

RECOGNIZING HANDWRITTEN CHINESE CHARACTERS BY STROKE-SEGMENT MATCHING USING AN ITERATION SCHEME

SHENG-LIN CHOU* and WEN-HSIANG TSAI†

**Institute of Computer Science and Information Engineering*

†Department of Computer and Information Science

National Chiao Tung University

Hsinchu, Taiwan 30050, R.O.C.

The problem of handwritten Chinese character recognition is solved by matching character stroke segments using an iteration scheme. Length and orientation similarity properties, and coordinate overlapping ratios are used to define a measure of similarity between any two stroke segments. The initial measures of similarity between the stroke segments of the input and template characters are used to set up a match network which includes all the match relationships between the input and template stroke segments. Based on the concept of at-most-one to one mapping, an iteration scheme is employed to adjust the match relationships, using the contextual information implicitly contained in the match network, so that the match relationships can get into a stable state. From the final match relationships, matched stroke-segment pairs are determined by a mutually-best match strategy and the degree of similarity between the input and each template character is evaluated accordingly. Certain structure information of Chinese characters is also used in the evaluation process. The experimental results show that the proposed approach is effective. For recognition of Chinese characters written by a specific person, the recognition rate is about 96%. If the characters of the first three ranks are checked in counting the recognition rate, the rate rises to 99.6%.

Keywords: Chinese character recognition; Handwritten Chinese characters; Stroke segment; Stable matching; Match network; Iteration; Similarity measure.

1. INTRODUCTION

Machine recognition of handwritten Chinese characters (HCC) has been becoming more and more important recently, as the volume of information in daily life increases rapidly. It is more effective to input Chinese characters by recognition than by keyboards. Unfortunately, machine recognition of handwritten Chinese characters is a very difficult problem. The difficulty comes from several aspects, and the most important one may be the very large vocabulary set of Chinese characters. There are more than 40 000 Chinese characters and about a tenth of them are used frequently in daily life.¹ The complex structures of many Chinese characters form another barrier. To conquer these problems, some researchers tried to segment Chinese characters into smaller primitives, called radicals, and recognize them accordingly. Each Chinese

* Sheng-Lin Chou is also with Advanced Technology Center, Computer and Communication Research Laboratories, Industrial Technology Research Institute, Chutung, Hsinchu, Taiwan 31015, R.O.C.

† To whom all correspondence should be sent.

character is made up of a few of the 350 distinct radicals available. Although it is argued that the recognition problem can be simplified if recognition is done according to the radicals,^{2,3} segmentation of Chinese characters into radicals is not easy because of the characteristics of the high levels of connectedness and cursiveness of handwritten Chinese characters. For handwritten character recognition, great variations in writing style form the third major source of recognition difficulty. Variations exist not only among characters written by different writers but also among the same characters written at different times by one person. Normalization is a way to compensate for certain variations; however, its effect is usually limited.

Another way to recognize Chinese characters is to segment each Chinese character into certain line or curve segments, called strokes, and solve the problem by using the features extracted from strokes. A survey of some works on the recognition of handwritten Chinese characters using strokes can be found in Ref. 4. One way to solve the problem based on the use of the stroke is to match the strokes of an input character with those of each template character in a database and then take the best match as the recognized one. Relaxation is a well-known matching technique which has been successfully used in many applications.^{5,6} It has also been employed for the recognition of Chinese characters.⁷⁻¹² Yamamoto and Rosenfeld first applied the relaxation technique to the HCC recognition problem⁷ and Yamamoto *et al.* improved the work later.^{8,9} In their work, Chinese characters were approximated by polygons and the line segments of the polygon of an input character were matched with those of the template characters by the relaxation method. Good recognition results were obtained. Another application of relaxation to HCC recognition was reported by Xie and Suk.¹⁰ Distinctive structure information, such as hook, T-shape, and cross, representing the structural information of characters was used as the major features, and matching between the features of input characters and the templates was obtained by relaxation. The technique was shown to be powerful in distinguishing similarly shaped characters within a cluster produced by preclassification. Because relaxation is a time-consuming process, Leung, Cheung, and Wang¹¹ proposed an idea to incorporate specific knowledge such as the positional relations between the strokes of a character into a training system so that the relaxation matching process can be greatly speeded up. Rather than using the relaxation method to match input strokes with those of templates, Cheng, Hsu, and Chen¹³ transformed the matching problem into a stroke assignment problem to find the optimal combination of matches between the input and template strokes. The location and orientation similarity values were used to define a measure of similarity between two strokes, and a function of fuzzy entropy was used in the measure. By slightly modifying the problem, such as adding some dummy strokes, the Hungarian method was then applied to solve the stroke assignment problem.

In this paper, we try to solve the HCC recognition problem by matching input and template characters using an iteration scheme proposed by Chou and Tsai.¹⁴ A character to be recognized is first thinned to be of one pixel width. The result is then traced and approximated by line segments. Such line segments, when connected and

smoothed appropriately, form the strokes of the characters, but they are not the strokes themselves. They will be called stroke segments here. After the stroke segments are extracted, the character is normalized to be of a fixed size. Length and orientation similarity properties, and the overlapping ratios in the x and y directions, of every two stroke segments are used to define a measure of similarity between the two stroke segments. The similarity measures of all possible matched stroke-segment pairs computed accordingly are used to set up a match network, which is then updated iteratively by Chou and Tsai's iteration scheme under the premise of at-most-one to one mapping. Each stroke segment competes with some other stroke segments during the iterations in order to get its best match in the final result. Through the iterations, the inherent ambiguity in the initial stroke-segment similarity measures can be greatly reduced and the match relationships between the input and template stroke segments can reach a more stable state. Final matched stroke-segment pairs are then determined from the stable match relationships according to a mutually-best match strategy. After the stroke-segment matching process, the degree of similarity between the input and each template character is evaluated according to the similarity measures of all the matched stroke-segment pairs as well as the structure information contained in the composing stroke segments of the characters. Decision making for character recognition is finally made.

The remainder of this paper is organized as follows. Section 2 includes the definition of the similarity measure for a stroke-segment pair. Chou and Tsai's iteration scheme is reviewed in Sec. 3, and the degree of similarity between two characters is defined in Sec. 4. Section 5 includes the experimental results and Sec. 6 gives some concluding remarks.

2. SIMILARITY MEASURE FOR STROKE-SEGMENT MATCHING

2.1. Length and Orientation Similarity Measures

To measure the degree of similarity between two stroke segments, three types of measures based on the length similarity, the orientation similarity, and the coordinate overlapping ratio are defined. For stroke segments l_1 and l_2 , the length similarity, denoted as $len(l_1, l_2)$, is defined to be

$$len(l_1, l_2) = \frac{shorter(l_1, l_2)}{longer(l_1, l_2)}, \quad (1)$$

where $shorter(l_1, l_2)$ and $longer(l_1, l_2)$ represent the shorter and longer lengths of l_1 and l_2 respectively.

To define the orientation similarity, the angle difference between two stroke segments is utilized. Because the position of a line segment may shift along the direction of the line containing the stroke segment due to writing variations, the angle difference between two stroke segments l_1 and l_2 , denoted by θ_{12} , is taken to be the

smaller angle made by the lines containing l_1 and l_2 . For example, the angle difference between the stroke segments l_1 and l_2 shown in Fig. 1 is taken to be α rather than β because there is no way to know where the correct positions of l_1 and l_2 are. The orientation similarity between l_1 and l_2 , denoted as $ori(l_1, l_2)$, is then defined to be

$$ori(l_1, l_2) = \begin{cases} \cos(k\theta_{12}) & \text{if } k\theta_{12} < \pi/2, \\ -\infty & \text{otherwise,} \end{cases} \quad (2)$$

where $k = 2$ if $longer(l_1, l_2) < 75$ and $k = 3$ for other cases, assuming that all characters are normalized to be of the size of 150×150 pixels. It was pointed out⁹ that the direction of short stroke segments is not stable, so the tolerance of the angle difference between two short stroke segments should be set larger. In Eq. (2), the tolerance of the angle difference between two stroke segments is set to 45° when both of the lengths of the two stroke segments are less than 75 pixels, and to 30° for other cases. When the angle difference between two stroke segments exceeds the tolerance, the possibility for the two stroke segments to form a match pair is pruned by setting the similarity measure value to $-\infty$.

2.2. Coordinate Overlapping Ratio

The need of measuring coordinate overlapping ratios for stroke segments originates from the requirement that the positions of the same stroke segment in the written variants of a character should not differ greatly. This requirement of location invariance can be reflected by the measurement of the coordinate overlapping ratios in the x and y directions. The x -overlapping ratio of two stroke segments is defined as the overlapping length of the two stroke segments over the sum of the lengths of the two stroke segments, after they are projected onto the x -axis. The y -overlapping ratio is defined similarly. However, the overlapping length in the x -axis (or y -axis) becomes zero when one or both of the two stroke segments become vertical (or horizontal). In this case the measure of the overlapping ratio becomes meaningless. In addition, when

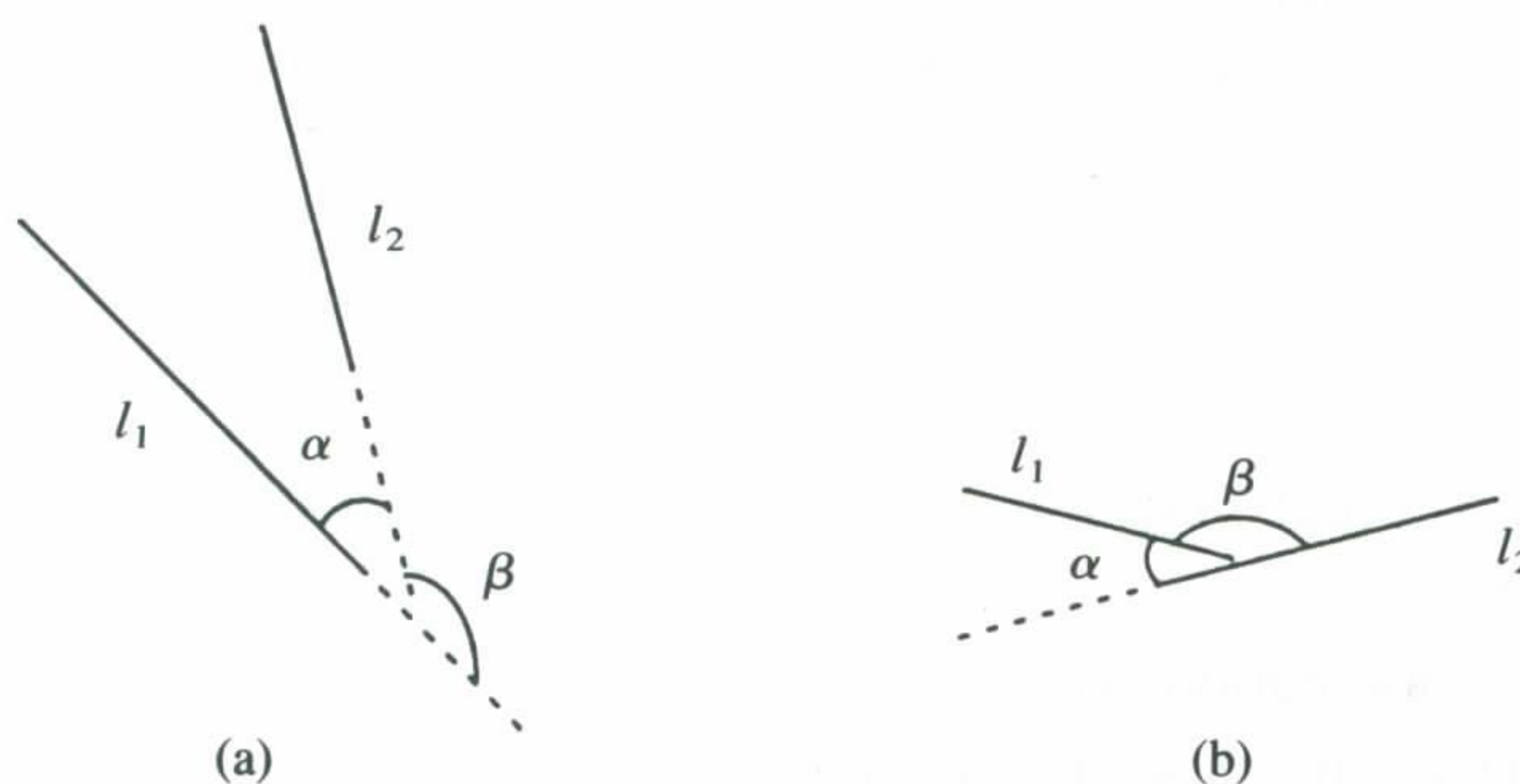


Fig. 1. Angle difference of two stroke segments.

both of the stroke segments are too short, a small variation in the stroke-segment position may induce a great change in the values of the x - and y -overlapping ratios. To handle such cases, a rectangular variation window is defined for use in computing the value of the coordinate overlapping ratio. The variation window of a stroke segment is defined as the smallest rectangle circumscribing the stroke segment if the length of each side of the resulting window is no less than a constant w . If the length of either side of the circumscribing rectangle is smaller than w , then it is equally extended along both directions of the line containing the side until the length of the extended side reaches w . The minimum side length for the variation window w is set to 40 for the character size of 150×150 pixels. Figure 2 shows the variation windows of three different stroke segments. Suppose that the variation window of stroke segment l_i is specified by its two diagonal corner points, $(\bar{x}_{i1}, \bar{y}_{i1})$ and $(\bar{x}_{i2}, \bar{y}_{i2})$, where $\bar{x}_{i2} > \bar{x}_{i1}$ and $\bar{y}_{i2} > \bar{y}_{i1}$. Then the x - and y -overlapping ratios can be defined in more detail as

$$x\text{-overlap} = \max \left(\frac{2[\min(\bar{x}_{12}, \bar{x}_{22}) - \max(\bar{x}_{11}, \bar{x}_{21})]}{(\bar{x}_{12} - \bar{x}_{11}) + (\bar{x}_{22} - \bar{x}_{21})}, 0 \right) \quad (3)$$

and

$$y\text{-overlap} = \max \left(\frac{2[\min(\bar{y}_{12}, \bar{y}_{22}) - \max(\bar{y}_{11}, \bar{y}_{21})]}{(\bar{y}_{12} - \bar{y}_{11}) + (\bar{y}_{21} - \bar{y}_{21})}, 0 \right) \quad (4)$$

respectively, and the (overall) coordinate overlapping ratio for stroke segments l_1 and l_2 , denoted by $ovlap(l_1, l_2)$, is defined to be

$$ovlap(l_1, l_2) = \sqrt{x\text{-overlap} \cdot y\text{-overlap}}. \quad (5)$$

One reason for defining the coordinate overlapping ratio as above is to reflect the fact that two stroke segments will in general overlap more in both directions if they are stroke segments of two variants of a character.

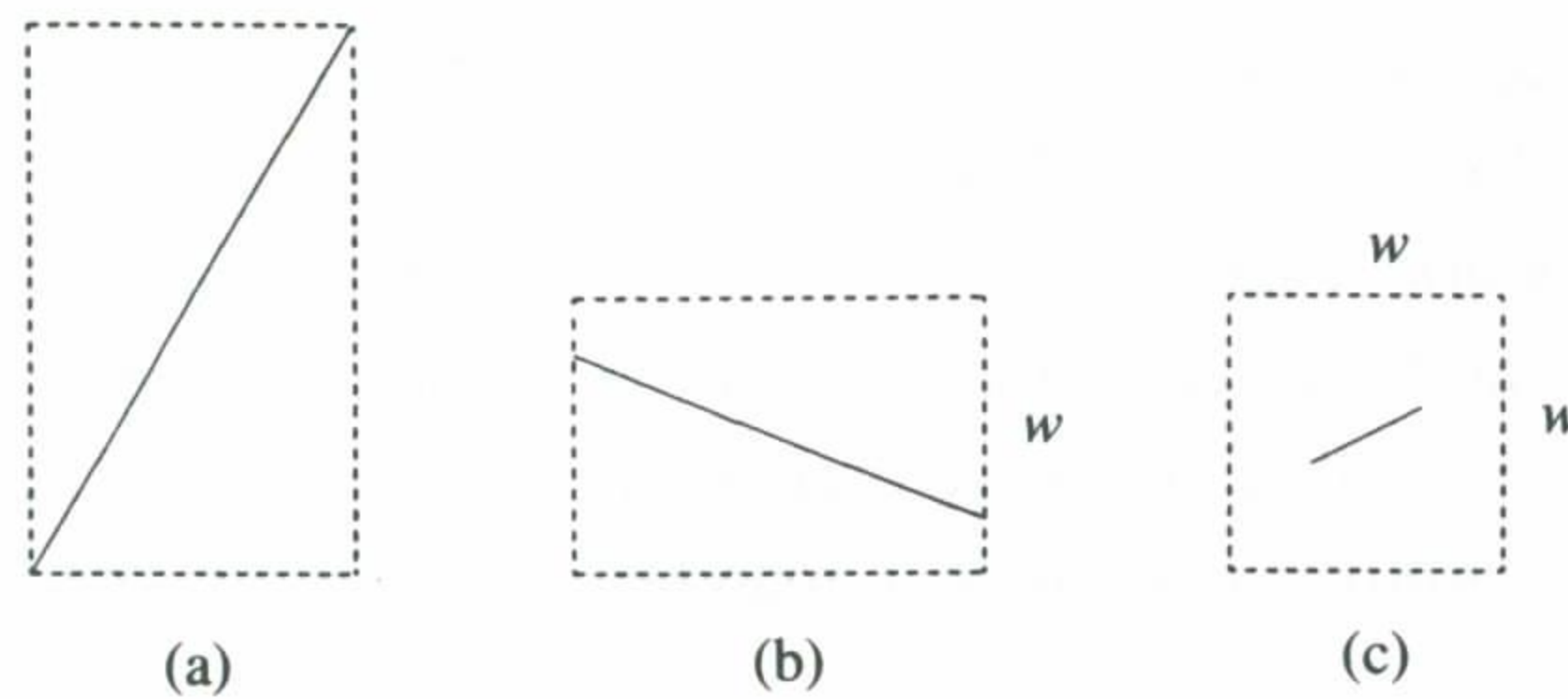


Fig. 2. Variation windows of three different stroke segments. The vertical side and both sides of the rectangles circumscribing the stroke segments in (b) and (c), respectively, are less than w and so are extended to be of the length of w .

2.3. Stroke-Segment Similarity Measure

The similarity measure for a stroke-segment pair (l_1, l_2) , denoted by $P(l_1, l_2)$, is defined by combining the measures of *len*, *ori*, and *ovlap* as follows:

$$P(l_1, l_2) = [a \cdot \text{len}(l_1, l_2) + (1 - a) \text{ori}(l_1, l_2)] \text{ovlap}(l_1, l_2), \quad (6)$$

where a is a constant for adjusting the weights of the measures *len* and *ori*. Because the values of *ori* and *ovlap* may be less than or equal to zero, the value of $P(l_1, l_2)$ may be non-positive, too. When $P(l_1, l_2)$ is less than or equal to zero, (l_1, l_2) is treated as an impossible match pair and is pruned at the beginning of matching. Note that the measure of *ovlap* is included in the stroke-segment similarity measure P by multiplication rather than addition. One reason is that if the locations of two stroke segments differ greatly, then the measures *len* and *ori* become meaningless; and the more the variation windows of the two stroke segments overlap, the more effective the measures *len* and *ori* are. For example, when two vertical stroke segments are identical in length and orientation but are located at the left and right boundaries of two characters respectively, the similarity measure of these two stroke segments should be zero; however, if the similarity is defined by adding up *len*, *ori*, and *ovlap* with weighting factors a , b , and $(1 - a - b)$ respectively, then the resulting measure will be $(a + b)$ rather than 0, which obviously is undesirable.

2.4. Match Network

A graph, called a match network, is used to represent the match relationships between input and template stroke segments. The match network is bipartite and each node on the left or right sides of the network represents a stroke segment of the input and template respectively. All possible match pairs of input and template stroke segments are connected by links, and each link is associated with a value, called match link strength, to reflect the degree of the match between the stroke segments. The strength of the link connecting nodes i and j is denoted by P_{ij} and is always kept positive. If the value of P_{ij} becomes non-positive in the iteration process, stroke-segment pair (i, j) is treated as an impossible match pair, and the link connecting them is removed from the graph. The initial link strengths of the match network are calculated from Eq. (6) by pruning the non-positive measures. They are then iteratively updated by Chou and Tsai's iteration scheme.

In the match network, each node connected to a certain node, say node i , is called a possible match of i and the set containing all the possible matches of node i is denoted as $PM(i)$. Node j is called the best match of i if $j \in PM(i)$ and $P_{ij} \geq P_{ik}$ for all $k \in PM(i)$. The possibility that node i matches node j , denoted by $\text{poss}(i \rightarrow j)$, is defined to be

$$\text{poss}(i \rightarrow j) = \frac{P_{ij}}{\sum_{k \in PM(i)} P_{ik}}. \quad (7)$$

Note that $poss(i \rightarrow j)$ is not necessarily equal to $poss(j \rightarrow i)$. Because the stroke segments in input and template characters are treated identically, the possibility that nodes i and j form a match pair, denoted by $poss(i \leftrightarrow j)$, is defined by combining the values of $poss(i \rightarrow j)$ and $poss(j \rightarrow i)$ in the following way:

$$poss(i \leftrightarrow j) = poss(i \rightarrow j) \cdot poss(j \rightarrow i) . \quad (8)$$

A pair of stroke segments, (i, j) , is called a mutually-best match pair if the following conditions are satisfied¹⁵:

$$\begin{aligned} poss(i \rightarrow j) &= \max_{k \in PM(i)} poss(i \rightarrow k) , \\ poss(j \rightarrow i) &= \max_{k \in PM(j)} poss(j \rightarrow k) . \end{aligned} \quad (9)$$

This mutually-best match strategy is used to determine the desired matches of stroke segments in Chou and Tsai's iteration scheme reviewed next.

3. ITERATION SCHEME

After the similarity measures are calculated from Eq. (6), we can match input stroke segments with those of each template and evaluate the degree of similarity between the input character and each template accordingly. The simplest way to get desired matches between input and template stroke segments is to make decisions *directly* from the stroke-segment similarity measures. In general, it is difficult to get good matching results this way because a very good similarity measure is hard to define. An improvement is to employ an iteration scheme to refine the stroke-segment similarity measures using contextual information support. By iteratively updating the measures, global consistency, in general, can be achieved and better matching results obtained. In this section, we apply Chou and Tsai's iteration scheme to stroke-segment matching using the stroke-segment similarity measures calculated from Eq. (6). Chou and Tsai's iteration scheme was proposed in Ref. 14 to solve the stereo line segment matching problem.

3.1. Basic Concept

In Chou and Tsai's iteration scheme, the match relationship between an input and template stroke-segment pair must be a one-to-one or none-to-one mapping. For convenience, this type of match relationship will be called at-most-one to one mapping subsequently. Since the input and template stroke segments are treated symmetrically, at-most-one to one mappings can also be interpreted as one to at-most-one mappings. For the stroke segments of two variants of a character, the match relationship between them is usually an at-most-one to one mapping although this cannot be guaranteed. As an illustration, Fig. 3 shows the stroke-segment correspondences in two writing variants of the Chinese character 'chou' (pronunciation), where stroke-segment u maps

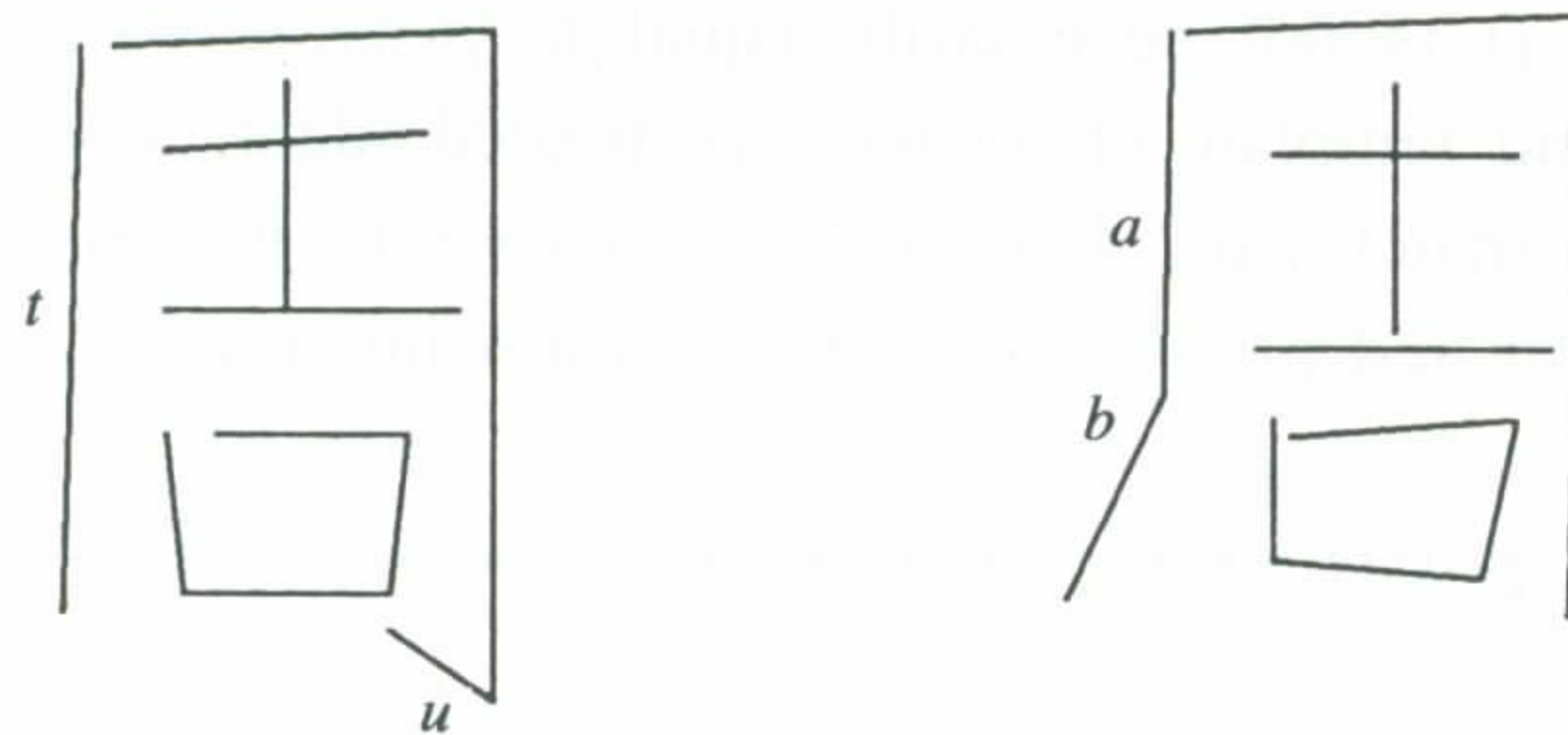


Fig. 3. Stroke variations of the Chinese character 'chou' (pronunciation) in different writings. Stroke segments a and b correspond to t , stroke segment u maps to nothing, and all other stroke segments satisfy one-to-one mappings.

to nothing, stroke segments a and b correspond to an identical stroke segment t , and the remaining stroke segments satisfy one-to-one mappings. To handle the one-to-many mapping, such that t can map to both a and b in Fig. 3, it is treated as a combination of a one-to-one and several none-to-one mappings in the iteration scheme; and after the iterations stop, the one-to-many mapping is recovered by postprocessing.

Under the premise of at-most-one to one mapping, it is natural for each stroke segment to correspond to its best match. However, since the similarity measure may not be well defined and the writings usually vary so much, matching ambiguity will always exist in the corresponding match network. Nevertheless, the contextual information contained implicitly in the measures is usually helpful in reducing the ambiguity contained explicitly in the measures. In Chou and Tsai's iteration scheme, such contextual information can be utilized to reduce matching ambiguity. In the iteration process, each stroke segment competes with others for its best match. When iterations go on, the ambiguity among the measures is reduced gradually and the match network becomes more stable. As the match network reaches a stable state finally, matched stroke-segment pairs can be determined accordingly.

3.2. Illustrative Example

Consider the stroke segments shown in Fig. 4(a). Suppose that l_1 , l_2 , and l_3 are three stroke segments of a character and that l'_1 , l'_2 , and l'_3 are the corresponding stroke segments of a variant of the character. Suppose that after the stroke-segment similarity measures are computed by Eq. (6), the corresponding match network looks like that shown in Fig. 4(b). The values associated with the links in Fig. 4(b) are assumed to be the computed stroke-segment similarity measures. The darkness of the links in the match network reflects the magnitudes of the match link strengths, and an arrow pointing from node a to node b indicates that the best match of a is b . The figure shows that l_1 and l'_1 are a match pair, denoted by $(l_1 \leftrightarrow l'_1)$, under the mutually-best match strategy. Stroke segments l_2 and l'_3 are also a mutually-best match pair initially. Match $(l_1 \leftrightarrow l'_1)$ is stable because it is a mutually-best match pair and there is no other stroke segment to interfere with it, but match $(l_2 \leftrightarrow l'_3)$ is not stable

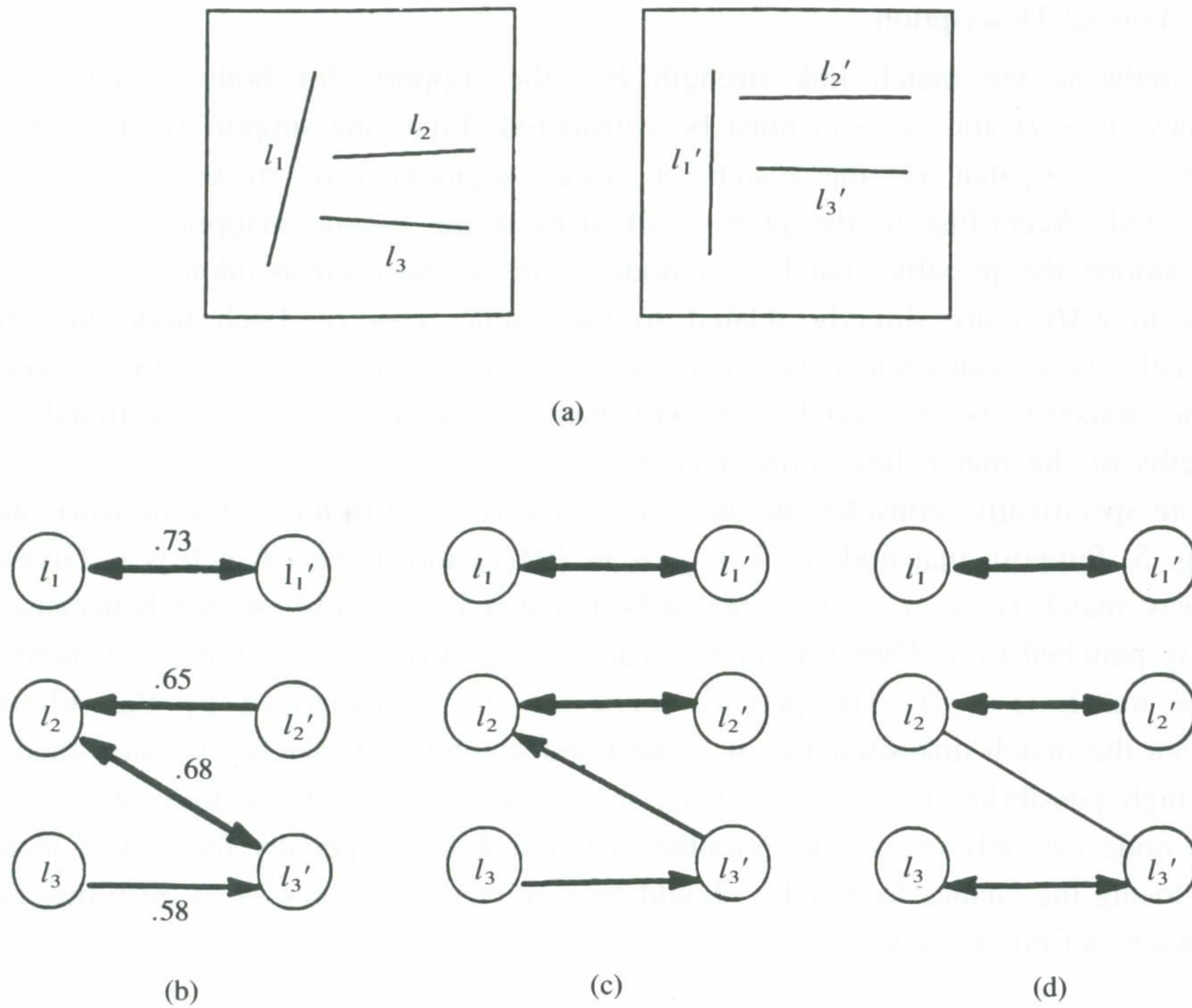


Fig. 4. An illustration of the steps of Chou and Tsai's iteration scheme for stroke-segment matching. (a) Input and template stroke segments. (b) Initial match network. (c) An intermediate step of Chou and Tsai's iteration process. (d) Final matching results after the iterations.

because it is interfered with by stroke segments l_2' and l_3 . As l_2 is the only possible match of l_2' , and it can be matched by only one stroke segment under the premise of at-most-one to one mapping, l_2 will surely compete with l_3' for the chance to match l_2 . To compare the competence, the difference between the best and the second-best matches of the competitors is checked.¹⁴ Because the difference between the strengths of the best and the second-best matches of l_3' is much less than that of l_2' , the competence of l_2 is much stronger than that of l_3' . This means that it makes little difference for l_3' to match with l_2 or l_3 , while l_2 will lose more when it loses the match with l_2 . A similar case occurs for stroke segments l_3 and l_2 in competing for the correspondence to l_3' . Therefore the match link strengths of (l_2, l_2') and (l_3, l_3') will increase by a larger amount than that of (l_2, l_3') as iterations go on. After some iterations, the match network will appear as the one shown in Fig. 4(c), where the best match of l_2 is changed to l_2' and (l_2, l_2') becomes a mutually-best match pair. Figure 4(d) shows a possible result of the match network after some more iterations, in which another match pair $(l_3 \leftrightarrow l_3')$ is obtained. In summary, through Chou and Tsai's iteration scheme, the originally incorrect match $(l_2 \leftrightarrow l_3')$ will be removed and two new correct matches, $(l_2 \leftrightarrow l_2')$ and $(l_3 \leftrightarrow l_3')$, will be finally obtained.

3.3. Formal Description

In updating the match link strength P_{ij} , the supports for both of the one-way matches $(i \rightarrow j)$ and $(j \rightarrow i)$ must be considered. First, the support for the one-way match $(i \rightarrow j)$, that is, the match of stroke segment i to stroke segment j , is considered. According to the premise of at-most-one to one mapping, at most one node among the possible matches of node i can be the correspondence of i . So, all nodes in $PM(i)$ are directly related to the event $(i \rightarrow j)$. Each node in $PM(i)$, especially those nodes which take i as the best match, will compete with one another for the chance to be the match of i , with their competence values proportional to the strengths of the match links connected to i .

More specifically, consider the one-way match $(i \rightarrow j)$ in the match network shown in Fig. 5. Suppose that node k , $k \neq j$, is in $PM(i)$ and its best match is i . Given the one-way match $(i \rightarrow j)$, node k can at best match its second-best match because i is already matched to j . Therefore node k can be regarded as encountering a penalty due to the match $(i \rightarrow j)$. The penalty for k can be characterized by the difference between the match link strengths of its best and second-best matches because there is a very high possibility for k to match its second-best match after it loses its best one. Since node i is only one of the possible matches of k , the penalty for node k incurred from losing the chance to match i should be weighted by $poss(k \rightarrow i)$, resulting in the following definition of the penalty:

$$\left[P_{ik} - \max_{h \neq i} (P_{hk}) \right] poss(k \rightarrow i) . \quad (10)$$

On the other hand, if the best match of node k is not i , then the possibility for node k

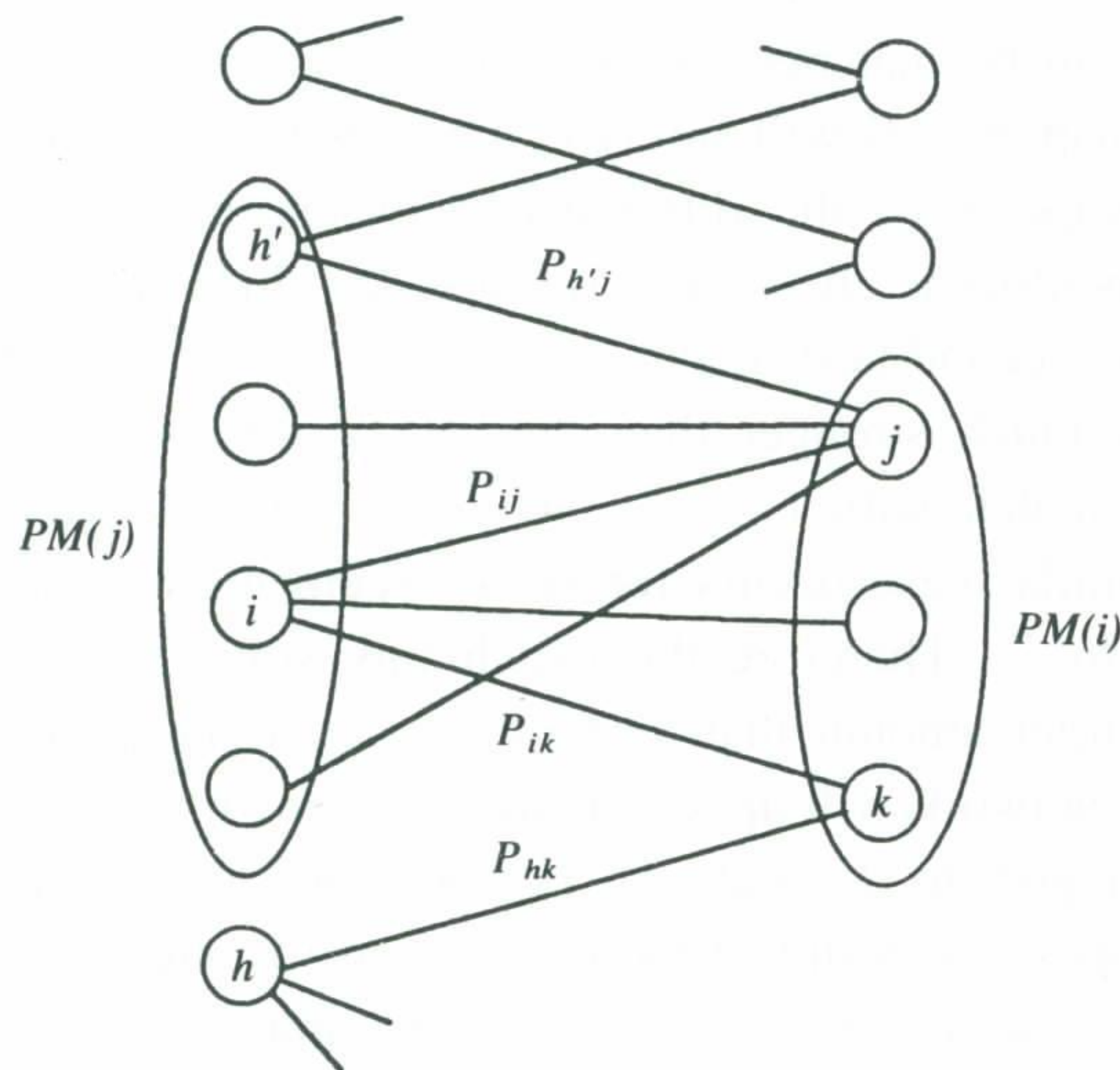


Fig. 5. An example of a match network for computing the support for the one-way match $(i \rightarrow j)$.

to match its best match increases relatively under the event $(i \rightarrow j)$ because the number of competitors of k decreases by one. Therefore node k gets a benefit from the match $(i \rightarrow j)$. The benefit can be defined similarly to the penalty (except for the sign) as

$$\left[\max_{h \neq i} (P_{hk}) - P_{ik} \right] \text{poss}(k \rightarrow i) . \quad (11)$$

The benefit or penalty of k for $(i \rightarrow j)$ can be seen as a positive or negative support of node k for the one-way match $(i \rightarrow j)$. Because each distinct k in $PM(i)$ owns a different match link strength P_{ki} , the supports obtained from different nodes must be weighted proportionally to the strengths of the match links connected to i from the viewpoint of i . The weighting factor is exactly the value of $\text{poss}(i \rightarrow k)$. Therefore the support of node k for the match $(i \rightarrow j)$ can be defined as

$$\begin{aligned} & \left[\max_{h \neq i} (P_{hk}) - P_{ik} \right] \text{poss}(k \rightarrow i) \text{poss}(i \rightarrow k) \\ & = \left[\max_{h \neq i} (P_{hk}) - P_{ik} \right] \text{poss}(i \leftrightarrow k) . \end{aligned} \quad (12)$$

For match $(i \rightarrow j)$, nodes j and k are in exactly the opposite positions if $j, k \in PM(i)$ and $j \neq k$. In other words, for the one-way match $(i \rightarrow j)$, the penalty and benefit of node k become the benefit and penalty of node j respectively.

By summing up the supports of all nodes in $PM(i)$, the (overall) support for the one-way match $(i \rightarrow j)$, denoted by $\text{supt}(i \rightarrow j)$, is defined as

$$\begin{aligned} \text{supt}(i \rightarrow j) &= \left[P_{ij} - \max_{h' \neq i} (P_{h'j}) \right] \text{poss}(i \leftrightarrow j) \\ &+ \sum_{\substack{k \in PM(i) \\ k \neq j}} \left[\max_{h \neq i} (P_{hk}) - P_{ik} \right] \text{poss}(i \leftrightarrow k) . \end{aligned} \quad (13)$$

The first term in Eq. (13) is the support of node j and the second term includes the supports from all other nodes in $PM(i)$ for $(i \rightarrow j)$. Equation (13) considers only the support resulting from the one-way match $(i \rightarrow j)$. To get the support for the two-way match $(i \leftrightarrow j)$, the support for the other-way match $(j \rightarrow i)$ must also be calculated. By symmetry, $\text{supt}(j \rightarrow i)$ can be defined similarly. The support for $(i \leftrightarrow j)$, denoted by $\text{supt}(i \leftrightarrow j)$, can now be defined by combining the supports $\text{supt}(i \rightarrow j)$ and $\text{supt}(j \rightarrow i)$ as

$$\text{supt}(i \leftrightarrow j) = [\text{supt}(i \rightarrow j) + \text{supt}(j \rightarrow i)]/2 . \quad (14)$$

With the support function defined as above, we are ready to define the procedure to update the match link strengths in the match network. Suppose that P_{ij}^t denotes the strength of the match link connecting nodes i and j in the match network after t iterations. The updating formula for P_{ij}^{t+1} is defined to be

$$P_{ij}^{t+1} = (1 - \beta)(P_{ij}^t + \beta) \text{supt}(i \leftrightarrow j), \quad (15)$$

where β , within the range $[0, 1]$, is a parameter used to adjust the weights of P_{ij}^t and support $\text{supt}(i \leftrightarrow j)$. If β is set close to 1, the similarity measure of a stroke-segment pair (i, j) will be determined mostly by the status of the nearby stroke segments included in $PM(i)$ and $PM(j)$. On the other hand, if β is set close to 0, then the effect of the neighboring support is limited, especially when the iteration process stops after only a few iterations.

3.4. Stop Criterion

The principle of the mutually-best match strategy defined by Eq. (9) is used to decide the stroke-segment matches from the match network during the iteration process. A match pair is determined as a correspondence if it becomes a mutually-best match pair. But a correspondence may be removed in later iterations if the condition of mutual best matching is no longer satisfied.

The iteration scheme stops when the number of mutually-best match pairs does not increase for a certain number of consecutive iterations. As with most iterative matching techniques, it is hard to prove the convergence of this iteration scheme. But our experiments show that the convergence is always obtained by the above stop criterion.

3.5. Postprocessing

Due to image processing or writing variations, certain one-to-many mappings may exist between the input and template stroke segments; however, only one stroke segment among the multiple ones corresponding to a particular stroke segment can get the final correspondence. For the remaining non-matched stroke segments, it is very possible that their final best matches are still correct. For example, as shown in Fig. 3, both stroke segments a and b correspond to stroke segment t but only a can get the match after the iterations because a is more similar to t than b . Stroke segment b may still take t as its best match after the iterations but will get no correspondence.

To overcome this problem, we can do the following additional processing after the correspondences are determined. For a non-matched stroke segment i , its best match, say j , is checked first to see whether it gets matched or not. If j gets no correspondence, nothing is done. If it does, stroke segment i is checked with the matched stroke segment of j , say i' , for the following two conditions:

connectedness: i and i' are connected within a distance threshold;

collinearity: the angle difference between i and i' is within a threshold.

(16)

If both conditions are satisfied, i and i' are considered to belong to an identical stroke segment which is broken due to image processing, and are merged to form a single stroke segment, which is then taken to be the final match of stroke segment j .

4. MEASURE OF SIMILARITY BETWEEN CHARACTERS

Consider the case in which we evaluate the similarity between an input character c_i and a template character c_t . Suppose that c_i and c_t are represented respectively by stroke segments l_i , $i = 1, 2, \dots, m$ and l_t , $t = 1, 2, \dots, n$, where m and n are the numbers of stroke segments of c_i and c_t respectively. Let L_i and L_t be the lengths of stroke segments l_i and l_t , and M be the set containing all the tuples (i, t) , each meaning that stroke segments l_i and l_t are a matched stroke-segment pair.

4.1. Simple Measurement

A measure of similarity between characters c_i and c_t , denoted by $S(c_i, c_t)$, can be defined as

$$S(c_i, c_t) = m(c_i, c_t) s(M), \quad (17)$$

where $m(c_i, c_t)$ is defined to be the total length of the matched stroke segment over the total length of all the stroke segments of c_i and c_t :

$$m(c_i, c_t) = \frac{\sum_{(i,t) \in M} (L_i + L_t)}{\sum_{i=1}^m L_i + \sum_{t=1}^n L_t}, \quad (18)$$

and $s(M)$ is defined as

$$s(M) = \frac{\sum_{(i,t) \in M} P(l_i, l_t)}{N(M)}, \quad (19)$$

where $N(M)$ is the size of set M and $P(l_i, l_t)$ is defined by Eq. (6).

The character similarity measure defined by Eq. (17) is simple and is not used in this study. It is refined by utilizing the structure information contained in Chinese characters.

4.2. Utilizing Character Structure Information

The structural relationships among the stroke segments which compose a character provide very useful information for recognizing Chinese characters. In Refs. 10, 11, 16, and 17, the character structure information is used as the main feature for recognizing handwritten Chinese characters. Three types of relationships between two stroke segments are examined in this study: cross, T-shape, and corner. Figure 6 shows their structures. They can be detected automatically. Although these features may not be stable in handwritten characters, especially the T-shape and the corner, they are still helpful for HCC recognition.

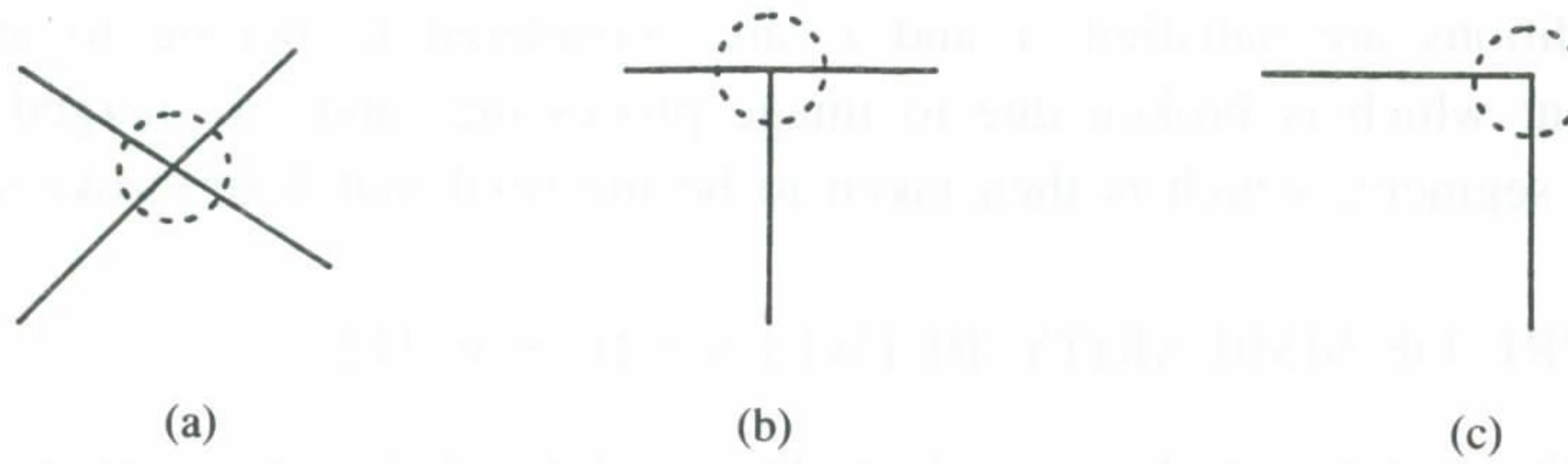


Fig. 6. Three types of structures between stroke segments. (a) Cross, (b) T-shape, and (c) corner.

To incorporate structure information into the character similarity measure, one way is to check if the structural relationships contained in the matched input stroke segments also exist in the corresponding template stroke segments. An input stroke segment may have structural relationships with several nearby stroke segments, and most of these relationships will be kept in the corresponding template stroke-segment pairs if the input and template characters belong to an identical class. Suppose that l_{i1} and l_{i2} are two stroke segments of the input character and they are matched to the template stroke segments l_{t1} and l_{t2} respectively. If the relationship between l_{i1} and l_{i2} is of one of the three types, cross, T-shape, or corner, and is different from that between l_{t1} and l_{t2} , then the match ($l_{i1} \leftrightarrow l_{t1}$) is said to be incompatible with ($l_{i2} \leftrightarrow l_{t2}$). If ($l_{i1} \leftrightarrow l_{t1}$) and ($l_{i2} \leftrightarrow l_{t2}$) are incompatible, then there is a high possibility that at least one match is erroneous. Since there is no way to know which match is incorrect, it is proposed that the penalty incurred by the incompatibility be distributed to both of the two matches. Let $str_num(l)$ and $cmpt_str_num(l)$ denote respectively the number of structural relationships and that of the compatible ones between stroke segment l and the other stroke segments. Then, the stroke-segment similarity measure for a matched stroke-segment pair, say (l_i, l_t), after incorporation of the structure information is defined by modifying the original measure (as in Eq. (6)) as follows:

$$P'(l_i, l_t) = \begin{cases} P(l_i, l_t) & \text{if } str_num(l_i) = 0, \\ P(l_i, l_t) ((1 - \gamma + \gamma) SC(l_i)) & \text{otherwise,} \end{cases} \quad (20)$$

where the value $SC(l_i) = cmpt_str_num(l_i)/str_num(l_i)$ defines the structure compatibility of l_i with its neighboring stroke segments, and the parameter γ defines the portion of the original similarity value $P(l_i, l_t)$ of a stroke-segment pair (l_i, l_t), which is supported by the value of $SC(l_i)$. If the structural relationships of an input stroke segment l_i with its neighboring stroke segments are all found in those of the corresponding template stroke segments, then $SC(l_i) = 1$, and $P'(l_i, l_t) = P(l_i, l_t)$. It is noted that the instability of the structural relationships must be taken into account in setting γ . In our study, γ is set to 0.4. The value of $s(M)$, defined by Eq. (19), is also re-defined accordingly to incorporate the structure information:

$$s(M) = \frac{\sum_{(i,t) \in M} P'(l_i, l_t)}{N(M)}. \quad (21)$$

5. EXPERIMENTAL RESULTS

5.1. Inputs and Templates

Experiments were done to examine the effectiveness of the proposed approach to HCC recognition. In total, 465 Chinese character classes were selected as the testing vocabulary set. The templates were created interactively by the use of a tablet so that the stroke-segment representations for each template character could be reliable and thinning could be avoided. Figure 7 shows the stroke-segment representations of some template characters.

Two different files of input characters were tested in our experiments. One file consists of the characters selected from the Computer and Communication Laboratories handwritten Chinese character image database, version 1 (CCL-HCI1), which contains 5401 character classes, each class having 100 variants arranged according to writing qualities. Two of the best variants of the 465 character classes from the CCL-HCI1 are selected as the first testing data set. The other file consists of the same 465 characters, each with two variants written by a specific person. The characters of the second file were written with care and their qualities are much better than those of the characters selected from the CCL-HCI1, on the average. Figure 8 shows the stroke-segment representations of some input characters.

5.2. Stroke-Segment Extraction

Stroke-segment extraction for Chinese characters is an important step in recognizing handwritten Chinese characters and much work on this topic has been done.¹⁸⁻²¹ In this study, stroke-segment extraction is not our main concern, so a simple procedure was used. Our procedure to extract stroke segments from input characters is divided into the following steps: thinning, tracing, line approximation, and linking.

伸克冷利吳坑壯孝尾弄
 志投杏村杉求沙沅沂牡
 秀究肚育谷豆赤車辛酉
 里京佳來卓卑和坪坡奉
 奇姓定官宜居岡岩幸店
 底延忠承昌昆明昇杭枋
 東林板松武河波治玟直
 知社股卧芳芝花芬虎金
 長門陀阿雨青亭信保前
 勇南哈城屏屋巷幽度建

Fig. 7. Some template characters used in the study (in stroke-segment representation).

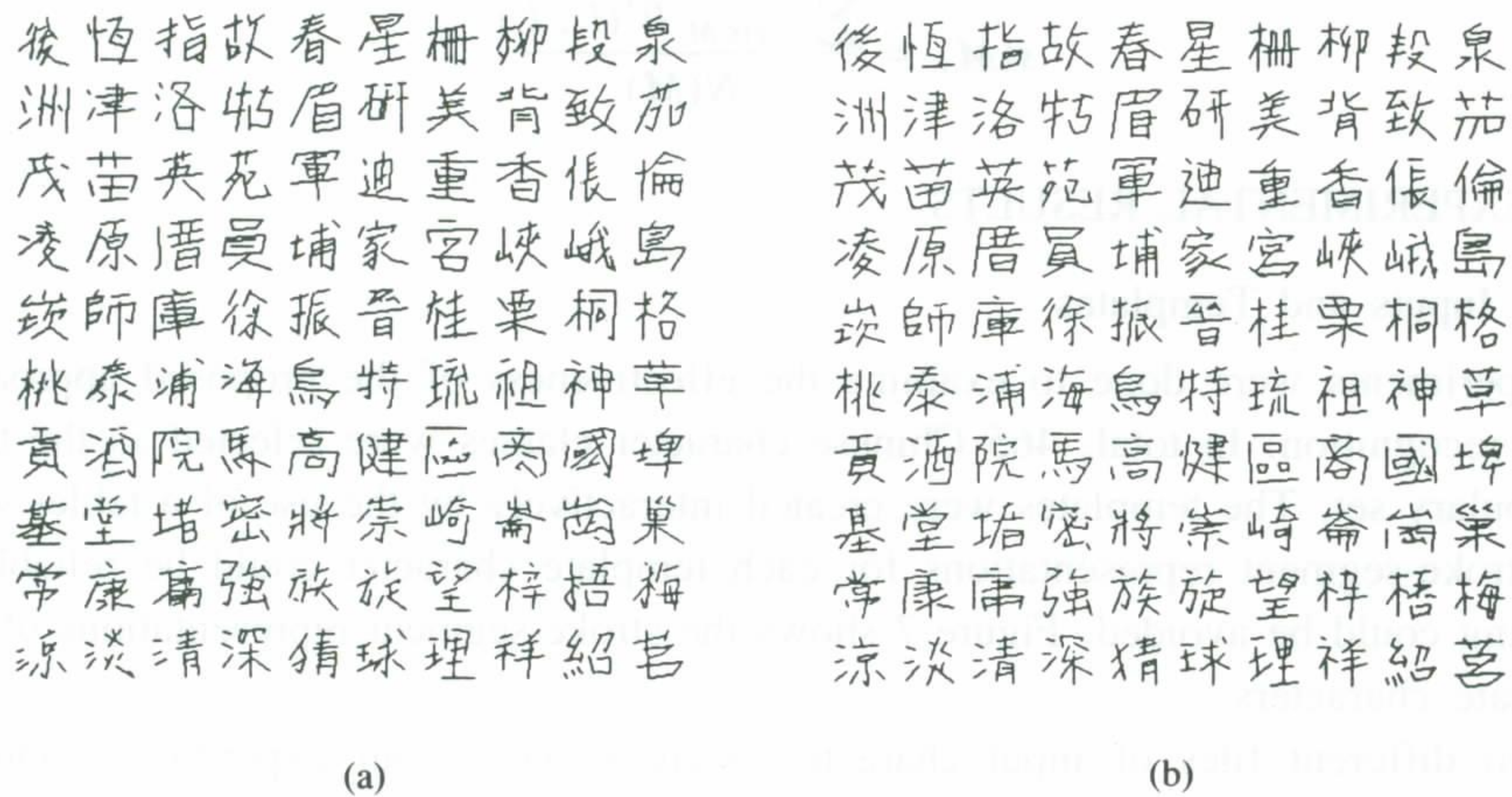


Fig. 8. Some input characters selected from the CCL-HCII (shown in (a)) and those written by a person (shown in (b)) (in stroke-segment representation).

Thinning plays an important role in the stroke-segment extraction procedure and consequently affects the recognition results. The thinning method proposed by Zhang and Suen²² was adopted in this study to thin input characters to be one pixel in width. Since concern was focused on the effectiveness of the matching technique and the similarity measures, no comparison was made of different thinning methods in this study. In fact, several thinning methods, such as the ones proposed by Wang and Zhang²³ and Lu and Wang,²⁴ have been shown to be generally better than that of Ref. 22. In particular, Lu and Wang's method is an improved version of that in Ref. 22. It is interesting to study how the recognition result of the proposed approach is affected by using different thinning methods in the stroke-segment extraction procedure.

After thinning, the resulting connecting segments are traced and segmented. The tracing is forwarded in the direction of producing a smaller angle change when a cross point is encountered so that a stroke segment can be extended as long as possible in the line approximation phase. For example, the thinning result like that shown in Fig. 9(a) will be segmented into that shown in Fig. 9(b) rather than into that shown in Fig. 9(c). Each segment l obtained in the tracing phase is then approximated²⁵ by line segments in the following way. First, connect the two end points of l to form an initial approximation line segment l' . Next, check the distances from each segment point of l and l' . If the maximum distance exceeds a certain threshold, then the segment l is broken into two at the point with the maximum distance. If the distances from all the segment points to l and l' are within a threshold, then the line approximation is done. Due to the inherent limitation of the thinning technique and tracing, a stroke segment may be broken into several pieces. To solve this problem, the conditions of connectedness and collinearity described by Eq. (16) are checked for every two stroke

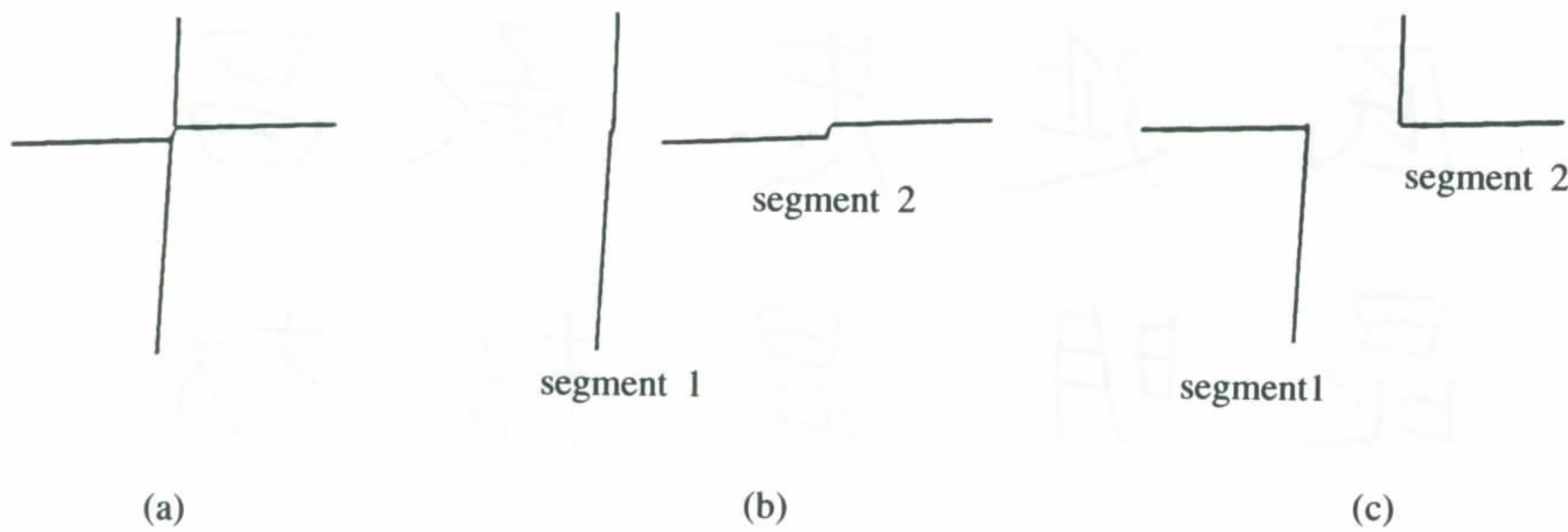


Fig. 9. Segment tracing results in stroke-segment extraction. (a) Thinning result. (b) Traced segments. (c) Undesired tracing results.

segments after the line approximation. Two stroke segments are merged into one if the two conditions are satisfied. Figure 10 shows the stroke-segment extraction results of some input characters.

5.3. Recognition Results

Before matching, each input character is preclassified coarsely by comparing its number of stroke segments with that of each template. Normally, the two numbers should not differ greatly if the input and template are of the same character class, even when writing variations exist. After preclassification, the input character is matched with each candidate template individually and a similarity measure is evaluated accordingly. In the matching, the measures of similarity between the input and template stroke segments are calculated first from Eq. (6) and Chou and Tsai's iteration scheme is then applied. In the experiments, the parameter β included in the updating formula, Eq. (15), was set to 0.25. The effect of Chou and Tsai's iteration scheme can be seen from the process of matching the character 'li' (pronunciation) with its corresponding template, as shown in Fig. 11. The input and template of the character class 'li' have nine and eight stroke segments respectively (Fig. 11(a)). The initial match network set up by the stroke-segment similarity measures computed by Eq. (6) is shown in Fig. 11(b). Each node is connected to its best match by a directed link. It is seen that the stroke segments numbered 8, 9, and 8' have no stroke segment to match and so are pruned. Four matches, $(2 \leftrightarrow 4')$, $(3 \leftrightarrow 3')$, $(6 \leftrightarrow 6')$, and $(7 \leftrightarrow 7')$, are obtained at the initial stage, but $(2 \leftrightarrow 4')$ is wrong. All the matches except the pair $(7 \leftrightarrow 7')$ are interfered with more or less by some stroke segments. For example, the match $(2 \leftrightarrow 4')$ encounters interferences from stroke segments 1, 4, 1', and 2'. So, the match relationships existing in the match network are not very stable. The match network gets more and more stable as the iterations go on, and reaches its final state as shown in Fig. 11(c) after seven iterations. All the stroke segments except the ones numbered 8, 9, and 8' are finally matched correctly.

Table 1 shows the recognition results for the two different input files. An input character is said to be recognized as being within rank k when the measure of

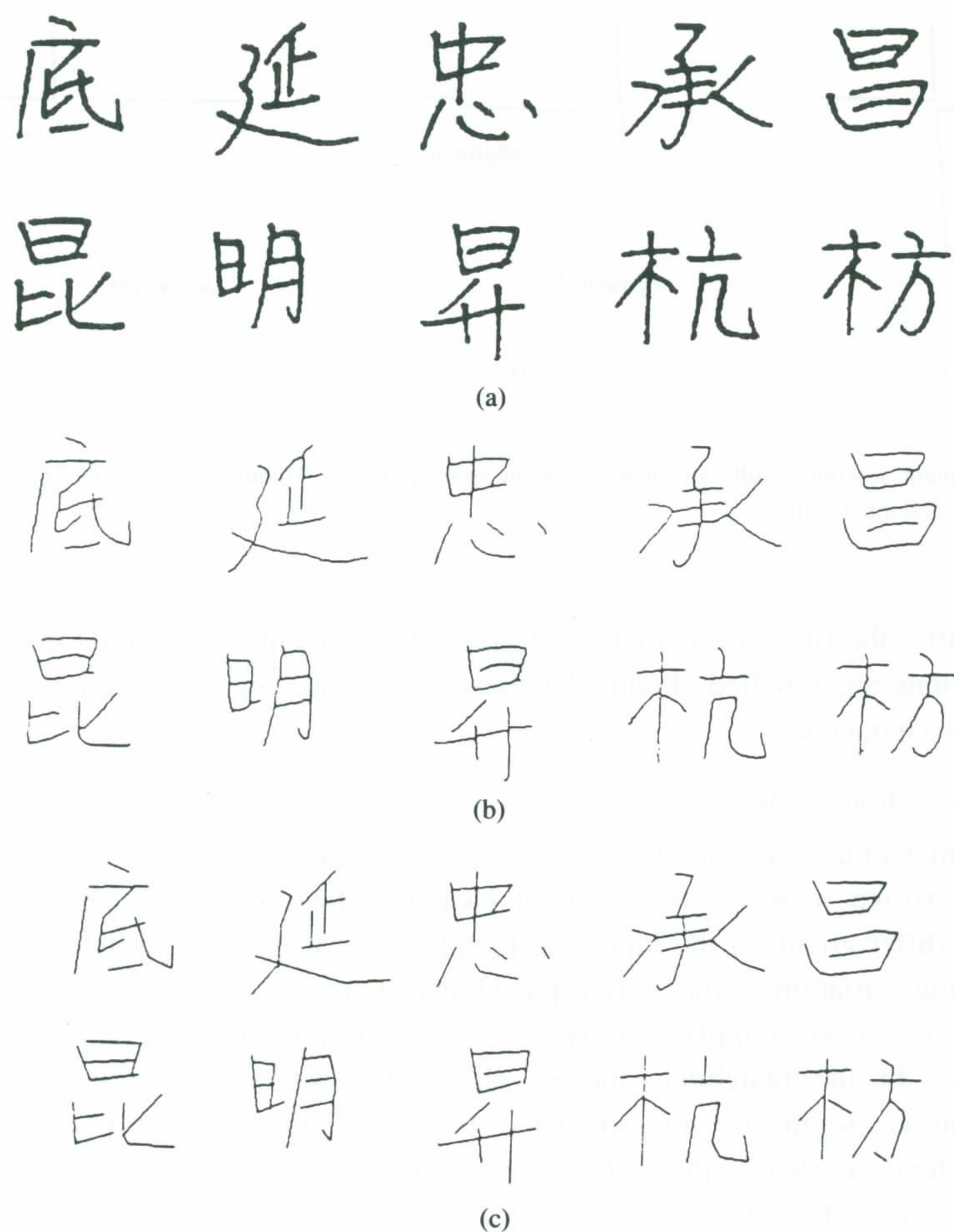


Fig. 10. Stroke-segment extraction results of some input characters. (a) Input images. (b) Thinning results. (c) Extracted stroke segments.

Table 1. Recognition results of the two different inputs.

Rank	CCL-HCI1 data		Personal writing data	
	Without iterations	With iterations	Without iterations	With iterations
1	73.54%	90.75%	84.73%	95.69%
2	82.36%	94.19%	90.32%	98.28%
3	86.66%	96.34%	93.76%	99.57%
4	89.46%	98.06%	94.41%	99.78%
5	90.53%	98.71%	95.27%	99.78%

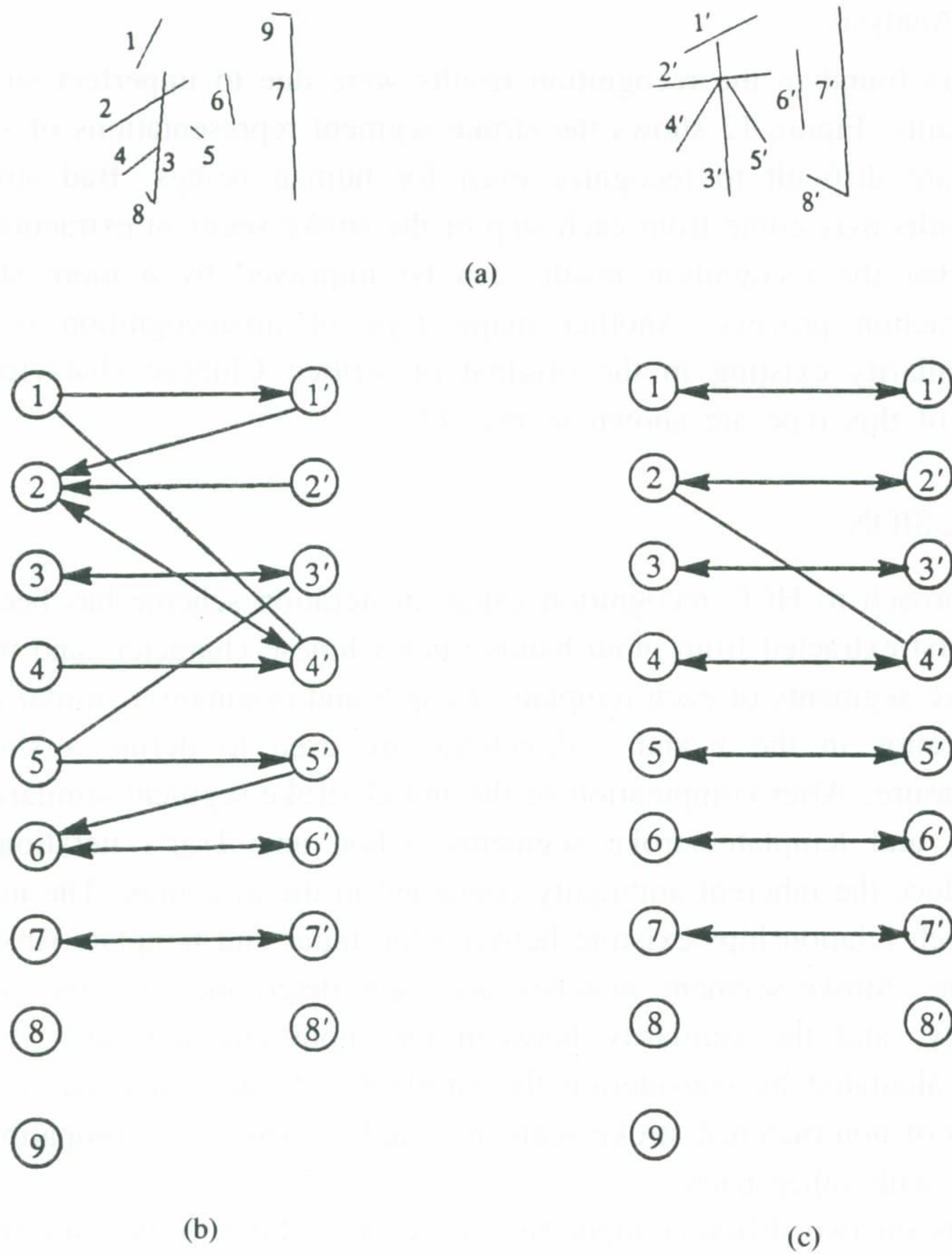


Fig. 11. Steps of Chou and Tsai's iteration scheme for matching the Chinese character 'li' (pronunciation) with its corresponding template. (a) Input and template of the Chinese character 'li'. (b) Initial match network. (c) Final match results after seven iterations.

similarity between it and its corresponding template is the k th highest. The recognition rates for the CCL-HCI1 and the personal writing data are 90.75% and 95.69% respectively. They increase to 96.34% and 99.57% respectively if input characters, when recognized to be within rank 3, are considered to be correctly recognized. Recognition results with stroke-segment matches directly determined from the initial similarity measures without iterations are also shown in the table. It is seen that the recognition rates are improved for about 17% for the CCL-HCI1 data and about 11% for the personal writing data after Chou and Tsai's iteration scheme is applied. The experiments were performed on a Sun-4 workstation and the average speed of recognition was about 3.5 seconds per character.

5.4. Error Analysis

Many errors found in the recognition results were due to imperfect stroke-segment extraction results. Figure 12 shows the stroke-segment representations of some characters, which are difficult to recognize even for human beings. Bad stroke-segment extraction results may come from each step of the stroke-segment extraction process. It is believed that the recognition results can be improved by a more stable stroke-segment extraction process. Another major type of misrecognition is due to the structural similarity existing in the original or written Chinese character structures. Some errors of this type are shown in Fig. 13.

6. CONCLUSION

A new approach to HCC recognition using an iteration scheme has been proposed. Strokes are first extracted from input handwritten Chinese characters and then matched with the stroke segments of each template. Length and orientation similarities, and the overlapping ratios in the x and y directions are used to define a stroke-segment similarity measure. After computation of the initial stroke-segment similarity measures for the input and template stroke segments, Chou and Tsai's iteration scheme is applied to reduce the inherent ambiguity contained in the measures. The iterations stop when the match relationships existing between the input and template stroke segments become stable. Stroke-segment matches are then determined by the mutually-best match strategy, and the similarity between the input character and each template character is calculated by considering the similarity of each matched stroke-segment pair, the ratio of non-matched stroke segments, and the structural compatibility of each matched pair with other pairs.

Experiments on two different input files have been done to examine the effectiveness of the proposed approach. The recognition rate is about 91% for a CCL-HCI1 data file and about 96% for a personal writing data file. The improvement of the recognition rate made by the iteration scheme is about 17% and about 13% for the CCL-HCI1 and personal writing data respectively. This shows that the proposed approach is feasible for HCC recognition.



Fig. 12. Some handwritten Chinese characters whose stroke-segment representations are not good enough.



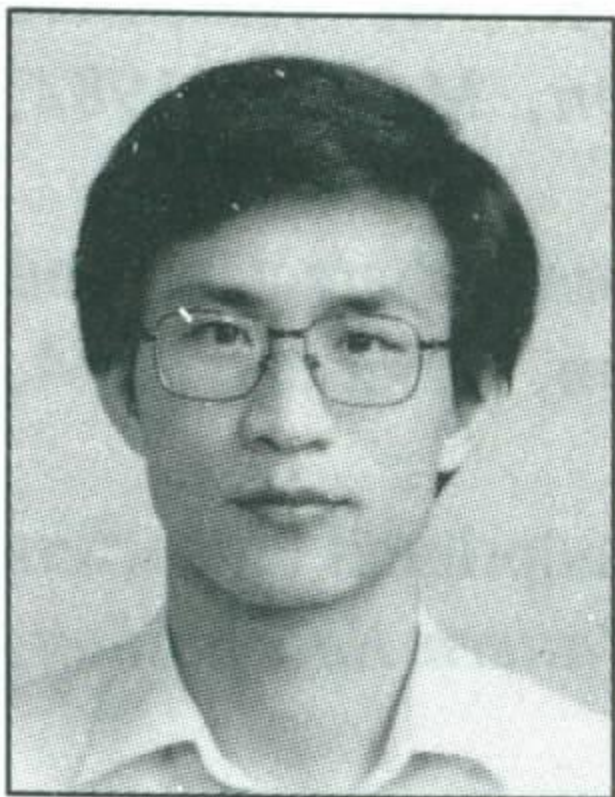
Fig. 13. Four misrecognized characters. In each of (a) through (d), the left is the input character, the middle is the correct class for the input, and the right is the erroneous class of which the input is recognized to be.

REFERENCES

1. P. S. P. Wang, "Knowledge pattern representation of Chinese characters", *Int. J. Pattern Recogn. Artif. Intell.* **2**, 1 (1988) 161-179.
2. R. Casey and G. Nagy, "Recognition of printed Chinese characters", *IEEE Trans. Electron. Comput.* **15**, Feb. (1966) 91-101.
3. C. W. Liao and J. S. Huang, "A transformation invariant matching algorithm for handwritten Chinese character recognition", *Pattern Recogn.* **23**, 11 (1990) 1167-1188.
4. S. Mori, K. Yamamoto and M. Yasuda, "Research on machine recognition of handprinted characters", *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 4 (1984) 386-405.
5. A. Rosenfeld, A. Hummel and S. W. Zucker, "Scene labeling by relaxation operations", *IEEE Trans. Syst. Man Cybern.* **6** (1976) 420-433.
6. L. Davis, "Shape matching using relaxation techniques", *IEEE Trans. Pattern Anal. Mach. Intell.* **1** (1979) 60-72.
7. K. Yamamoto and A. Rosenfeld, "Recognition of handprinted KANJI characters by relaxation method", *Proc. 6th Int. Conf. on Pattern Recognition*, Munich, 1982, pp. 395-398.
8. K. Yamamoto and H. Yamada, "Recognition of handprinted Chinese characters and Japanese cursive syllabary", *Proc. 7th Int. Conf. on Pattern Recognition*, Montreal, 1984, pp. 385-388.
9. K. Yamamoto, H. Yamada, T. Saito and I. Sakaga, "Recognition of handprinted characters in the first level of JIS Chinese characters", *Proc. 8th Int. Conf. on Pattern Recognition*, Paris, 1986, pp. 570-572.
10. S. L. Xie and M. Suk, "On machine recognition of hand-printed Chinese characters by feature relaxation", *Pattern Recogn.* **21**, 1 (1988) 1-7.
11. C. H. Leung, Y. S. Cheung and Y. L. Wang, "A knowledge-based stroke-matching method for Chinese character recognition", *IEEE Trans. Man Cybern.* **17**, 6 (1987) 993-999.
12. L. H. Chen and J. R. Lieh, "Handwritten character recognition using a 2-layer random graph model by relaxation matching", *Pattern Recogn.* **23**, 11 (1990) 1189-1205.
13. F. H. Cheng, W. H. Hsu and C. A. Chen, "Fuzzy approach to solve the recognition problem of handwritten Chinese characters", *Pattern Recogn.* **22**, 2 (1989) 133-141.
14. S. L. Chou and W. H. Tsai, "Line segment matching for 3-D computer vision using a new iteration scheme", *Proc. 1990 Int. Computer Symposium*, Hinshu, Taiwan, Republic of China, 1990, pp. 414-421.

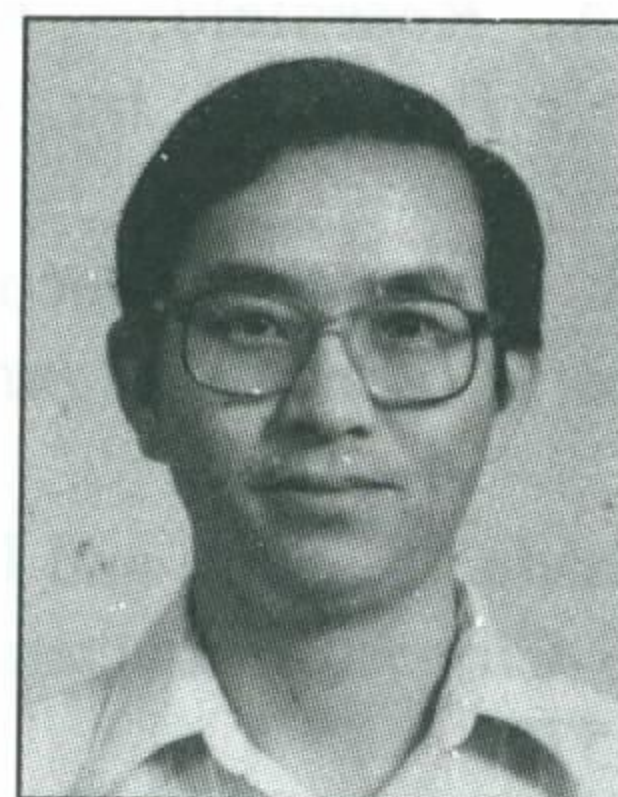
15. J. H. McIntosh and K. M. Mutch, "Matching straight lines", *Comput. Vision Graph. Image Process.* **43** (1988) 386–408.
16. W. H. Hsu and F. H. Cheng, "Recognition of handwritten Chinese characters by structure analysis of strokes", *Comput. Process. Chinese Oriental Lang.* **2** (1985) 101–112.
17. P. N. Chen, Y. S. Chen and W. H. Hsu, "Stroke relation coding—A new approach to the recognition of multi-font printed Chinese characters", *Int. J. Pattern Recogn. Artif. Intell.* **2**, 1 (1988) 149–160.
18. Y. T. Wang and W. H. Hsu, "Stroke extraction and matching methods for handwritten Chinese characters recognition", *Proc. 1984 Int. Computer Symposium*, Taichung, Taiwan, Republic of China, 1984, pp. 441–446.
19. P. S. P. Wang, "A new approach for Chinese handwriting recognition", *Proc. 1989 Int. Conf. on Electronic Imaging*, Pasadena, CA, U.S.A., pp. 373–379, 1989.
20. H. T. Tsui, "Guided stroke structure extraction for the recognition of handprinted Chinese characters", *Proc. 6th Int. Conf. on Pattern Recognition*, Munich, 1982, pp. 786–788.
21. H. Ogawa and K. Taniguchi, "Thinning and stroke segmentation for handwritten Chinese character recognition", *Pattern Recogn.* **15**, 4 (1982) 299–308.
22. T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns", *Commun. ACM* **27**, 3 (1984) 236–239.
23. P. S. P. Wang and Y. Y. Zhang, "A fast and flexible thinning algorithm", *IEEE Trans. Comput.* **38**, 5 (1989) 741–745.
24. H. E. Lu and P. S. P. Wang, "A comment on 'a fast parallel algorithm for thinning digital patterns'", *Commun. ACM* **29**, 3 (1986) 239–242.
25. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

Received 11 March 1991; revised 18 March 1991.



Sheng-Lin Chou received the B.S. degree in electrical engineering from National Tsing Hua University in 1980, and the M.S. degree in computer engineering from National Chiao Tung University in 1982.

In 1983, he began work in Electrics Research and Service Organization, Industrial Technology Research Institute (ITRI) at Hsinchu, Taiwan, as a software engineer. Now, he is an associate researcher at the Advanced Technology Center of the Computer and Communication Research Laboratories, ITRI. In 1986, he entered the Institute of Computer Science and Information Engineering at National Chiao Tung University, and is now working toward his Ph.D. degree. His current research interests include computer vision, pattern recognition, and image processing.



Wen-Hsiang Tsai received the B.S. degree from National Taiwan University in 1973, the M.S. degree from Brown University in 1977, and the Ph.D. degree from Purdue University in 1979, all in electrical engineering.

Since November 1979, he has been on the faculty of the Institute of Computer Science and Information Engineering at National Chiao Tung University, Hsinchu, Taiwan. From 1984 to 1986, he was an Assistant Director and later an Associate Director of the Microelectronics and Information Science and Technology Research Center at National Chiao Tung University. He joined the Department of Computer and Information Science at National Chiao Tung University in August 1984, acted as the head of the Department from 1984 to 1988, and is now a professor there. He is also a researcher in the Institute of Information Science at Academia Sini-

ca, Taipei. He serves as a consultant to several research institutes and industrial companies. His current research interests include pattern recognition, computer vision, image processing, and Chinese information processing.

Dr. Tsai is an Associate Editor of *Pattern Recognition*, *International Journal of Pattern Recognition and Artificial Intelligence*, *Journal of the Chinese Institute of Engineers*, *Journal of Information Science and Engineering*, and *Proceedings of the National Science Council of the Republic of China*. He was elected as an Outstanding Talent of Information Science of the Republic of China in 1986. He was the winner of the 13th Annual Best Paper Award of the Pattern Recognition Society of USA. He obtained the 1987 Outstanding Research Award and the 1988-89 and 1990-91 Distinguished Research Awards, from the National Science Council of the Republic of China. He also obtained the 1989 Distinguished Teaching Award of the Ministry of Education of the Republic of China. He was winner of the 1989 Acer Long Term Award for Outstanding Ph.D. Dissertation Supervision and the 1990 Outstanding Paper Award of the Computer Society of the Republic of China.