

Correspondence

Matching of Complex Patterns by Energy Minimization

C. H. Leung and Ching Y. Suen

Abstract—Two patterns are matched by putting one on top of the other and iteratively moving their individual parts until most of their corresponding parts are aligned. An energy function and a neighborhood of influence are defined for each iteration. Initially, a large neighborhood is used such that the movements result in global features being coarsely aligned. The neighborhood size is gradually reduced in successive iterations so that finer and finer details are aligned. Encouraging results have been obtained when applied to match complex Chinese characters. It has been observed that computation increases with the square of the number of moving parts which is quite favorable compared with other algorithms. The method was applied to the recognition of handwritten Chinese characters. After performing the iterative matching, a set of similarity measures are used to measure the similarity in topological features between the input and template characters. An overall recognition rate of 96.1% is achieved.

Index Terms—Elastic matching, handwritten Chinese character recognition, pattern matching.

I. INTRODUCTION

The matching of structural patterns is a very difficult problem when the patterns are complex and distorted. Since computational complexity grows rapidly with the complexity of the problem, the applicability of many algorithms is greatly limited. For example, the relaxation labeling method has a complexity proportional to N^4 where N is the number of feature points [1] or line elements [2]–[4] in each pattern. Syntactic methods have also been proposed but their applicability has only been demonstrated with simple patterns of mild distortions [5], [6]. There is much difficulty in grammatical inference and parsing when both the complexity and distortions are considerable, e.g., with characters with broken, touching and missing strokes, etc. Other methods include dynamic programming [7], elastic matching [8], [9], and neural networks [10]–[13].

An alternative approach is to formulate the problem of template matching as a minimization problem. The template pattern is elastically deformed in order to match with the input pattern. The objective is to achieve maximum similarity between the resulting patterns while minimizing the deformation. This is achieved through the minimization of a cost (or energy) function [11], [14]–[17]. A variety of energy functions are employed with applications including the matching of brain contours [14], character patterns [15], contours of the mitral valve of the beating heart [16], stereo images [17], and other miscellaneous shapes [11]. Methods along a similar direction but different in details and interpretations have also been proposed [18], [19].

In this article, the approach of pattern matching by energy minimization is adopted. A preliminary report has been given by the author at a conference [15]. The present article contains a compre-

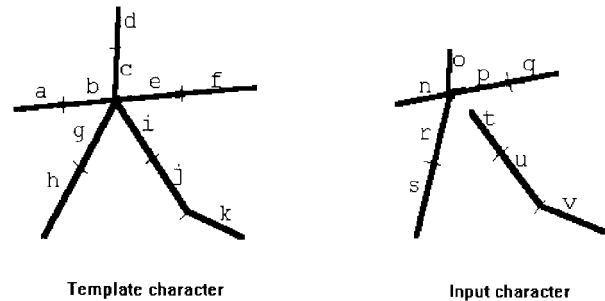


Fig. 1. Illustration of pattern matching.

hensive account as well as its extension to complex handwritten Chinese character recognition. The main features of this method are the applicability to the matching of complex patterns and the relatively low computational complexity.

In the proposed method, one pattern is put on top of the other and the individual parts are iteratively moved until most of the corresponding parts are aligned. A neighborhood of influence is defined for each iteration. Initially, a large neighborhood is used which results in global features being coarsely aligned. The neighborhood size is gradually decreased in successive iterations resulting in finer and finer details being aligned. An energy function is defined for each iteration so that optimal matching can be achieved by minimizing this function. This top-down procedure is found to be quite effective in arriving at good matches.

In the following sections, a brief review of the method proposed by the authors in a previous conference [15] is given, as well as its extensions to the recognition of complex handwritten Chinese characters.

II. THEORY OF THE METHOD

A. Iterative Matching

The patterns to be matched are assumed to consist of binary lines and curves, e.g., the edge points of an object or the skeleton of a character. The lines and curves are then approximated by fitting a set of short straight lines of approximately equal lengths. Let us call these short lines “Elements.” In the actual implementation, the pattern was of size 64×64 and each element was about 11 pixels long. Furthermore, let us call the two patterns to be matched the “Template Pattern” and the “Input Pattern.” An example for illustration is shown in Fig. 1. Hence the pattern matching problem becomes that of matching two sets of elements from the template and input patterns. The number of elements in the two patterns need not be equal.

An energy function E_1 is defined for guiding the movements of template elements toward the input elements. It consists of two terms. Qualitatively speaking, the first term measures the distances between each input element and the nearby template elements and thus serves as an overall distance measure between the two patterns. The template pattern is moved in successive steps to align with the input pattern. This introduces distortions in the template pattern and the second term of the energy function is a measure of this distortion. The objective is to align the two patterns as much as possible without excessive distortion. This can be achieved by guiding the movements in such

Manuscript received January 15, 1995; revised March 17, 1996 and July 5, 1997.

C. H. Leung is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong, R.O.C.

C. Y. Suen is with the Centre for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, P.Q. H3G 1M8, Canada.

Publisher Item Identifier S 1083-4419(98)07300-2.

a way that the energy function is minimized. A size parameter is incorporated so that large scale features are aligned at the beginning and finer details at later stages. Details of the energy function are as follows:

$$E_1 = -\alpha K_1^2 \sum_{i=1}^{N_I} \ln \sum_{j=1}^{N_T} \phi(|\mathbf{T}_j - \mathbf{I}_i|, K_1) f(\theta_{T_j, I_i}) + \beta \sum_{j=1}^{N_T} \sum_{k=1}^{N_T} w_{jk} (d_{T_j, T_k} - d_{T_j, T_k}^0)^2 \quad (1)$$

where

N_T	number of template elements;
N_I	number of input elements;
\mathbf{T}_j	position vector of the midpoint of the j th template element;
θ_{T_j}	direction of the j th template element;
\mathbf{I}_i	position vector of the midpoint of the i th input element;
θ_{I_i}	direction of the i th input element;
$\phi(\mathbf{T}_j - \mathbf{I}_i , K_1)$	$= \exp(- \mathbf{T}_j - \mathbf{I}_i ^2 / 2K_1^2)$
θ_{T_j, I_i}	angle between template element T_j and input element I_i , restricted within $0-90^\circ$;
$f(\theta_{T_j, I_i})$	$= \max(\cos \theta_{T_j, I_i}, 0.1)$
d_{T_j, T_k}	current value of $ \mathbf{T}_j - \mathbf{T}_k $
d_{T_j, T_k}^0	initial value of $ \mathbf{T}_j - \mathbf{T}_k $
w_{jk}	$= \phi(\mathbf{T}_j - \mathbf{T}_k , K_2) / \sum_{n=1}^{N_T} \phi(\mathbf{T}_j - \mathbf{T}_n , K_2)$
$K_1 \& K_2$	size parameters of the Gaussian windows which establish neighborhoods of influence, and are decreased monotonically in successive iterations;
α and β	coefficients used to weigh the importance of the two terms.

The first term on the right-hand side of (1) is a measure of the overall distance between elements of the two patterns. The form of the expression is modified from the ‘‘elastic net method’’ proposed by Durbin and Willshaw for solving the traveling salesman problem [20]. For each element \mathbf{I}_i of the input pattern, the summation $\ln \sum_{j=1}^{N_T} \phi(|\mathbf{T}_j - \mathbf{I}_i|, K_1) f(\theta_{T_j, I_i})$ is dominated by the contribution from the nearest template element \mathbf{T}_j with a similar slope. The value of the multiplicative factor $f(\theta_{T_j, I_i})$ is large for similar slopes and small for slopes nearly perpendicular to each other. It is defined as $\max(\cos \theta_{T_j, I_i}, 0.1)$ so that the minimum value is 0.1 instead of 0. This provides the flexibility such that even if the slopes of two elements are at 90° to each other, the chance of match is not reduced to zero. As the size K_1 of the Gaussian window decreases monotonically in successive iterations, in order for the energy E_1 to attain a minimum, each \mathbf{I}_i should have at least one \mathbf{T}_j attracted to it.

The second term consists of a weighted sum of all relative displacements between each template element and its neighbors within the Gaussian weighted neighborhood of size parameter K_2 . Minimization of this term minimizes the structural distortion of the template pattern while each element is being moved. Each template element normally does not move toward its nearest input element but tends to follow the weighted mean movement of its neighbors in order to minimize the distortions within the neighborhood. K_2 is initially chosen to be large so that the distortions of large scale features are kept small and the template elements move collectively to align with the input pattern in a coarse (or global) manner. As K_2 is gradually and monotonically decreased in successive iterations, finer and finer details of the two patterns are aligned. Parameters K_1 and K_2 can have the same initial values but different schedules for decrement in successive iterations. This has the flexibility of controlling the rate of

distortion during the iterations. The actual schedules for decreasing K_1 and K_2 are given by (6) and (7) in Section III.

Minimization of E_1 is carried out by gradient descent. The movement $\Delta \mathbf{T}_j$ applied to \mathbf{T}_j is equal to $-\partial E_1 / \partial \mathbf{T}_j$ and is given by

$$\Delta \mathbf{T}_j = \alpha \sum_{i=1}^{N_I} u_{ij} (\mathbf{I}_i - \mathbf{T}_j) + 2\beta \cdot \sum_{m=1}^{N_T} (w_{mj} + w_{jm}) [(\mathbf{T}_m - \mathbf{T}_m^0) - (\mathbf{T}_j - \mathbf{T}_j^0)] \quad (2)$$

where

$$u_{ij} = \phi(|\mathbf{I}_i - \mathbf{T}_j|, K_1) f(\theta_{I_i, T_j}) / \sum_{n=1}^{N_I} \phi(|\mathbf{I}_i - \mathbf{T}_n|, K_1) f(\theta_{I_i, T_n})$$

\mathbf{T}_j^0 = initial value of \mathbf{T}_j .

Template element \mathbf{T}_j thus moves according to the attraction from the input elements as well as following the weighted mean movements of all template elements within the neighborhood. The weights u_{ij} are normalized so that each input element has the same total influence (attraction) on the template elements. This is important because if an input element is far away from every template element and no normalization is done, then its force attracting template elements will be relatively weak and hence will probably fail to match to any template element.

The above procedure moves the template elements toward the input elements. The minimization of energy E_1 tends to find a matching template element for each input element. However, it does not take an active role to find a matching input element for each template element. To correct this, the roles of the template and input patterns are swapped during each iteration and there are two passes in each iteration. In the first pass, the template elements are attracted to the input elements, and in the second pass, the input elements are attracted to the template elements. A second energy function E_2 which is similar to E_1 is thus defined for guiding the movement of input elements toward template elements

$$E_2 = -\alpha K_1^2 \sum_{i=1}^{N_T} \ln \sum_{j=1}^{N_I} \phi(|\mathbf{I}_j - \mathbf{T}_i|, K_1) f(\theta_{I_j, T_i}) + \beta \sum_{j=1}^{N_I} \sum_{k=1}^{N_I} x_{jk} (d_{I_j, I_k} - d_{I_j, I_k}^0)^2 \quad (3)$$

where

$$x_{jk} = \phi(|\mathbf{I}_j - \mathbf{I}_k|, K_2) / \sum_{n=1}^{N_I} \phi(|\mathbf{I}_j - \mathbf{I}_n|, K_2).$$

The negative of the gradient gives the movement $\Delta \mathbf{I}_j$ applied to \mathbf{I}_j

$$\Delta \mathbf{I}_j = \alpha \sum_{i=1}^{N_T} v_{ij} (\mathbf{T}_i - \mathbf{I}_j) + 2\beta \cdot \sum_{m=1}^{N_I} (x_{mj} + x_{jm}) [(\mathbf{I}_m - \mathbf{I}_m^0) - (\mathbf{I}_j - \mathbf{I}_j^0)] \quad (4)$$

where

$$v_{ij} = \phi(|\mathbf{T}_i - \mathbf{I}_j|, K_1) f(\theta_{T_i, I_j}) / \sum_{n=1}^{N_T} \phi(|\mathbf{T}_i - \mathbf{I}_n|, K_1) f(\theta_{T_i, I_n})$$

K_1 and K_2 in (3) are the same as K_1 and K_2 in (1) above.

B. Pattern Recognition

After all the iterations, the template and input elements will have moved toward each other. Hopefully most of the corresponding elements will overlap or have at least moved closer to each other. The input pattern is matched with every member of the set of template patterns and the “best matched” template is taken as the recognized class. Hence we need to devise a measure for the “goodness of match.” The most natural candidate for this measure is the energy function itself which measures the degree of closeness between the elements of the two patterns and also the amount of distortion suffered by each pattern. A small value indicates good match and vice versa. However, it turned out that this function failed to serve as a reliable classification criterion in practical experiments. An alternative function for measuring the amount of distortion was also tried but the result was again not satisfactory.

Besides this, six other distance measures proposed in [21] and [22] were tested to check whether they can be used for measuring the goodness of match. The first two are nearest neighbor distances. For each black pixel of the template elements, the nearest black pixel belonging to the input elements is identified and the Euclidean distance is measured. The procedure is repeated with the roles of the template and input elements interchanged. The mean distance obtained gives the first measure, while the square root of the mean square distance forms the second measure.

The third measure is the Hamming distance. Each pixel in the template is compared with the pixel at the same position in the input. If they do not agree (i.e., one is black while the other is white), the distance is increased by one. The fourth measure is a similarity measure. Each black pixel in the template is compared with the pixel at the same position in the input. If they are both black, the similarity measure is increased by one. The fifth and sixth measures are modified versions of the fourth one.

Experimental results show that all these measures do not give satisfactory recognition results. The failure may be due to the fact that patterns are often classified according to topological features, e.g., there are two loops for the digit 8. Even though the distortion of the loops may be very severe, we still would classify it as digit 8. Hence if we only use the geometric distortion measure in Euclidean space for classification, the classification will be not be very reliable.

In view of this, the approach finally adopted was to use the matching result as a starting point, and then check the topological features of the two patterns for agreement. A measure of goodness of match is then evaluated. The steps involved are discussed below.

1) *Derivation and Refinement of the Matching List:* After the iterations are over, the positions of the template and input elements are examined and the pairs mutually closest to each other are identified and entered into a matching list. For example, template element b is matched to input element n , as shown in Fig. 1. There will be some unmatched elements, such as template element a . However, due to the variations in size and distortions, the template and input patterns will often have different numbers of elements. Hence one element should sometimes be matched to two and vice versa. For example, template elements a and b should be matched to input element n . Hence the matching list should be refined. The algorithm works by examining each unmatched element (such as a in Fig. 1) to see whether it forms an approximate straight line with its connected neighbor (b). If yes, it should be considered as a match to the same element (n) as its neighbor. After this refinement, the remaining unmatched elements can be considered as truly unmatched. A matching score S_{match} can be evaluated as

$$S_{\text{match}} = 1 - N_u / \min(N_T, N_I)$$

where N_u is the number of unmatched elements.

2) *Evaluation of Directional Conformity:* Although there are variations in the directions of strokes in human handwriting, the variations in the majority of cases should not deviate too much from the expected range of values. Hence a measure of the directional conformity can serve as a measure of match. Specifically, the direction of the i th template element T_i is compared with its matched input element. An allowance of 15° is adopted. If the difference in direction $\Delta\theta_i$ (absolute value) is greater than the allowance, a matching score S_{dir}^i is deducted by an amount proportional to the excess deviation. An overall score for directional conformity S_{dir} is then obtained by taking the mean of the individual scores of each element

$$S_{\text{dir}}^i = 1 - \max(0, \min(\Delta\theta_i - 15, 60))/60$$

$$S_{\text{dir}} = E[S_{\text{dir}}^i].$$

The expression for S_{dir}^i above maps $\Delta\theta_i$ within $[0^\circ, 15^\circ]$ to the single output value of 1.0, while deviations within $[15^\circ, 75^\circ]$ are mapped to $[1, 0]$, and that within $[75^\circ, 90^\circ]$ are mapped to 0. The rationale is that a deviation of less than 15° in stroke direction is an acceptable value for handwritten Chinese characters and no penalty should be imposed, while a deviation of more than 75° is considered extremely unlikely and should be given the full penalty (for Chinese characters, the stroke directions are quite significant).

3) *Evaluation of Connectivity Conformity:* A continuous stroke should be matched to a continuous stroke and a crossing point should be matched to a crossing point. Although there may be occasional breakage due to noise, we can still get a reasonably reliable measure if we examine many elements and take the overall statistics. For each pair of connected template elements such as g and h in Fig. 1, we examine the corresponding matched pair r and s in the input pattern to see whether they are also connected. If connected, a score of 1 is given. Otherwise a 0 is recorded. After examining all connected pairs, we can arrive at a mean score for the conformity of connectivity S_{conn} .

Unfortunately, this measure may not be reliable. In handwriting, some strokes which normally do not touch each other are sometimes drawn in such a way that they touch by pure chance. For example, input element t in Fig. 1 does not touch the junction formed by n , p , and r , but the corresponding template element i is drawn to touch with the junction. Hence if the template is taken as the reference, this will contribute to a negative score.

One solution is to build a knowledge base which specifies whether each connection is mandatory or not, and only the mandatory connections are examined for conformity. Another way is to collect a large number of sample characters and compute the statistics on the probability of connectivity. The former method is labor intensive while the latter requires a large number of samples for each character category which may be difficult to realize for Chinese characters having more than 5000 character categories in daily use. If no action is taken to avoid this pitfall, we may have to reduce the weighting of this score (for the conformity of connectivity) in the final matching score calculation in order to reflect its inherent uncertainty.

4) *Evaluation of Curvature Conformity:* Chinese characters are made up of both straight and cursive strokes. For the cursive strokes, some are drawn in a clockwise direction while others are counterclockwise. Hence the degree of match in curvature can serve as some indication of the similarity of the matching characters. For each pair of connected template elements such as j and k (in Fig. 1), the corresponding input pair u and v are examined to check whether the angular deviation in going from j to k is about the same as that from u to v . Depending on the difference in curvature ΔC_{jk} (absolute value in degrees), a matching score S_{cur}^{jk} can then be computed

$$S_{\text{cur}}^{jk} = 1 - \max(0, \min(\Delta C_{jk} - 15, 30))/30.$$

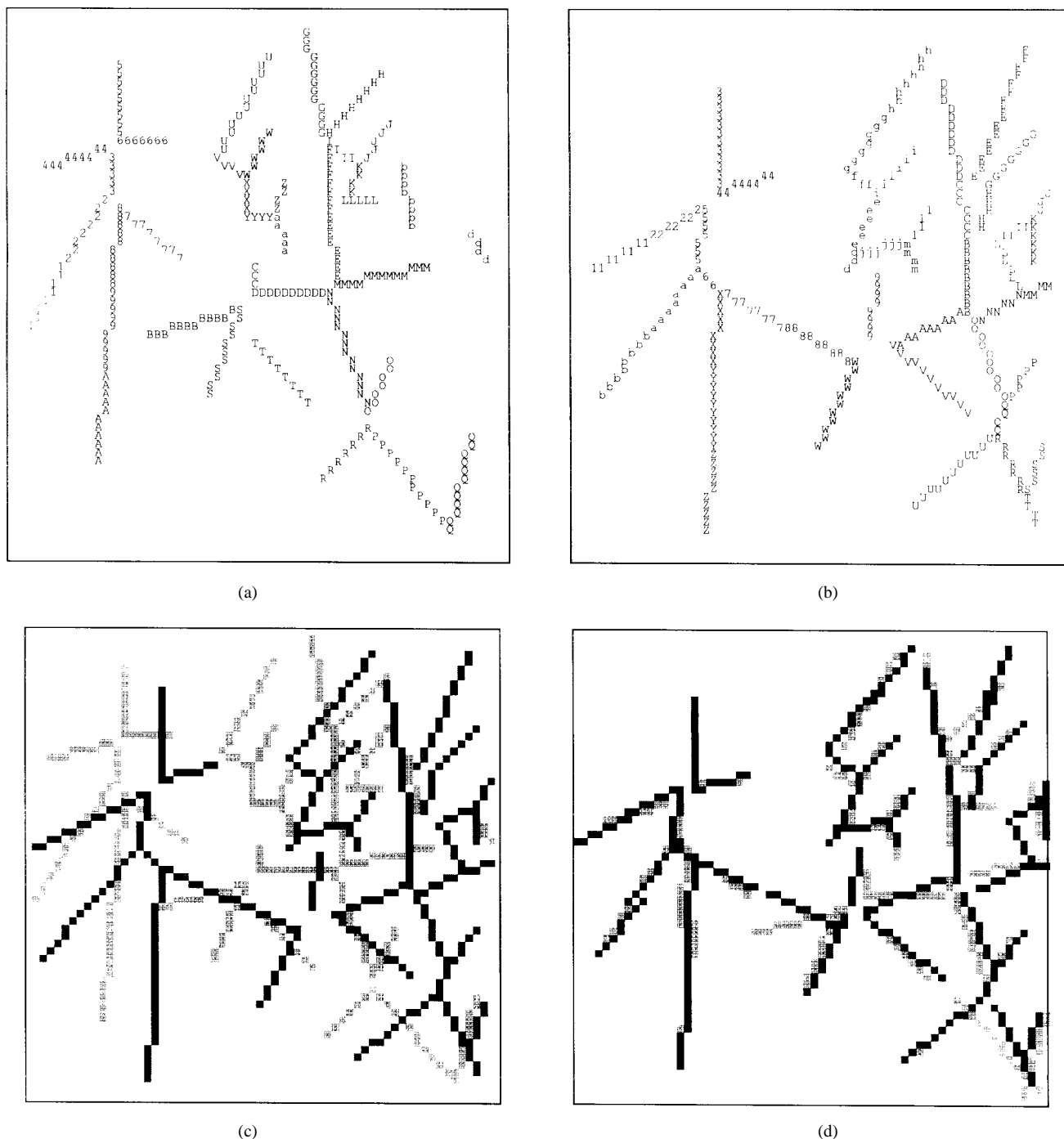


Fig. 2. (a) Template image, elements are labeled by alphanumeric characters, (b) input image, (c) overlapped images of (a) and (b) before pattern matching, and (d) overlapped images of template and input images after matching.

The form of the expression is similar to that of S_{dir}^i discussed previously. The choice of the threshold values is based on our experience. For those elements connected to a multi-element junction such as the junction comprising b, c, e, i , and g , the pairs most similar in slope are identified. For example, g and c form one pair, b and e form another, and i is alone. The corresponding pairs in the other pattern are then checked for similarity in curvature. After an exhaustive checking, a mean score for the conformity in curvature S_{cur} can be evaluated.

$$S_{cur} = E[S_{cur}^{jk}].$$

5) *Incorporation of Preclassification Statistics:* Since iterative matching is time consuming, it is suitable as a validation tool in the final-stage of recognition. There should be a preclassification stage to screen out a small set of most probable classes from the thousands of candidate classes. Usually, a statistical classifier is employed which gives some figures to indicate how similar a template is to the input. In our experiment, we used the conventional Bayes classifier assuming Gaussian statistics. The output of the classifier is proportional to the logarithm of the probability of the template being the same class as the input. This log probability can contribute to measuring how good the match is. In our experiment, the most probable ten candidate

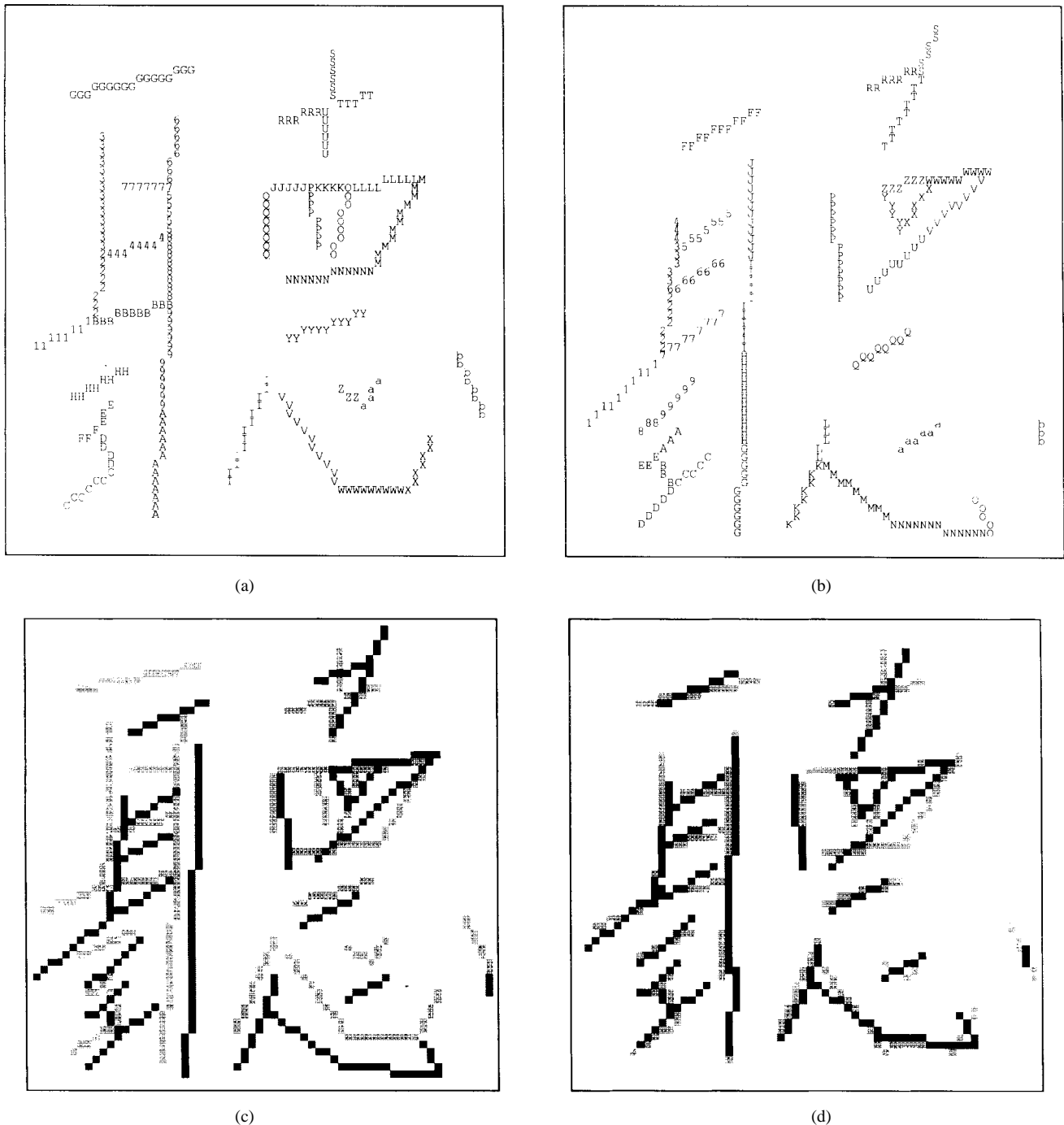


Fig. 3. (a) Template image, elements are labeled by alphanumeric characters, (b) input image, (c) overlapped images of (a) and (b) before pattern matching, and (d) overlapped images of template and input images after matching.

classes from the preclassifier are sorted out. Their log probabilities $P(k), k = 1, \dots, 10$ are converted into a matching score S_{pre}^k for the k th candidate as follows:

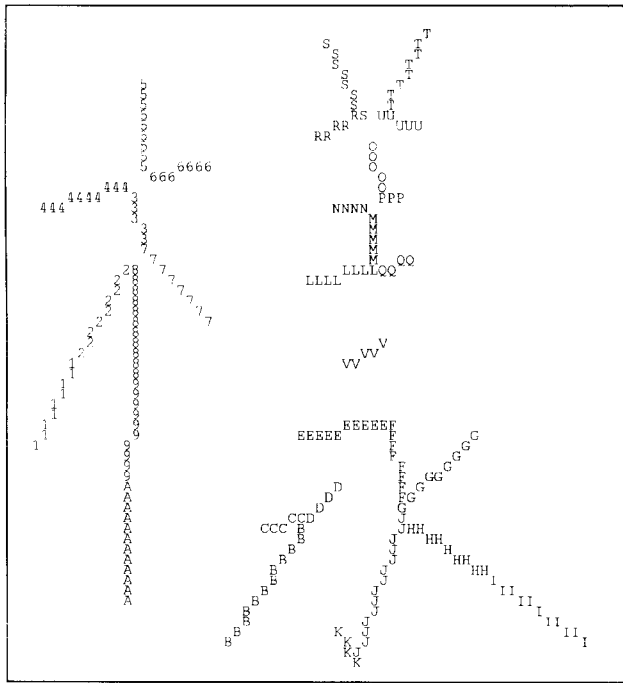
$$S_{pre}^k = 1 - \max(0, \min(P(1) - P(k) - 5, 30))/30.$$

Again, the form of the expression is similar to the previous measures. The threshold values were chosen after some analyses on the output values of the preclassifier had been carried out.

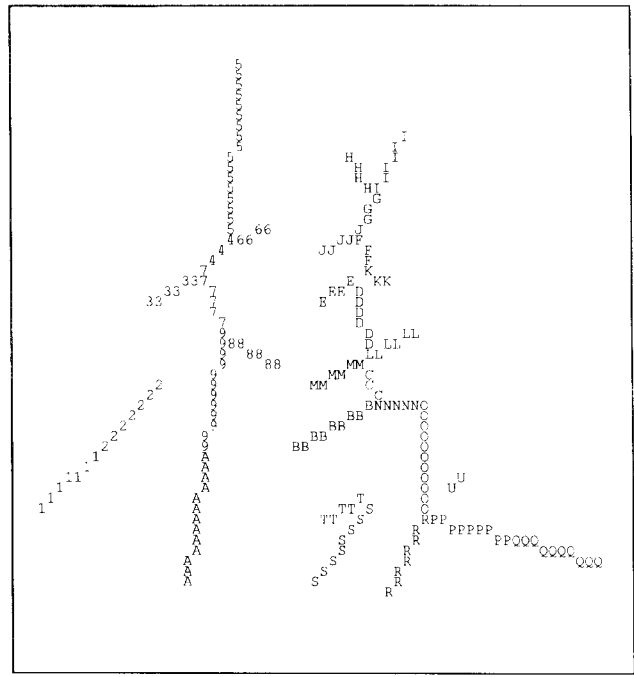
6) *Evaluation of Overall Score for Recognition Decision:* Each of the five scores is not always reliable. However, by combining

them into an overall score, the reliability tends to be higher. The simplest way is to use a linear combination of each score S_i with an attached weight w_i . The set of weights can be the same for all classes. Alternatively, some scores may be more effective in classifying some patterns. For example, if a pattern contains a lot of curve strokes, the curvature score may be more significant. Hence a different set of weights for the five scores may be adopted for different classes. Alternatively, a neural network may be used to combine the scores [18], [23].

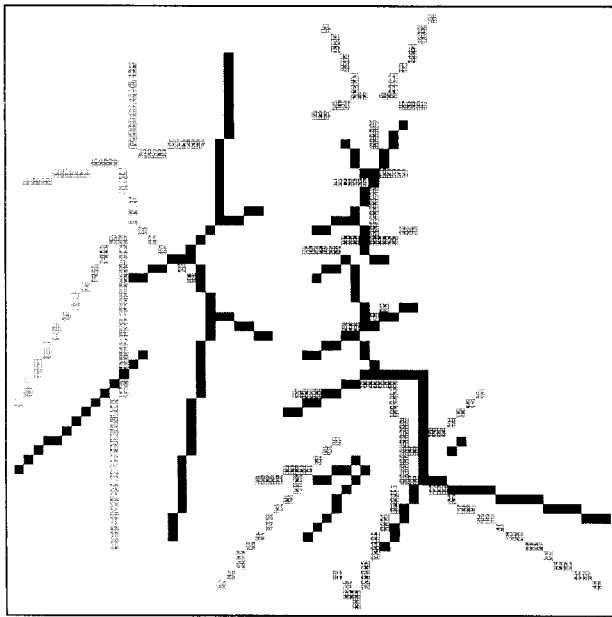
In practice, the optimal approaches require a large number of training samples. In the present experiment, most of the available



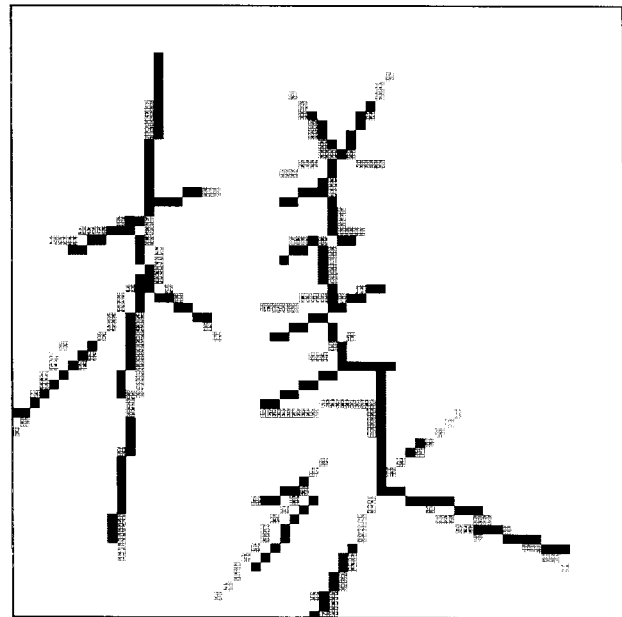
(a)



(b)



(c)



(d)

Fig. 4. (a) Template image, elements are labeled by alphanumeric characters, (b) input image, (c) overlapped images of (a) and (b) before pattern matching, and (d) overlapped images of template and input images after matching.

samples were used up for training the statistical preclassifier and only a small portion was left for testing. Hence a simple approach was taken. The variances of the scores as well as the reliability of its classification output were examined and a set of weights was adopted by taking these into consideration. A large weight does not necessarily mean that it is more important than others. It may be due to the small variance of the particular score compared with others so that a larger weight is needed to restore its effect on the classification decision. The set of weights are the same for all classes. Specifically,

the total score is computed according to

$$S_{\text{total}} = w_1 S_{\text{match}} + w_2 S_{\text{dir}} + w_3 S_{\text{conn}} + w_4 S_{\text{cur}} + w_5 S_{\text{pre}} \quad (5)$$

where w_1, w_2, \dots, w_5 are given values of 0.4, 0.4, 0.2, 0.1, and 0.5, respectively.

III. EXPERIMENTAL RESULTS

The proposed algorithm was first evaluated by applying it to the matching of complex handwritten Chinese characters. It was subse-

quently incorporated into a handwritten Chinese character recognition system for further evaluation.

A. Pattern Matching

In the matching experiment, each character was first normalized to a 64×64 binary matrix. Preprocessing steps included thinning and approximation of the thinned pattern by straight lines. Each straight line was then divided into equal-length elements of about 11 pixels long. The midpoint and slope of each element was computed and stored into a data structure. This data structure constituted the input to the present matching algorithm.

The initial neighborhood sizes K_1 and K_2 in (1)–(4) were chosen to be ten pixels. The coefficients α and β were equal to 0.12 and 0.21, respectively, and were kept constant throughout the iterations. The choice of α and β was based on a separate experiment in which a number of patterns were matched using different values of α and β and the detailed iterative sequences were interactively examined. The values which gave the best results were adopted. It was observed that the exact values of α and β were not critical. The second significant digit had only a slight impact on the matching result.

To reduce the frequent computation of weight matrices u , v , w , and x , their values as well as the values of K_1 and K_2 were not updated until ten iterations had been performed. Each iteration consisted of two passes, one pass for moving the template segments and another for the input segments according to (2) and (4), respectively. After each set of ten iterations, the values of K_1 and K_2 were decreased according to the following formulas:

$$K_1 := K_1 - \max(0.4, 0.15K_1) \quad (6)$$

$$K_2 := K_2 - \max(0.4, 0.10K_2). \quad (7)$$

Iterations were stopped when K_1 was reduced to below 1.5 pixels. With K_1 equal to ten to start with, 120 iterations were needed. The schedules for decreasing K_1 and K_2 were made different. It was found that if they were made the same, distortions during the last few iterations (when K_1 and K_2 were about two pixels) would be too much. By decreasing K_2 at a lower rate than K_1 , the neighborhood size governing the distortion terms in (1) and (2) would be larger so that more neighboring elements would interact among themselves to preserve the local shape better.

Regarding the problem of local minima and stability, extensive trials showed that the algorithm is quite stable. Usually the local minima reached are reasonably good solutions (i.e., the patterns are quite well matched). The patterns are observed to approach each other smoothly, with greater movements during the initial iterations and less movements later on. Only on rare occasions will the pattern get stuck in a local minimum which represents a poor solution. Moreover, oscillations between the input and template patterns have not been observed.

Examples of matching are shown in Figs. 2–4 and Tables I–III. In each figure, the template and input characters (after thinning and approximation by straight elements) are shown in Figs. 2(a)–4(a) and 2(b)–4(b), respectively. Each element is given an alphanumeric label. The characters are overlapped and shown in Figs. 2(c)–4(c). Iterative pattern matching was performed and the results are shown in Figs. 2(d)–4(d). After matching, the pairs of template and input elements closest together were sought and put into the matching lists. The lists were then refined according to the algorithm discussed in the Section II-B1. The refined matching lists for Figs. 2–4 are shown in Tables I–III, respectively.

TABLE I
MATCHING LIST FOR THE TEMPLATE AND INPUT
IMAGES SHOWN IN FIG. 2(a) AND (b), RESPECTIVELY

Template element	Input element (s)
1	b
2	a
3	5
4	2 , 1
5	3
6	4
7	7 , 6
8	X
9	Y
A	Z
B	8
C	9
D	A
E	B
F	C
G	D
H	E , F
I	no match
J	G
K	H
L	no match
M	N , M
N	O
O	P , Q
P	R , T
Q	S
R	U
S	W
T	V
U	g , h
V	f
W	i
X	e , d
Y	j
Z	l
a	m
b	I , J
d	no match

Inspection of the figures shows that the algorithm succeeds in finding the correct match pairs most of the time. The failures are often due to the lack of correspondence between the template and input elements. Timing statistics shows that the computational time is roughly proportional to $(N_T + N_I)^2$ (excluding preprocessing time, such as the time taken for thinning). It takes about 1 s to match two patterns of 20 elements each (or 4 s for patterns with 40 elements each) on the VAX-6320 computer. About 30% of the total time is spent on updating the weight matrices u , v , w , and x (only updated once in every ten iterations) in (1)–(4), and the rest of the time is on updating the coordinates of each element according to (2) and (4) in each iteration.

B. Character Recognition

In the recognition experiment, the Chinese character set consisted of the 240 most frequently used character categories (classes). A

TABLE II
MATCHING LIST FOR THE TEMPLATE AND INPUT
IMAGES SHOWN IN FIG. 3(a) AND (b), RESPECTIVELY

Template element	Input element (s)
1	1
2	2 , 3
3	4
4	6
5	I
6	J
7	5
8	I
9	H
A	G
B	7
C	D , C
D	B
E	A
F	E
G	F
H	9
I	K , L
J	Z
K	Z
L	W
M	V
N	U
O	X
P	Y
Q	P
R	R
S	S
T	R
U	T
V	M
W	N
X	O
Y	Q
Z	a
a	a
b	b

training set with 7680 handwritten samples (32 samples per class with each sample within the same class written by a different author) was used to train a statistical preclassifier which was a conventional Bayes classifier assuming Gaussian statistics. The feature extraction method was given in [24] and the feature dimension was 28. To make up for the insufficient number of training samples, a distortion model was employed to generate 600 samples per class (total 144 000 samples) by applying random shearing and local expansion and contraction distortions to each training sample [25]. The test set consisted of 960 samples (4 samples/class, each sample from the same class is written by a different author). Authors writing the training set were not the same as those writing the test set. The recognition rate was 86.6% which was rather low. However, if the best ten candidates were taken as the output, the rate of including the correct class in the output was 98.6% which was rather high.

The ten candidates at the output from the preclassifier were examined. For each candidate, the preclassifier gave a value proportional

TABLE III
MATCHING LIST FOR THE TEMPLATE AND INPUT
IMAGES SHOWN IN FIG. 4(a) AND (b), RESPECTIVELY

Template element	Input element (s)
1	1 , 2
2	1
3	7
4	3 , 4
5	5
6	6
7	8
8	9
9	9
A	A
B	S
C	T
D	S
E	B , N
F	O
G	U
H	P
I	Q
J	R
K	R
L	L
M	D
N	E
O	F
P	K
Q	L , M
R	G
S	H
T	I
U	G , J
V	C

TABLE IV
RESULTS OF EXPERIMENTS ON HANDWRITTEN CHINESE CHARACTER RECOGNITION

No. of training samples	7680
No. of test samples	960
No. of correctly classified samples	923
No. of incorrectly classified samples	37
Recognition rate	96.15%
Error rate	3.85%

to the logarithm of the probability of the unknown input belonging to the candidate class. If the confidence level of the best candidate was sufficiently high, it was taken as the recognized class. The decision was based on the difference in the output of the preclassifier between the best and the second best candidates. A threshold value of 20 was adopted. If the threshold was exceeded, the best candidate was taken as the recognized class. Otherwise the iterative matching algorithm proposed in this article was called to evaluate the matching scores for each of the ten candidates. With the threshold value of 20, about 85% of the cases required iterative matching. If a larger threshold was adopted, the confidence level would increase. But more calls to the iterative matching algorithm would take place and thus a longer time would be needed for recognition.

Only one template (reference) character was used for each class for matching with the unknown input. In principle, it would be desirable to have several templates per class, with one template to take care of each distinctive style of writing. However, due to the complexity of the characters, the variety of styles is great. Moreover, the time needed for matching would be increased by N times if N templates per class were used. With these considerations, only one template per class was adopted.

As explained in the Theory Section, the four scores from iterative matching and the 1 score from the preclassifier were linearly combined according to (5) to produce one final score for each candidate. The recognition decision was to select the candidate class with the highest final score. The recognition rate of the final stage was 97.5%. The overall recognition rate was the product of this with the preclassification rate of 98.6%. This gave the overall rate of 96.15%. A summary of the results is given in Table IV. The results could be considered satisfactory. Hence the incorporation of the iterative stage has raised the recognition rate from 86.6 to 96.1%.

IV. DISCUSSION AND CONCLUSION

In this paper, a top-down elastic approach to pattern matching and its application to complex handwritten Chinese character recognition is discussed. The matching algorithm works satisfactorily. The elements are not attracted to their nearest neighbors, but are moved in order to preserve global structural relations. This results from the multi-scale strategy adopted. A large neighborhood is used in the beginning so that large-scale features are coarsely aligned. As the neighborhood size shrinks in successive iterations, finer and finer details are aligned also. Another mechanism worth mentioning is the normalization of the weights u_{ij} and v_{ij} in (2) and (4) which gives each input element the same total attractive force on the set of template elements and vice versa. Hence an isolated element (remote from others) is still powerful enough to attract some elements for matching.

The method has been tried on the recognition of handwritten Chinese characters. Since the matching operation is based on Euclidean distances while characters are recognized according to topological or relational features, the original energy function cannot be used as a measure for recognition. A set of measures for the "goodness of match" is proposed which mainly concerns topological features. Results of experiments are satisfactory. The use of iterative matching with the proposed measures improves the recognition rate of a statistical classifier from 86.6 to 96.1%.

The computation complexity of the present method is proportional to N^2 , where N is the total number of template and input elements. This is determined from the updating equations for the coordinates of the elements [(2) and (4)]. This compares favorably with other methods, such as relaxation labeling [1]–[4] which has a computation complexity proportional to N^4 . Hence for complex patterns involving a large number of elements, the present method is more efficient. With respect to recognition accuracy, experimental results show that the two methods are about the same [2], [3].

REFERENCES

- [1] S. Ranade and A. Rosenfeld, "Point pattern matching by relaxation," *Pattern Recognit.*, vol. 12, pp. 269–275, 1980.
- [2] K. Yamamoto and A. Rosenfeld, "Recognition of hand-printed Kanji characters by a relaxation method," in *Proc. 6th Int. Conf. Pattern Recognition*, 1982, pp. 395–398.
- [3] C. H. Leung, Y. S. Cheung, and Y. L. Wong, "A knowledge-based stroke-matching method for Chinese character recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-17, pp. 993–1003, 1987.
- [4] L. Lam and C. Y. Suen, "Structural classification and relaxation matching of totally unconstrained handwritten zip-code numbers," *Pattern Recognit.*, vol. 21, pp. 19–31, 1988.
- [5] K. C. You and K. S. Fu, "A syntactic approach to shape recognition using attributed grammars," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, pp. 334–345, 1979.
- [6] Q. Y. Shi and K. S. Fu, "Parsing and translation of (attributed) expansive graph languages for scene analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 472–485, 1983.
- [7] R. K. Moore, "A dynamic programming algorithm for the distance between two finite areas," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 86–88, 1979.
- [8] D. J. Burr, "Elastic matching of line drawings," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 708–713, 1981.
- [9] M. Moshfeghi, "Elastic matching of multimodality medical images," *CVGIP: Graph. Models Image Process.*, vol. 53, pp. 271–282, 1991.
- [10] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, pp. 119–130, 1988.
- [11] P. N. Suganthan, E. K. Teoh, and D. P. Mital, "Pattern recognition by homomorphic graph matching using Hopfield neural networks," *Image Vision Comput.*, vol. 13, pp. 45–60, 1995.
- [12] P. Morasso, L. Barberis, S. Pagliano, and D. Vergamp, "Recognition experiments of cursive dynamic handwriting with self-organizing networks," *Pattern Recognit.*, vol. 26, pp. 451–460, 1993.
- [13] L. P. Cordella, C. De Stefano, and M. Vento, "A neural network classifier for OCR using structural descriptions," *Mach. Vision Applicat.*, vol. 8, pp. 336–342, 1995.
- [14] R. Bajcsy and S. Kovacic, "Multiresolution elastic matching," *Comput. Vision, Graph. Image Process.*, vol. 46, pp. 1–21, 1989.
- [15] C. H. Leung, "Pattern matching by energy minimization," in *Proc. 7th Scandinavian Conf. Image Analysis*, Aalborg, Denmark, Aug. 13–16, 1991, pp. 370–377.
- [16] I. Cohen, N. Ayache, and P. Sulger, "Tracking points on deformable objects using curvature information," *Proc. ECCV '92, 2nd Eur. Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992, pp. 458–466.
- [17] T. Ozanian, "Approaches for stereo matching," *Model., Ident., Contr.*, vol. 16, pp. 65–94, 1995.
- [18] M. Revow, C. K. I. Williams, and G. E. Hinton, "Using mixtures of deformable models to capture variations in hand printed digits," in *Proc. 3rd Int. Workshop Frontiers Handwriting Recognition*, Buffalo, NY, May 25–27, 1993, pp. 142–152.
- [19] T. Wakahara, "Shape matching using LAT and its application to handwritten numeral recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 618–629, 1994.
- [20] R. Durbin and D. Willshaw, "An analogue approach to the traveling salesman problem using an elastic net method," *Nature*, vol. 326, pp. 689–691, 1987.
- [21] C. Y. Suen and C. Shiau, "An iterative technique of selecting an optimal 5×7 matrix character set for display in computer output systems," in *Proc. Soc. Information Display*, 1980, vol. 21, pp. 9–15.
- [22] R. Plamondon, C. Y. Suen, M. Bourdeau, and C. Barriere, "Methodologies for evaluating thinning algorithms for character recognition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, pp. 1247–1270, 1993.
- [23] Y. S. Huang, K. Liu, and C. Y. Suen, "The combination of multiple classifiers by a neural network approach," *J. Pattern Recognit. Artif. Intell.*, vol. 9, pp. 579–597, 1995.
- [24] Y. S. Cheung and C. H. Leung, "Chain-code transform for Chinese character recognition," in *Proc. IEEE Int. Conf. Systems, Man, Cybernetics*, Tucson, AZ, 1985, pp. 42–45.
- [25] C. H. Leung, Y. S. Cheung, and K. P. Chan, "A distortion model for Chinese character generation," in *Proc. IEEE Int. Conf. Systems, Man, Cybernetics*, Tucson, AZ, 1985, pp. 38–41.