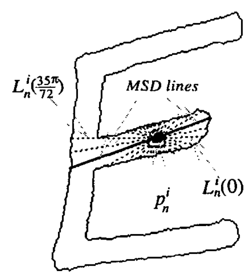


Figure 2: The MSD lines.

After finding the MSDs of p_n^i, the 5 elements of the feature vector $\mathbf{s}_n^i$ are determined as follows. The elements x_n^i and y_n^i are the coordinates of p_n^i. Let $\widetilde{\phi_n^i}, 0 \leq \widetilde{\phi_n^i} \leq \pi$, be the angle between the two MSD lines. We define $\phi_n^i = \frac{\pi-\widetilde{\phi_n^i}}{\pi}$. Let $(\hat{x_n^i}, \hat{y_n^i})$ be a unit vector which extends from the seed and equally divides the angle $\widetilde{\phi_n^i}$. Let u_n^i be the average length of the two MSD lines. As in the previous subsection, we discard those deformity ellipses and small ellipses. After applying the method to all the seeds of the ith standard character pattern, we obtain a set of feature vectors, $\mathbf{S}^i = \{\mathbf{s}_n^i | 1 \leq n \leq N^i\}$.

3 The Devised Self-Organization Network

To classify the feature vectors obtained using the methods described in the previous section, we have devised a self-organizing network to define the correspondences between pairs of feature vectors across the handprinted pattern and a standard pattern. We then measure the similarity between these two patterns according to such correspondences. The standard character pattern which is most similar to the unknown handprinted pattern is selected as the classification result.

The idea of the design is to use the self-organizing process efficiently such that an unknown handprinted pattern can evolve into the correct template pattern with control over training parameters. Those patterns which do not have the same topology will evolve into different templates. Each character may use many template patterns to cope with topology differences. These template patterns are obtained experimentally from a database[1]. Note that whenever a handprinted pattern cannot properly evolve into an existing template, we may include that pattern as a new template to save time in designing various templates. With this selection for templates, the number of templates for each standard character is no more than 13.

The neuron support is reduced in the devised network and is similar in appearance to the template character. Usually, a neuron support has the appearance as a plane square with neurons on the regular grid points. Since the training patterns come from a handprinted character, the hierarchical topology relations of its features have the same structure as the character in a two-dimensional plane. The network can deduce the topology order gradually from the relations that are implicit in the training patterns (features). According to [1], these features may even be very crude, nonlinear,

[1]The database is the NIST Special Database 19.

and mutually dependent in high dimensional space. In our case, they are in 5-dimensional space. When we use a square support, the appearance of the matched neurons on the support is similar to that of the training character. Figs. 3(a) show these cases. We may discard those neurons on the square support which are undefined, neglected, or not well matched. The remaining neurons can be used to configure a new neuron support, which has fewer neurons and has the appearance of the training character. Fig. 3(b) shows this reduced support. We will use this reduced support to replace square support in the training process. Those discarded neurons will be ignored in the training process. This means that the object topological relations are fixed in the support. The process will converge for patterns with similar topology. Each devised support is configured only for one specific character template. The training feature vectors will be forced to match this devised support. A match can always be obtained for training patterns having the same topology. Each template can be effectively matched using its various handprinted patterns. We expect this match to be able to withstand various distortions which can not be overcome using other methods. We will describe the devised process below.

The neurons in the network are distributed among the locations of the seeds of the standard template pattern. Each neuron contains 5 synapses (weights). This is shown in Fig. 3(b). The 5 weights constitute a 5-dimensional weight vector. The weight vector of the nth neuron is initialized with the vector $\mathbf{s}_n^i$. All the neurons constitute the neuron support which is the same in appearance as the ith standard character. We use $\mathbf{w}_n^i(t)$ to denote the weight vector of the nth neuron in the ith neuron support at iteration time t(or cycle t). We set $\mathbf{w}_n^i(0) = \mathbf{s}_n^i$. The algorithm consists of two operations: selecting and updating. For a training feature vector $\mathbf{h}_m$ which is the mth feature vector of an unknown handprinted pattern, we select a representative neuron in the ith neuron support. The similarity between this training vector and the weight vector of a neuron is defined as:

$$D(\mathbf{h}_m, \mathbf{w}_n^i(t)) = \|\mathbf{h}_m - \mathbf{w}_n^i(t)\|. \qquad (5)$$

The representative neuron $\dot{n}$ is the neuron which is closest to the training vector:

$$D(\mathbf{h}_m, \mathbf{w}_{\dot{n}}^i(t)) = \min_{1 \le n \le N^i} D(\mathbf{h}_m, \mathbf{w}_n^i(t)). \qquad (6)$$

After selecting the representative neuron, the weight vectors of the neighboring neurons of this representative neuron are updated using the following amount:

$$\Delta \mathbf{w}_n^i(t) = \alpha(t)(\mathbf{h}_m - \mathbf{w}_n^i(t)), \quad n \in C_{\dot{n}}, \qquad (7)$$

where $C_{\dot{n}}$ is the neighborhood of the representative neuron. $\alpha(t)$ is a controlled training constant which is obtained experimentally for each template character. All the training feature vectors are randomly presented to the neuron support in a training cycle. We update the value of $\mathbf{w}_n^i(t)$ once per cycle; i.e., the update values obtained in Eq. (7) for all $\mathbf{h}_m$ are accumulated within each cycle and added to $\mathbf{w}_n^i(t)$ before the next cycle begins. Note that all the elements in the vector $\mathbf{W}_n^i(t)$ are kept within their limits during the training, $0 \le x_n^i, y_n^i, u_n^i, \phi_n^i \le 1$.

The controlled parameter $\alpha(t)$ decreases linearly from 1 to 0 as the number of training cycles increases. The number of training cycles is experimentally determined for each character template. Any handprinted pattern which has the same topology as does the template can match this template within the determined number of cycles. Properly setting the maximal number of training cycles will guarantee correct convergence of the weight vectors. See Fig. 4 for an example. A correct convergence has fewer matching distortions (shown by the dashed lines). Usually, the number of cycles is within the range 50-100. Our design can restore the rotation of a handprinted pattern within $(-30, 30)$ degrees as shown in the last two patterns in Fig. 4.

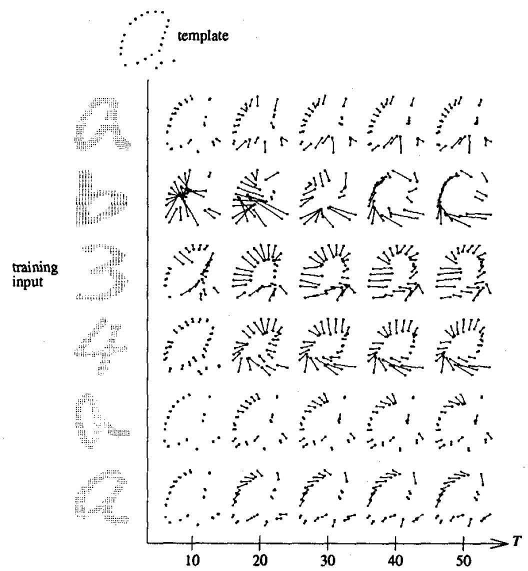
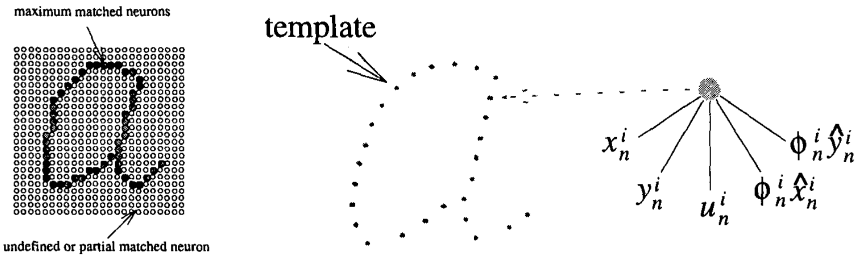


Figure 4: Different handwritten patterns, 'a', 'b', '3', and '4', were used as training patterns in the network with template support 'a'. The dashed lines show the matching distortions. A global rotation of the dashed vectors can significantly reduce the $\mathcal{D}$ values for the last two patterns.

After the training time T ends, we select the template which has the minimum dissimilarity with the classification result. The dissimilarity is measured as follows:



(a) (b)

Figure 3: (a) Neuron support with neurons on square grid points. (b) Neuron support with the same configuration as in the template pattern.

$$\mathcal{D}(\mathbf{H}, \mathbf{S}^i) = \frac{1}{N^i} \sum_{1 \leq n \leq N^i} |\mathbf{w'}_n^i(0) - \mathbf{w'}_n^i(T)|, \quad (8)$$

where $\mathbf{w'}_n^i$ is a 2-D vector which contains the x_n^i and y_n^i components of $\mathbf{w}_n^i$. Remember that the neuron support is initialized by the 5-D features of the standard template pattern, so $\mathbf{w'}_n^i(0)$ represents the geometric shape of the ith standard template character. After training, the weights of the neuron support contain values of matched training features. The distance is just the mean difference of the neurons' locations before and after elastic stretching.

4 Simulation Results

In this work, we used a set of bended-ellipse receptive fields as features to represent character patterns. We devised a self-organization network to classify patterns according to these features. The feature vectors for each standard template character can be prepared and stored in advance. To test the performance, we carried out simulations using the standard 52 English characters and 10 arabic digits as 62 template patterns. Each pattern contained 20-50 feature vectors. Note in this case each standard character has a template.

The feature set $\mathbf{S}^i$ of the ith standard pattern was extracted from a 128 pixel × 128 pixel image using method 2 described in section 2. The handprinted pattern was obtained by scanning the formatted sheet using an optical scanner. The bended-ellipse features were also obtained from the handprinted pattern. The scanned handprinted pattern had a resolution of 128 pixels × 128 pixels. For each standard pattern, we sampled 50 different handprinted patterns from the database to test the performance of the system.

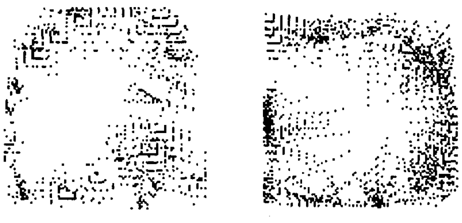
The simulations were carried out on a personal computer. The execution time, rejection rate, and recognition rate are listed in Table 1. There is a similar method [8], called the spatial topology distance in plane space(STD-2D), which does not employ the last 3 elements, $u_n^i, \phi_n^i \hat{x}_n^i$, and $\phi_n^i \hat{y}_n^i$, in the representation. Following that method, we call our method in the table the spatial topology distance in 5-dimensional space(STD-5D). The execution time is longer than that of most character recognition systems because we test the template sequentially for each handprinted pattern. We observe that the elastic matching process takes more than 95% of total execution time.

The accumulated deformation vectors after elastic matching are shown in Fig. 5. All of the 50 handprinted character 'a' patterns were presented separately to the neuron supports of the standard 'a' and 'b' character patterns. The deformation vectors are drawn with line segments and are overlaid to obtain the figure. Long line segments denote serious deformations and will extend across a large area. Based on the density of the overlaid deformation vectors, we can evaluate the correctness of matching. A high density area indicates a large number of overlaid lines. Fig. 5(a) shows the results when the 50 handprinted 'a' patterns were presented to the neuron support of standard pattern 'a'. In (b), the 50 handprinted characters 'a' were presented to the neuron support of standard pattern 'b'. It is obvious that the area of high density in (b) is much larger than that in (a). This suggests that (b) contains many long line segments, resulting from very bad matching due to incorrect topology. On the other hand, the high density area in Fig. 5(a) is localized along the contour of neuron support 'a'. In this case, the standard character pattern 'a' tends to be the best classification choice.

The correctness of this classification can be further verified by investigating the probability distribution of distances measured using Eq. 8 after many matchings. The distribution of the measured distances is shown in Fig. 6. For each standard character pattern, we show the distance distribution for both correct and incorrect handprinted patterns. Here, a 'correct matching' was obtained by per-

	Accurate Recognition Rate	Rejection Rate	Execution Time
STD-5D	92.3	1	1.2 sec
STD-2D	90.3	2	1.0 sec

Table 1: The performance.



(a) (b)
Figure 5: The accumulated deformation vectors.

forming elastic matching between the handprinted and standard patterns of the same class; e.g., handprinted pattern and standard patterns were both 'A'. On the other hand, 'incorrect matching' was done between patterns of different classes, e.g., 'A' and 'B'. The left column of the line segments for each standard pattern depicts the distance distribution of the correct matchings. The right column depicts the distribution of incorrect matchings. The length of each horizontal line segment displays the probability of the distance. We observe that there is a gap in between the maximum of the correct matchings and the minimum of the incorrect matchings. The credit threshold is in the middle of this gap. The 50 handprinted patterns could always be classified. The simulations provide sound support for developing of the representations and the network.

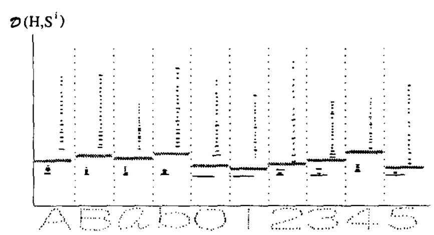


Figure 6: The distance distributions.

References

[1] T. Kohonen, *Chapter 5.4.2 & 5.4.3 in Self-organization and Associative Memory 2nd ed.*, Berlin:Springer-Verlag, 1988.

[2] S. Mori, K. Yamamoto, and M. Yasuda, "Research on machine recognition of handprinted characters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 386-405, Jul. 1984.

[3] T. Hildebrandt and W. Liu, "Optical recognition of handwritten Chinese characters: advances since 1980," *Pattern Recognition*, vol. 26, no. 2, pp. 205-225, 1993.

[4] E. H. Land, "The retinex theory of color vision," in *The Perceptual World*, edited by I. Rock, New York: W. H. Freeman and Company, pp. 3-24, 1990.

[5] D. J. Burr, "Elastic matching of line drawings," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 708-713, Nov. 1981.

[6] C. C. Tappert, "Cursive script recognition by elastic matching," *IBM J. Res. Develop.*, vol. 26, pp. 765-771, Nov. 1982.

[7] R. S. Mitra and N. N. Murthy, "Elastic, maximal matching," *Pattern Recognition*, vol. 24, pp. 747-753, 1991.

[8] C. Y. Liou and H. C. Yang, "Handprinted Character Recognition Based on Spatial Topology Distance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.18, pp.941-945, 1996.

[9] E. Bienenstock and R. Doursat, "Issues of representation in neural networks," *Representations of Vision: Trends and Tacit Assumptions in Vision Research*, edited by A. Gorea, pp. 47-67, Cambridge:Cambridge University Press, 1991.

[10] G. E. Hinton, C. K. I. Williams, and M. D. Revow, "Adaptive elastic models for hand-Printed character recognition," in *Advances in Neural Information Processing Systems 4*, edited by J. E. Moody, S. J. Hanson and R. P. Lippmann, Morgan Kauffmann, 1992.