<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Consistent relaxation matching for handwritten Chinese character recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Jin, LW; Chan, KP; Xu, BingZ</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>Iee Conference Publication, 1995 n. 410, p. 95-99</td>
</tr>
<tr>
<td><strong>Issued Date</strong></td>
<td>1995</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10722/45567">http://hdl.handle.net/10722/45567</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.; ©1995 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.</td>
</tr>
</tbody>
</table>
CONSISTANT RELAXATION MATCHING FOR HANDWRITTEN CHINESE CHARACTER RECONSTRUCTION

L-W Jin, K-P Chan and Bing-Z Xu*

The University of Hong Kong, Hong Kong
*South China University of Technology, People's Republic of China

Abstract

Due to the complexity in structure and the various distortions (translation, rotation, shifting, and deformation) in different writing styles of Handwritten Chinese Characters (HCCs), it is more suitable to use a structural matching algorithm for computer recognition of HCC. Relaxation matching is a powerful technique which can tolerate considerable distortion. However, most relaxation techniques so far developed for Handwritten Chinese Character Recognition (HCCR) are based on a probabilistic relaxation scheme. In this paper, based on local constraint of relaxation labelling and optimization theory, we apply a new relaxation matching technique to handwritten character recognition. From the properties of the compatibility constraints, several rules are devised to guide the design of the compatibility function, which plays an important role in the relaxation process. By parallel use of local contextual information of geometric relaxation among strokes of two characters, the ambiguity between them can be relaxed iteratively to achieve optimal consistent matching.

1 Introduction

Computer recognition of handwritten Chinese characters (HCCs) is regarded very difficult due to the large size of character set, high level of complexity in character structure, similarities among different categories and wide variability among different handwriting styles. Although a lot of handwritten Chinese character recognition system achieving very high recognition rate (greater than 96%) have been reported[1]-[3], the performance of these systems are highly dependent on the quality and variability of the database used. The development of a complicated robust HCCR system which can achieve high recognition rate is still in its infancy[3].

Many relaxation techniques have been developed to solve the problem of HCCR [5]-[8]. Most of them make use of a probabilistic relaxation procedure which is originally proposed in [10]. Unfortunately, these methods would have the problem of "ambiguous matching" and "Nil class" if distortions among objects are too large[9]. Another drawback of relaxation is the heavy computation involved. In this paper, we make use of an alternative relaxation technique which is based on optimization theory and local constraint of relaxation matching[11]. A local average consistent function is defined as an objective function which will be maximized to search for an optimal consistent matching. By parallel use of local contextual information of geometric relaxation among strokes of two characters, the ambiguity between them can be relaxed iteratively to achieve an optimal consistent matching. We refer this method as Consistent Relaxation Matching (CRM). As CRM is a process of contextual matching, a contextual associative matrix which indicates the local relationship among strokes of a character is defined to guide the relaxation process. Only the contextual related strokes are taken into account in the process. In this way, the heavy computation can be reduced to make the proposed method faster and more effective.

2 Foundation of CRM

In a labelling problem, one is given: [11]

- A set of objects denoted by variable $a_i$, where $i$ takes the value between 1 and $n$.
- A set of labels for each objects denoted by variable $b_j$. Here $j$ takes the value between 1 and $m$.
- A neighbor relation over the objects.
- A constraint relation over labels at pairs of neighboring objects. The constraint are generalized to real-valued Compatibility Function $c_{i,k}(j,k')$, representing how label $b_k$ at object $a_j$ influence label $b_{k'}$ at object $a_i$. The magnitude of $c_{i,k}(j,k')$ is proportional to the strength of the constraint.
- A strength measure of how label $b_k$ is assigned to object $a_i$, denoted by $p_i(k)$, subject to:

$$0 \leq p_i(k) \leq 1$$
$$\sum_{k=1}^{m} p_i(k) = 1$$

Generally speaking, the solution to the labelling problem mentioned above is to find an assignment of label $b_k$ to object $a_i$ in a manner which is consistent
to the constraint. A consistent matching assignment is one in which the constraint is satisfied.

**Definition 1 Label Assignment Space** \( \Omega \): The space of label assignment is given by:

\[
\Omega \in \mathbb{R}^{n \times m},
\]

\[
\Omega = \{ (\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_m) \}
\]

\[
\vec{p}_i = (p_i(1), p_i(2), \ldots, p_i(m))
\]

\[
0 \leq p_i(k) \leq 1, \forall i, k; \sum_{k=1}^{m} p_i(k) = 1; \quad i = 1, 2, \cdots, n
\]

**Definition 2 Support Function**: Support function for label \( b_k \) at object \( a_i \) is defined by:

\[
S_i(k) = \sum_{j=1}^{n} \sum_{k'=1}^{m} c_{i,k}(j, k') p_j(k')
\]

Support Function \( S_i(k) \) denotes how the assignments at all other objects influence the assignment \( b_k \) at object \( a_i \).

**Definition 3 Consistent Assignment** \( \vec{P} \): Let \( \vec{P} \in \Omega \), \( \vec{P} \) is said to be consistent provided that:

\[
\sum_{k=1}^{m} S_i(k) p_i(k) \geq \sum_{k=1}^{m} S_i(k) v_i(k), \quad i = 1, 2, \cdots, n
\]

for all labellings \( (\vec{V}_1, \vec{V}_2, \ldots, \vec{V}_n) \in \Omega \).

**Definition 4 Average Local Consistency (ALC)**:

\[
A(\vec{P}) = \sum_{i=1}^{n} \sum_{k=1}^{m} p_i(k) S_i(k)
\]

\[
= \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} p_i(k) c_{i,k}(j, k') p_{j'}(k')
\]

It can be proved we have the following theorems[11]:

**Theorem 1** A label assignment \( \vec{P} \in \Omega \) is consistent if and only if:

\[
\sum_{i,j,k} c_{i,k}(j, k') p_{j'}(k') [v_i(k) - p_i(k)] \leq 0
\]

for all labellings \( (\vec{V}_1, \vec{V}_2, \ldots, \vec{V}_n) \in \Omega \).

**Theorem 2** Suppose the matrix of compatibility function \( \{ r_{i,k}(j, k') \} \) is symmetric, i.e.:

\[
r_{i,k}(j, k') = r_{j,k}(i, k), \quad \text{for all} \quad i, j, k, k'
\]

If \( A(\vec{P}) \) attains a local maximum at \( \vec{P} \in \Omega \), then \( \vec{P} \) is a consistent labeling.

Theorem 2 shows that maximizing \( A(\vec{P}) \) leads to consistent labeling assignment. The conventional method to find the maximum of \( A(\vec{P}) \) is by gradient ascent, that is, the increase in \( A(\vec{P}) \) due to a small step of length \( \alpha \) in the direction of \( \vec{u} \) is approximately the directional derivation:

\[
A(\vec{P} + \alpha \vec{u}) - A(\vec{P}) = \frac{\partial}{\partial \alpha} \big|_{\alpha=0} [A(\vec{P} + \alpha \vec{u})]
\]

\[
= \alpha \vec{u} \cdot \nabla A(\vec{P})
\]

where \( \nabla A(\vec{P}) \) is the gradient of \( A(\vec{P}) \) which can be calculated by:

\[
g_i(k) = \frac{\partial}{\partial p_i(k)} \left[ \sum_{j=1}^{n} \sum_{k'=1}^{m} p_i(k) p_j(k') c_{i,k}(j, k') \right]
\]

\[\cdots\]

\[
= \sum_{j=1}^{n} \sum_{k'=1}^{m} c_{i,k}(j, k') + c_{j,k}(i, k) p_j(k')
\]

Accordingly, if the directional vector \( \vec{u} \) has been found, the updating scheme in relaxation is in the simple way:

\[
\vec{P}^{t+1} = \vec{P}^t + \alpha \vec{u}
\]

The problem of finding \( \vec{u} \) has been solved in [12].

### 3 Applying CRM to HCCR

Handwritten Chinese characters can be represented by a set of stokes \( \{ a_1, a_2, \ldots, a_n \} \). Each stroke is featured by:

- the coordinate of its starting point \( M_i = (x_i, y_i) \);
- the coordinate of its ending point \( M_e = (x_e, y_e) \);
- the coordinate of its central point \( M_c = (x_c, y_c) \);
- the length of the stroke \( l \);
- the slope angle \( \theta \).

Let \( a_i \) and \( b_j \) be the \( i \)-th stroke and \( j \)-th stroke of two characters respectively. Matching these two characters is the process of finding a way to assign each of \( b_j \) to \( a_i \) accordingly.

An initial assignment of \( b_j \) to \( a_i \) should take into consideration of the similarities of the length, slope angle, and distance between these two characters. A conceivable definition of the initial assignment could be given by:

\[
p(k) = \begin{cases} 
1 & \text{if} \; \theta_k \text{ is similar to } \theta_i, \\
0 & \text{otherwise}
\end{cases}
\]

where \( l_k = l_i - l_k \), \( \theta_k = \theta_i - \theta_k \), \( d_k = \| M_{ic} - M_{ek} \| \), and

\[
f_i(x) = \exp(\eta_1 x^2) - 1, \; f_j(x) = \exp(\eta_2 x^2) - 1, \; f_{ij}(x) = \exp(\eta_3 x^2) - 1.
\]

Here \( \eta_1, \eta_2, \eta_3 \) are some parameters predetermined.
We say that \( b_k \) is similar to \( a_i \) if:
\[
\begin{align*}
|l_i - l_k| &< \alpha_1 \\
|\theta_i - \theta_k| &< \alpha_2 \\
d_{ik} &< \alpha_3
\end{align*}
\]
where \( \alpha_1, \alpha_2, \alpha_3 \) are some proper parameters.

The compatibility function \( c_{i,k}(j,k') \) which plays an important role in relaxation is usually determined by empirical justification. Here we will attempt to describe some properties of \( c_{i,k}(j,k') \) and develop rules which would guide the design of the form of compatibility function.

Compatibility Function \( c_{i,k}(j,k') \) has the following properties:

1. \( c_{i,k}(j,k') \) signifies the relative support for stroke \( b_k \) match stroke \( a_i \) that arise from stroke \( b_j \) match \( a_j \).

2. \( c_{i,k}(j,k') \) denotes contextual information between pairs of strokes \((a_i, b_k)\) and \((a_j, b_k')\).

3. If matching stroke \( a_j \) to stroke \( b_k \) leads to high support to matching stroke \( a_i \) to \( b_k \), then \( c_{i,k}(j,k') \) shall be large and positive. If \( a_j \) matches \( b_k \) lead to low support to \( a_i \) matching \( b_k \), then \( c_{i,k}(j,k') \) shall be negative. If there is no interaction between pairs of \((i,k)\) and \((j,k')\), then \( c_{i,k}(j,k') \) shall be zero.

From the properties of \( c_{i,k}(j,k') \) described above, the following rules are devised which \( c_{i,k}(j,k') \) should obey.

Rule 1 \(-1 \leq c_{i,k}(j,k') \leq 1\)

Rule 2 \( c_{i,k}(j,k') \) would be zero if:
- \( a_i \) is not similar to \( b_k \)
- \( a_j \) is not similar to \( b_k' \)
- \( a_i \) is not associated with \( a_j \)
- \( b_k \) is not associated with \( b_k' \)

A stroke \( a_i \) is said to be associated with stroke \( a_j \) if:
\[
\max \{d_{ij}, d_{ij}', d_{ij}''\} \leq \gamma
\]
where
\[
d_{ij}', d_{ij}'' \text{ are defined in the same way.}
\]

Rule 3 \( c_{i,k}(j,k') \) would be 1 if \( i=j \) and \( k=k' \).

Rule 4 \( c_{i,k}(j,k') \) would be -1 if \( i=j \) but \( k \neq k' \) or \( k=k' \) but \( i \neq j \).

Rule 5 If \( p_i(k) \) has high value and \( p_j(k') \) has high value, \( c_{i,k}(j,k') \) would be positive. If both \( p_i(k) \) and \( p_j(k') \) have low value, \( c_{i,k}(j,k') \) would be negative.

Based on the general consideration of \( c_{i,k}(j,k') \), one reasonable way to define the compatibility function is given by:
\[
c_{i,k}(j,k') = \begin{cases} 
1 & : \text{Rule 3} \\
0 & : \text{Rule 2} \\
-1 & : \text{Rule 4} \\
\frac{|p_i(k) + p_j(k')|}{2} \sin[(p_i(k) + p_j(k') - \tau)] & : \text{Rule 5}
\end{cases}
\]

where \( \tau \) and \( \sigma \) are some proper parameters predetermined.

After the initial assignment and compatibility function have been determined, the CRM technique is employed in the relaxation process. The complete algorithm is described in Appendix A.

4 Experiments

4.1 Database used

100 categories of handwritten Chinese characters are used in the experiment, as shown in figure 1. Each category consists of 35 different writing styles of characters, thus giving a total of 3500 samples of experimental datum. Every character has previously been digitized and normalized to an image pattern of \( 64 \times 64 \) pixels.

Figure 1: 100 categories of experimental datum.

4.2 Feature primitive extraction

The normalized character pattern is first skeletonized. A stroke extractor based on feature points of the skeleton is then employed to extract all the strokes from the character[13]. Figure 2 shows some of the results of stroke extraction.

4.3 Using contextual associative matrix to guide the relaxation process

One kind of the heavy computation involved in CRM is to compute the support function for all pairs of stroke \((i,k)\):
\[
S_i(k) = \sum_j \sum_{k'} p_i(k)c_{i,k}(j,k') \quad (11)
\]

Computation complexity of calculating (11) for all \((i,k)\) pairs is \( O(m^2n^2) \). However, it is not necessary to
perform all the computation in equation (11). In fact, as the relaxation matching is a process that makes local contextual information to search for an optimal matching, when stroke $a_i$ is matched with stroke $b_k$, only those strokes which are contextual related with $a_i$ and $b_k$ are needed to be considered. A contextual association matrix which indicates the contextual relationship between pairs of stroke in a character is defined as:

$$\text{CAM}(i,j) = \begin{cases} 1 & \text{if stroke } i \text{ and } j \text{ are associated} \\ 0 & \text{otherwise} \end{cases}$$

Therefore, when computing the support function $S_t(k)$, we first check the CAM value related with strokes $i$ and $k$. Only those strokes with non-zero CAM value are taken into consideration. In this way, the large computation involved can be reduced. In fact, each sub-stroke in a Chinese character is usually associated with less than eight sub-strokes. Suppose the average number of total sub-strokes in a Chinese character is twenty-four, then computing all $S_t(k)$ guided by CAM is nine times faster than normal scheme.

4.4 Experimental result

Two characters from the experimental data are selected as the templates for each category, the others are used as testing data (Containing 33 characters per category). Matching similarity between two characters is measured by the Local Average Consistency defined in equation (3). When an unknown character is given, it is matched with all the templates, the one which gives the largest ALC value is picked as recognition result. If two or more templates produce almost the same ALC value (their difference less than a given threshold), then this character is rejected. Experiment on total 3300 characters produces the recognition rate of 95.1%, rejection rate of 2.6% and error rate of 2.3%.

5 Conclusion and Discussion

In this paper, based on the consistent relaxation technique, we developed a new relaxation matching scheme for handwritten Chinese character recognition. In the application of a relaxation matching process, there are several important parts which should be carefully considered:

- the feature primitive used,
- initial matching assignment,
- compatibility and supporting function,
- updating scheme,
- similarity measurement.

Among these factors, it is thought that the proper selection of feature primitive is of the most important and thus the performance of a recognition system would be highly dependent on how good the feature primitive has been extracted. In our study, we have used strokes of a Chinese character as feature primitive. It is good for constrained handwritten Chinese characters which are clearly written. However, for unconstrained handwritten Chinese characters, strokes extracted are not stable. In such case, a better feature primitive with more stable contextual information invariant to different writing styles is desired.

Appendix A: Complete CRM Algorithm

Step 1 Assign the initial matching value $p_i(k)$ and compute the compatibility function $c_{ij}(k,k')$ accordingly, for all $i, j = 1, 2, \ldots, n; k, k' = 1, 2, \ldots, m$. Set $t=0$.

Step 2 Compute $\tilde{q}^{(t)} = \frac{1}{2} \text{grad}[A(p^{(t)})] = S_t(k)$.

Step 3 Set $t'= 0$, $S_t^{(t')} = \Phi$, $D_i = \{k \mid p_i(k) = 0\}$, $i = 0, 1, \ldots, n$.

Step 4 Let:

$$w_i^{(t')} = \frac{1}{n-n_i} \sum_{k=1}^{m} q_i(k)$$

$ns$ = number of elements in set $S_i^{(t')}$

$$S_i^{(t'+1)} = \left\{ k \in D_i \mid q_i(k) < w_i^{(t')} \right\}, i = 1, 2, \ldots, n$$

Step 5 If $S_i^{(t'+1)} \neq S_i^{(t)}$, go to Step 4, else go to Step 6.

Step 6 Compute:

$$y_i(k) = \begin{cases} q_i(k) - w_i^{(t')} & \text{if } k \in S_i \\ 0 & \text{otherwise} \end{cases}$$

Step 7 Compute:

$$u_i^{(t)}(k) = \begin{cases} 0 & \text{if } ||y_i^{(t)}|| = 0 \\ w_i^{(t)}(k) & \text{otherwise} \end{cases}$$
where
\[ ||\vec{y}|| = \sqrt{\sum_{i}^{m} \sum_{k}^{n} |y_{i}(k)|^2} \]

**Step 8** Updating assignment value according:
\[ p_{i}^{(t+1)}(k) = p_{i}^{(t)}(k) + \xi \delta_{i}^{(t)}(k), \forall i, k \]
where \( \xi \) is a small value predetermined, and may decrease as \( t \) increase to facilitate convergence.

**Step 9** If \( \sum_{i} \sum_{k} |P_{i}^{t+1}(k) - P_{i}^{t}(k)| \leq \epsilon \), stop, (here \( \epsilon \) is a small value), otherwise set \( t = t+1 \), go to Step 2.

**References**


