Finite geometrical relations loading in Hopfield model

Cheng-Yuan Liou¹, Hong-Yi Shih¹, Daw-Ran Liou²

¹ Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
² The Sibley School of Mechanical and Aerospace Engineering, Cornell University, USA

cyliou@csie.ntu.edu.tw, egg30107@gmail.com, dawran6@gmail.com

Abstract: Hand-printed character recognition is an important application in our life. A method called the feature-to-feature adhesion method is developed to achieve the task. There are two geometrical relations constructed in an earlier method. This work rewrites geometrical relations and constructs them into the Hopfield model to improve the matching result. We also provided some new applications which can be solved by our improved method.

Keywords: Hand-printed character recognition, Pattern recognition, Hopfield model, Geometrical relation, Bended-ellipse feature

1. Introduction

Hand-printed character recognition is an important application in our life. Many of the technological products nowadays, such as the smart phone and tablet PC, allow usage of handwriting as input. The recognition system for license plates or for the postal codes are also the familiar applications of the hand-printed character recognition system.

There are several different methods to accomplish hand-printed character recognition. For Chinese characters, a familiar way to recognize them is to find their radicals first because these radicals are limited and the recognition is relatively easy. However, most of those methods have the common disadvantage that they usually require precise extraction of features and radicals [1], [2], [3]. In the feature-to-feature adhesion method developed by Liou and Yang [4], they do not segment a pattern into radicals in advance. Instead of extracting radicals of a pattern, they calculate the probability of each template radical to be in the character and find the relation between radicals. Finally, use a Hopfield network [5] to solve a maximization problem collectively, using neural network to complete the classification of characters. This method can be applied not only to character recognition, but also to many other pattern recognition tasks so that the pattern can be divided into small components.

In order to generate the features for template radicals and the patterns, Liou and Yang used the bended-ellipse features [6]. Each feature of a radical or pattern is represented by a five dimensional vector which include the coordinates, the direction, the angle, and the lengths information. Once the features are generated, they will find the topological relations between those features. They simply obtain a feature-to-feature (FTF) order by defining the “neighbor” of the features. In this way the geometrical relation will be preserved. Each FTF order can be represented by a symmetric matrix.

After obtaining bended-ellipse features and FTF order information, we can begin the classification. It is achieved by measuring the compatibility of every radical with the hand-printed
pattern and standard pattern. The standard pattern which minimizes the dissimilarity is the classification result. For this purpose, Liou and Yang define two similarity measurements called inter-feature similarity and inter-link similarity to measure the compatibility. The inter-feature similarity measures the similarity between two features and the inter-link similarity measures whether both corresponding feature pairs are neighbors. Together with these two similarities, the computing of the compatibility can be formulated as an optimization problem, which can be solved by a developed Hopfield network.

The original feature-to-feature adhesion method uses only two similarities for classification. There are some other relations of features that can be included for more accurate match. In this paper, we improve the method by making some changes to original rules and adding new similarities among features. We also give some other examples of application of this method by reusing the calculated similarities.

The rest of the paper is organized in 5 sections. In Section 2 we introduce the feature-to-feature adhesion method, including bended-ellipse features, FTF order, and the details of the classification. In Section 3 we give new conditions and changes for classifying improvement. Section 4 discusses some new applications and shows our results. Section 5 provides summary and conclusion.

2. Feature-to-Feature Adhesion Method

We will review the original feature-to-feature adhesion method in this section. There are two major steps in this method and we will discuss them in the following three subsections. In the first and the second subsection we introduce the method for generating bended-ellipse features and feature-to-feature order for a given radical or a pattern [6]. In the third subsection we show how to use above information and a devised Hopfield network to achieve the classification.

2.1. Bended-Ellipse Features

Chinese characters are composed by limited radicals. Some radicals may have similar shapes but in fact they are in different sizes or locations in the characters. Fig. 1 shows
two examples that show that radicals appear in different patterns. We should regard them as different template radicals. This means each radical should be shifted and normalized according to its position and size in the characters. The shapes of these radicals are relatively easy and we want to decompose them into small features.

We chose $L$ template radicals $R_1, R_2, \ldots, R_L$. Each radical is composed of a set of features. For a given radical $R_j$, $1 \leq j \leq L$, we will generate $L^j$ bended-ellipse features. So we can write $R_j = \{r_{jl} \mid 1 \leq l \leq L^j\}$, where $r_{jl}$ is the $l$th feature vector of the $j$th template radical.

Now we discuss the method used to generate the bended-ellipse feature fields. Given a radical $R_j$, we choose $L^j$ seeds $p_{j\ell}$, $1 \leq \ell \leq L^j$. $p_{j\ell}$ is usually obtained by uniformly sampling along the skeleton of the radical. We then find the most significant directions (MSDs) for each seed. A direction (with respect to a seed) is an MSD if the length of the line extending from the seed in this direction is a local maximum. The line must not pass the unprinted area. Without loss of generality, we subsample all the direction in $[0, 2\pi)$ to decrease the computation. Subsample by every angle $\frac{2\pi}{72}$ is a typical choice for preserving accuracy. We grow lines from the seed along these directions and stop growing when the lines reach the boundary. Fig. 2(a) shows an example of finding MSD lines.

![Figure 2](image_url)

**Figure 2.** (a) The MSD lines. (b) The bended-ellipse feature (five dimensional vector).

Define $\bar{L}_\ell^j(\alpha)$ be the line segment begins from seed $p_{j\ell}$ along direction $\alpha$ and ends at the boundary. Let $|\bar{L}_\ell^j(\alpha)|$ denote the length of $\bar{L}_\ell^j(\alpha)$. Using the following rule, we can find the MSDs of $p_{j\ell}$:

Direction $\alpha$ is a MSD if

$$\frac{1}{2w+1} \sum_{q=-w}^{w} |\bar{L}_\ell^j(\alpha + q \cdot \frac{\pi}{36})| > \frac{1}{2w+1} \sum_{q=-w}^{w} |\bar{L}_\ell^j(\alpha + (q + o) \cdot \frac{\pi}{36})|$$

and

$$\frac{1}{2w+1} \sum_{q=-w}^{w} |\bar{L}_\ell^j(\alpha + q \cdot \frac{\pi}{36})| > \frac{1}{2w+1} \sum_{q=-w}^{w} |\bar{L}_\ell^j(\alpha + (q - o) \cdot \frac{\pi}{36})|$$

for $o = 1, \ldots, O$ (1)
where \( w \) and \( O \) are constants. \( w \) is used to control the window size for averaging and \( O \) decides the range of selecting the maximum. We use \( w = 1 \) and \( O = 1 \) in our work. If \( \alpha \) is an MSD, we call \( \tilde{L}_j(\alpha) \) an MSD line. Fig. 2(a) shows two MSD lines.

After obtaining the MSD lines, we are ready for generating bended-ellipse features. Recall that we picked \( L^j \) seeds from the \( j \)th template radical \( R^j \) and we will generate \( L^j \) bended-ellipse features \( \{ r^j_l | 1 \leq l \leq L^j \} \). Note that each seed may create more than one feature. So we have \( L^j \geq L^j \). These bended-ellipse features can be represented by five-dimensional vectors as follows. Let \( r^j_l = [x^j_l, y^j_l, u^j_l, \phi^j_l, \hat{x}^j_l, \hat{y}^j_l] \) be the five-dimensional vector. Without loss of generality we may assume that \( r^j_l \) is generated from the seed \( p^j_\ell \). The elements \( x^j_l \) and \( y^j_l \) in \( r^j_l \) are the coordinates of seed \( p^j_\ell \). Let \( r^j_{\ell'} \) and \( r^j_{\ell''} \) be the lengths of the two MSD lines. We set

\[
u^j_l = \frac{r^j_{\ell'} + r^j_{\ell''}}{2}.
\]

be the average length of the two arms. Let \( \tilde{\phi}^j_l, 0 < \tilde{\phi}^j_l \leq \pi \), be the angle between the two MSD lines. We define

\[
\phi^j_l = \pi - \frac{\tilde{\phi}^j_l}{\pi}.
\]

Let \( (\hat{x}^j_l, \hat{y}^j_l) \) be the unit vector that extends from the seed \( p^j_\ell \) and equally divides the angle \( \tilde{\phi}^j_l \). Now every element in \( r^j_l \) is well-defined. Fig. 2(b) depicts a bended-ellipses feature example.

Note that we will discard some deformity ellipses. It will exclude several kinds of noise and decrease the complexity in our later processes. It will speed up the execution time also. An ellipse with \( \tilde{\phi}^j_l = 0 \) is a degenerated ellipse that will be discarded. A feature is also defined as a defect if a small arm of the arm ratio is not uniform. More precisely, we will use the ellipses which satisfy \( r^j_{\ell'} > r^* \), \( r^j_{\ell''} > r^* \) and \( 0.5 \leq r^j_{\ell'}/r^j_{\ell''} \leq 2 \), where \( r^* \) is a constant. In practice we set \( r^* \) to be the stroke width of the template pattern.

If a seed has more than two arms, it will generate more than one bended-ellipse feature. For example, a seed with 3 arms can generate 3 bended-ellipses. In general, a seed with \( n_a \) arms, where \( n_a \geq 2 \), can generate

\[
\binom{n_a}{2}
\]

bended-ellipses. So a branch will generate \( C(3, 2) = 3 \) bended-ellipses and a cross will generate \( C(4, 2) = 6 \) bended-ellipses. Fig. 3 gives some examples for indicating that. We will group the features generated from the same seed to simplify our later processes.

![Figure 3. A seed with different arm number can generate more than 2 bended-ellipses.](image-url)
the same process to every of $N$ template patterns to obtain $S^i = \{s^i_n|1 \leq n \leq N^i\}$ for $1 \leq i \leq N$, where $s^i_n = [x^i_n, y^i_n, u^i_n, \phi^i_{nx}, \phi^i_{ny}]$ is the $n$th feature vector of the $i$th template pattern. For an unknown handprinted pattern input, we also apply the same process to it and we get $H = \{h_m|1 \leq m \leq M\}$, where $h_m = [x_m, y_m, u_m, \phi_{mx}, \phi_{my}]$ is the $m$th feature vector of the unknown handprinted pattern.

2.2. Feature-to-Feature Order

As soon as we obtain the bended-ellipse features, we can compute the FTF order. This order will preserve the geometrical relations between features. The idea is from the intelligent cell-to-cell adhesion mechanism [7], [8]. Fig. 4 sketches the concept of this mechanism. We give each block a number and keep track of the interface relations between them. There will be a unique mark on the interface between two connected blocks. With these marks, the topological structure can be reconstructed correctly. Fig. 4(a) shows the reconstruction of the character and Fig. 4(b) gives the matrix representation of the feature-to-feature order.

![Figure 4](image)

(a) (b)

Figure 4. (a) The scattered blocks. (b) Matrix representation of the topological order.

The computation of the FTF order for bended-ellipse features is similar to the concept of the cell-to-cell adhesion mechanism but simpler. In our method the features are similar to the blocks in Fig. 4(a). For a given feature, we define the topological neighbor of this feature to be the feature which overlaps it. We then set a link between two features if these two features are topological neighbors. A link is just like a marked adhesion except it is not a unique mark. The uniqueness is not necessary because our method will insure that there is at most one link between two features. So the only information we need to know is whether two features are topological neighbors or not. The FTF order can be represented by an undirected graph, where the vertices stand for the features and the edges stand for the links. Similar to cell-to-cell adhesion, the links can be represented by a symmetric matrix. Fig. 5 gives one example. We use the same character example to show the similarities and differences. Note that each node (including the degree one nodes) in Fig. 5(a) stands for a seed.

We apply this method to every template radical $R^j$, $1 \leq j \leq L$. For each $R^j$, the FTF order we obtained is represented by an $L^j \times L^j$ matrix $\Psi^j$. Formally, define $\Psi^j(l_1, l_2)$ to be
the element in the $l_1$th row and $l_2$th column of matrix $\Psi^j$. And define
\[ \Psi^j(l_1, l_2) = \begin{cases} 1, & \text{if there is a link between } r^j_{l_1} \text{ and } r^j_{l_2} \\ 0, & \text{otherwise.} \end{cases} \]

The same process is applied to the template patterns and unknown handprinted pattern. We use an $N \times N$ matrix $\Gamma^i$ to represent the FTF order of the $i$th template pattern $S^i$ for $1 \leq i \leq N$ and an $M \times M$ matrix $Y$ to represent the FTF order of the unknown handprinted pattern $H$. Fig. 5(b) shows an example, a dark box in the matrix means that there is a link between the corresponding two seeds.

2.3. Classification

We are now ready for finding the most likely standard pattern for the hand-printed character. Before we discuss the method of the classification, we list the information we have known.

— $L$ template radicals $R^j = \{r^j_l|1 \leq l \leq L\}$, $1 \leq j \leq L$ and the FTF order $\Psi^j$ for each $R^j$.

— 1 unknown handprinted pattern $H = \{h_m|1 \leq m \leq M\}$ and the FTF order $Y$ for $H$.

— $N$ template patterns $S^i = \{s^i_n|1 \leq n \leq N\}$, $1 \leq i \leq N$ and the FTF order $\Gamma^i$ for each $S^i$.

We will compare the compatibility of every $R^j$ with $H$ and every $S^i$, then choose the most likely template pattern to be the classification result. Formally speaking, let $c^h_j$ be the compatibility between the $j$th radical $R^j$ and the handprinted pattern $H$. We will discuss the method of computing the compatibility $c^h_j$ later. Let $\vec{c}^h = [c^h_1 \ldots c^h_L]^T$ be the compatibility vector that specifies the overall compatibility between $H$ and all radicals $R^1 \ldots R^L$. For each template pattern $S^i$, $1 \leq i \leq N$, we apply the same process to compute the overall compatibility between $S^i$ and all radicals $R^1 \ldots R^L$. We represent this overall compatibility by a vector $\vec{c}^i = [c^i_1 \ldots c^i_L]^T$.

Note that the only things we need to compute on-line is the vector $\vec{c}^h$. The compatibilities between every $S^i$ and every $R^j$ can be pre-computed off-line, which means that the vectors
corresponding feature pairs. Here we will introduce the two similarity measurements in the original feature-to-feature adhesion method. The first similarity, called inter-feature similarity, measures the similarity between two feature vectors. Let \((r^j_{1k}, h_{mk})\), where \(r^j_{1k} \in \mathbb{R}^j\) and \(h_{mk} \in \mathbb{H}\), be one feature pair. We define the inter-feature similarity as

\[
D_1(r^j_{1k}, h_{mk}) = -\|r^j_{1k} - h_{mk}\|
\]

The Euclidean distance indicates the similarity of these two features. The bigger value of \(D_1\) specifies that the two features are more similar. The second similarity is the inter-link similarity, which measures the similarity between the links of the corresponding feature pairs. More precisely, let \((r^j_{1i}, h_{m1})\) and \((r^j_{1j}, h_{m2})\) be the two corresponding feature pairs. We say that these two feature pairs have high inter-link similarity if

\[
\Psi^j(l_1, l_2) = 1 \quad \text{and} \quad Y(m_1, m_2) = 1.
\]

That is, there is a link between \(r^j_{1i}, r^j_{1j}\) and there is a link between \(h_{m1}, h_{m2}\). Together with these two similarities, we can define the compatibility between the two corresponding feature pairs \((r^j_{1i}, h_{m1})\) and \((r^j_{1j}, h_{m2})\) as follows:

\[
D_2(r^j_{1i}, r^j_{1j}, h_{m1}, h_{m2}) = \begin{cases} 
\frac{D_1(r^j_{1i}, h_{m1}) + D_1(r^j_{1j}, h_{m2})}{2} - \mu, & \text{if } \Psi^j(l_1, l_2) = 1 \quad \text{and} \quad Y(m_1, m_2) = 1 \\
-\mu, & \text{otherwise.}
\end{cases}
\]

where \(\mu\) is a large positive constant or 0. We set \(\mu = 2\) in our work. Note that under this definition the compatibility of the features is not commutative. That is,

\[
D_2(r^j_{1i}, r^j_{1j}, h_{m1}, h_{m2}) \neq D_2(r^j_{1j}, r^j_{1i}, h_{m1}, h_{m2})
\]

and

\[
D_2(r^j_{1i}, r^j_{1j}, h_{m1}, h_{m2}) \neq D_2(r^j_{1i}, r^j_{1j}, h_{m2}, h_{m1}).
\]

It is straightforward to compute the total compatibility between a radical and a pattern. One must just sum up all the compatibilities of all the corresponding feature pairs. The key problem is finding the correspondence between the radical features and pattern features. An idea to achieve this goal is solving the subgraph matching problem. Recall that with the bended-ellipse features and the FTF order, a radical or a pattern can be represented by an undirected graph (Fig. 5(a)). Let the graph of the \(j^{th}\) radical \(\mathbb{R}^j\) be \(G^j\) and the graph of the handprinted pattern \(\mathbb{H}\) be \(G^h\). The subgraph matching problem is finding the one-to-one correspondence between the nodes of \(G^j\) and \(G^h\).
We will use a route $\eta^j$ to denote the matched features. Formally we define $\eta^j = \{(r^j_l, h^j_m) | 1 \leq l \leq L^j\}$, where $h^j_m$ is matched to $r^j_l$. If the route $\eta^j$ has been decided, the total compatibility between $R^j$ and $H$ can be computed using the equation in (2). We note that the point correspondence with maximal overall compatibility should have the sense that both similarity measurements are relatively large. So we formulate these objectives as the following optimization problem:

$$\text{Maximize } \sum_{1 \leq l_1, l_2 \leq L^j} \sum_{1 \leq m_1, m_2 \leq M} D_2(r^j_{l_1}, r^j_{l_2}, h^j_{m_1}, h^j_{m_2})$$

(3)

This optimization problem can be solved by a Hopfield network [5]. There are some other works used a Hopfield network to solve the pattern recognition problem without involving the FTF order [9], [10]. The feature-to-feature adhesion method we used gives us a topological intuition to formulate the energy function.

We use the connectivity matrix $V$ to represent a route in the Hopfield network. $V$ is defined as an $L^j$ by the $M$ matrix, where the $l$th row stands for the feature $r^j_l$ of the radical $R^j$, and the $m$th column stands for the feature $h^j_m$ of the handprinted pattern $H$. This route will give us the point correspondence we need as soon as the network has converged. Let $V_{lm}$ be the element of $V$ in the $l$th row and the $m$th column. The value of $V_{lm}$ is either $1$ or $0$. $V_{lm} = 1$ means that $h^j_m$ is matched to $r^j_l$; otherwise, $V_{lm} = 0$. We will check the state of $V$ after the network converges and add $(r^j_l, h^j_m)$ to $\eta^j$ if $V_{lm} = 1$.

Note that feature-to-feature can not be one-to-many or many-to-one; moreover, our objective is solving the optimization problem in (3). For these purposes, we need the matrix $V$ to satisfy the following three rules:

1. There is only one 1 in each row.
2. There is at most one 1 in each column.
3. The overall compatibility according to $V$ is maximal when the network converges.

We can use the following energy function to formulate the above constraints:

$$E = \frac{A}{2} \sum_l \sum_{m_1 \neq m_2} V_{lm_1} V_{lm_2} + \frac{B}{2} \sum_m \sum_{l_1 \neq l_2} V_{l_1 m} V_{l_2 m} + \frac{C}{2} \sum_l (\sum_m V_{lm} - 1)^2$$
where $A$, $B$, $C$, and $D$ are constants to be decided. Here the first three terms in (4) are designed to make sure that the matrix $V$ satisfies the first and the second rules [11], and the last term in (4) will make $V$ satisfy the third rule. In the network, the state of $V_{lm}^{(t)}$ will change as time goes on. For time step $t$, the state of $V_{lm}^{(t)}$ is defined as

$$V_{lm}^{(t)} = \frac{1}{2}(1 + \tanh(v_{lm}^{(t)} / v_0))$$

(5)

where $v_0$ is a constant. The last term in (5) is the equation of motion which changes the state of the network. This motion equation is defined as

$$\frac{\partial v_{lm}}{\partial t} = -\frac{v_{lm}}{\tau} - A \sum_{m_1 \neq m} V_{lm_1} - B \sum_{l_1 \neq l} V_{l_1 m} - C(\sum_m V_{lm} - 1)$$

$$+ D \sum_{l_1 \neq l} \sum_{m_1 \neq m} D_2(r_{l_1}^{(i)}, r_{l}^{(i)}, h_m, h_{m_1}) V_{l_1 m_1}$$

(6)

where $\tau$ is a constant.

We need to set the initial state of $V$. A reasonable thought is that if two features are matched, then they should have a relatively higher inter-feature similarity. So we use the inter-feature similarity as the criterion to set the initial value of each matched, then they should have a relatively higher inter-feature similarity. So we use the

in our work. Fig. 6 gives one example that we run the Hopfield network and get its converged result. The black box shows the place of the element which $V_{lm} = 1$. We list the matching result on the right in order to check it easily. Note that some features may not have a matching feature. We also give a possible best path on the right bottom as a reference.

After running the Hopfield network for the unknown pattern and for every template pattern, we get the overall compatibility vectors $\vec{c}^h = [c_1^h \ldots c_L^h]^T$ and $\vec{c}^i = [c_1^i \ldots c_L^i]^T$ for $1 \leq i \leq N$. For simpler pattern, we use the pattern $S^i$ which minimizes $D(\vec{c}^h, \vec{c}^i)$ as our classification result. If the patterns are more complex, we will use a four-layer backpropagation network for the classification [13]. Fig. 7 shows the network we use. This network can be sped up by using the same method as Liou and Yu [14]. We use $\vec{c}^i$, $1 \leq i \leq N$ as input to train the network, and obtain the classification result when the network converges.

3. Our improvement

We have already discussed the original feature-to-feature adhesion method. In the original method, Liou and Yang [4] considered only two similarity measurements: the inter-feature similarity and the inter-link similarity. In this section, we add two new similarity measurements for improving the matching result of the Hopfield network. We will introduce them in the first subsection. Also, we generalize the definition of the bended-ellipse feature for fitting our application and compare the difference between the original and the new definitions. We get a trade-off and we discuss this in the second subsection.
3.1. Adding new similarity measurements

Liou and Yang [4] used only the inter-feature similarity and the inter-link similarity in their classification step. There are a lot of other relations among features that we can use to classify. We now discuss two new similarity measurements. First we note that the inter-link similarity indicates whether there is a link between two features or not, but it cannot tell us the relative direction between them. To solve this problem, we add a new similarity called inter-direction similarity. This similarity measures the similarity between the directions of the corresponding feature pairs. Recall that \((r_{j_1}^l, h_{m_1})\) and \((r_{j_2}^l, h_{m_2})\) are the two corresponding feature pairs, where \(r_{j_k}^l \in R^l\) and \(h_{m_k} \in H\) for \(k = 1, 2\). The concept of inter-direction similarity is easy. For example, if \(r_{j_2}^l\) is at the right of \(r_{j_1}^l\) and \(h_{m_2}\) is at the right of \(h_{m_1}\), we say they have high inter-direction similarity. Here we use the eight cardinal directions for our classification. Fig. 8 shows the eight directions we use. In this example \(h_{m_2}\) is at the north-east of \(h_{m_1}\). We will compute the inter-direction similarity for the corresponding feature pairs according to the angle differences of the two directions.

For a given feature \(r_{j_1}^l\), the other features will be classified into three parts according to how far they are from \(r_{j_1}^l\). The first part is the features which have a link between \(r_{j_1}^l\) and each one of them. If there is a link, then the two features are close to each other. We want to know the similarity between the directions of corresponding feature pairs, but sometimes some habits in our handwriting will cause big inaccuracies for close features. An example is provided in fig. 9(a). Note that although these two patterns stand for the same character “a”, the tails of them are totally different. This example tells us that if features are too close, the
Finite geometrical relations loading in Hopfield model

Figure 9. (a) Different habits cause different handwriting results. (b) An example of a radical in a pattern, we don’t want the information from other part of the pattern causing much noise.

relative direction of them may have higher inaccuracies. We should decrease their influence by computing the inter-direction similarity.

For the feature \( r^j_{l_1} \), those features which are too far away from \( r^j_{l_1} \) are not good reference resources for measuring the inter-direction similarity either. This is because these features may not belong to the same radical in the pattern in most cases. Fig. 9(b) is a common example in Chinese characters. The template pattern at the right is composed of the radical at the left and another part. Considering the relative directions between the features at left and the features at right will provide us with wrong information, so we will abandon this part of direction information in computing the inter-direction similarity.

The remainder part are those features which have middle distances from the feature \( r^j_{l_1} \). These features are good candidates for providing the direction information. Note that in Fig. 9(a) the tails of the two patterns are different, but the directions of those features at the top-middle of the characters and the directions of the features at the tails are about the same. We can use these middle-distance features to help us decide the relative location of a given feature. After putting these ideas together, we can replace the equation (2) by the following equation:

\[
D_2(r^j_{l_1}, r^j_{l_2}, h_{m_1}, h_{m_2}) = \begin{cases} 
(1 + \lambda_1) \times \bar{D}_1, & \text{if } \Psi(l_1, l_2) = 1 \text{ and } Y(m_1, m_2) = 1 \\
-\mu, & \text{if } D_0(r^j_{l_1}, r^j_{l_2}) > \delta \text{ or } D_0(h_{m_1}, h_{m_2}) > \delta \\
- (\mu + \lambda_2), & \text{otherwise.}
\end{cases}
\]

where \( \bar{D}_1 = \frac{D_1(r^j_{l_1}, h_{m_1}) + D_1(r^j_{l_2}, h_{m_2})}{2} \)

(7)

Another similarity measurement we added is the inter-ratio similarity, which measures the similarity between the “neighbor-rate” of the corresponding feature pairs. We only compute the “neighbor-rate” for connected features, i.e., the feature pairs which there is a link between. This strategy can not only save a lot of computing time but also exclude unnecessary noise. We now introduce the “neighbor-rate” we used. Given two features with a link between them, say, \( h_{m_1}, h_{m_2} \), we can find their topological midpoint \( o \), and then count the number of features
Figure 10. (a) The concept of computing the “neighbor-rate”. (b) An example of computing the “neighbor-rate”.

in a circle that range from the midpoint \( o \). We give the two features an orientation according to their order, so we can define the “left-hand side” and the “right-hand side” of these two features. Fig. 10(a) depicts the concept of our definition. We obtain the “left neighbor-rate” and the “right neighbor-rate” by computing the percentage of the number of features at the “left-hand side” and the “right-hand side” separately. An example is shown in fig. 10(b). Note that we also count \( h_{m_1} \) and \( h_{m_2} \). To avoid the situation that denominator may equal zero, we let \( h_{m_1} \) belongs to the “right-hand side” and \( h_{m_2} \) belongs to the “left-hand side” to keep the sum of the percentages correct.

The reason we only count the number of the features in a particular range totally agrees with the inter-direction similarity. We don’t want the features from other part in the pattern causing noise. Now we can compute the differences of the “left neighbor-rate” and the “right neighbor-rate” between the corresponding feature pairs, and obtain the inter-ratio similarity by computing the average of the above two differences. Together with other similarity measurements, we can further replace the equation (7) by the following equation:

\[
\mathcal{D}_2(\mathbf{r}_1^j, \mathbf{r}_2^j, \mathbf{h}_{m_1}, \mathbf{h}_{m_2}) = \begin{cases} 
(1 + \gamma)(1 + \lambda_1) \times \bar{\mathcal{D}}_1, & \text{if } \Psi^j(l_1, l_2) = 1 \text{ and } Y(m_1, m_2) = 1 \\
-\mu, & \text{if } \mathcal{D}_0(\mathbf{r}_1^j, \mathbf{r}_2^j) > \delta \text{ or } \mathcal{D}_0(\mathbf{h}_{m_1}, \mathbf{h}_{m_2}) > \delta \\
-(\mu + \lambda_2), & \text{otherwise.}
\end{cases}
\]

where \( \gamma \) specifies the inter-ratio similarity and we set \(-0.5 \leq \gamma \leq 0.5 \) in our simulations. Note that our design mostly depends on the inter-link similarity. The reason is that in our application most features in a chosen radical are connected. Moreover, a radical in a pattern is always in a specific range. The inter-link similarity will make the matching result satisfy the above property. We have tried separating our new similarity measurements from the inter-link similarity in the motion equation of the Hopfield network, but the matching results we got were not good enough, so we choose to combine the inter-direction similarity and the inter-ratio similarity with the inter-link similarity. Fig. 11 gives the improved result of the example in fig. 6 after adding the two new similarity measurements.

3.2. Weighted bended-ellipse feature

In the last subsection we have made some improvement based on the inter-link similarity. Here, we make some changes about the inter-feature similarity. Let us begin by taking a look at the bended-ellipse features closely. Recall that a bended-ellipse feature is represented by a five-dimensional vector. For a given feature \( \mathbf{r}_i^j = [x_i^j, y_i^j, u_i^j, \phi_i^j \hat{x}_i^j, \phi_i^j \hat{y}_i^j] \) of a radical, the first
two elements \( x^j_l, y^j_l \) indicate the geometric position of \( r^j_l \), the third element \( u^j_l \) indicates the size of \( r^j_l \) and the last two elements indicate the shape of \( r^j_l \).

We have discussed in the previous section that in Chinese characters some radicals may have similar shapes but they differ in size or location. The original feature-to-feature adhesion method suggests that we regard them as different template radicals. Fig. 1 has shown some simple examples, but in fact, Chinese characters are much more complex than these examples. Fig. 12 gives some complex examples to indicate this. Sometimes it is hard for us to list all the possibilities of the locations and the sizes that a radical appears in all kinds of patterns. It may also waste a lot of time on computing the compatibilities between the unknown pattern and such a large amount of template radicals.

Our method for solving the above problem is quite easy: for a chosen template radical, we only choose an appropriate amount of variations of this radical. We observed that although a chosen radical may have large variations in sizes and locations in different patterns, their shapes are roughly the same. So we change the weight of the elements in the bended-ellipse features in order to reduce the influence from the size and the location. More specifically, we define \( \vec{r}^j_l = [\omega_1 x^j_l, \omega_1 y^j_l, \omega_2 u^j_l, \omega_3 \phi^j_l x^j_l, \omega_3 \phi^j_l y^j_l] \) to be the weighted bended-ellipse feature of the original feature \( r^j_l \). By setting the weight of each element in the feature, we can emphasize or reduce the importance of particular characteristics in computing the inter-feature similarity. Fig. 13 gives the result of using the weighted bended-ellipse features and here we set \( \omega_1 = 0.1, \omega_2 = 0.1, \omega_3 = 1 \). We can find that the matching result does not become incorrect; in fact, it looks even better in shape.
Figure 13. The matching result of using the weighted bended-ellipse features. Note that the feature at the branch is mostly matched according to the shape of the radical.

Figure 14. Weighted bended-ellipse features provide the “local match” effect.

In practice we set $\omega_1 = 1$, $\omega_2 = 1$, the value of $\omega_3$ is between 1 and 4. We do not set
an overgreat value to $\omega_3$ because there are a lot of similar shapes in Chinese characters. If we only focus on the shape of the features, we will get a wrong matching result with high probability. In fact, letting $\omega_3$ be slightly larger than $\omega_1$ and $\omega_2$ is a balanced choice. Each feature will automatically match a similar feature in a local range. Fig. 14 shows an example that the same radical in different locations matches to different parts of a pattern. Note that under our design, the other similarity measurements can be used to make up the deficiency of the location and the size information. The inter-link similarity tells us that if there is a link between two features, then these two features could not be too far. The size of the shape is then restricted indirectly. The inter-direction similarity gives the relatively direction for each feature as we having discussed earlier. The inter-ratio similarity also provides some location information. For example, if the “right neighbor rate” of two neighbor features is small, these two features may have a higher chance to be on the boundary. Using the weighted bended-ellipse features with these similarity measurements, we can reduce the number of radicals in our database (so as to save computational time) without a loss in the matching accuracy.

4. Applications and discussion

We have introduced some new geometric relations for improving the original feature-to-feature adhesion method in the last section. In this section, we will discuss some possible applications of our method. Let us begin with an easy application. The recognition of the vehicle registration plate is a common application in our life. The numbers and the letters on the license plate are regular and not complex; moreover, the radicals are not many, so the recognition is quite easy. The only thing we should take care of is making sure we choose radicals in different viewpoints to ensure a correct match. Fig. 15 gives a simple example of the matching result of a license plate. Here we can use the unweighted bended-ellipse features to speed up the process. The same setting can be used to solve the recognition of the postal codes.

![Figure 15. The matching result of a license plate.](image)

A more complex application is the recognition of the music score. A music score is much more complex than a character because there are a lot of similar shapes in it. The complicated composition of the music score can also be a big problem. Fig. 16 shows a small example. We use the weighted bended-ellipse features in this application and we focus on the location to discriminate the musical notations. We obtained a correct matching result from a complex pat-
tern. The same approach can be applied to some similar applications such as the recognition of the circuit diagrams or kinds of pipeline diagrams.

The feature-to-feature adhesion method can not only find the matched pattern, but it can also find the differences of two similar patterns. The process is exactly the same as with the previous discussion. Given two patterns we want to compare, we find the bended-ellipse features and run the Hopfield network to find the matching result. Then we just list the inter-feature similarities of the matched feature pairs. Note that we do not need to compute anything more because the inter-feature similarities are prepared in the previous process. Fig. 17 lists the similarity of each matched feature pair of a previous example. The negative values at the right show the differences. A smaller value indicates a larger difference between two features. If one feature does not match any other feature, we set the inter-feature similarity value of this feature to be infinitely small. Fig. 18 gives another example, where we compare two contour maps of Taiwan. Recently, the greenhouse effect has caused a rising of the sea levels. Because of this and through using our geometrical relations, we can find changes of the contour of a coastline. This same process can also be used to find defects in a chip or other similar applications.
Figure 18. The difference of two similar contours is found by our method.

5. Summary

The original feature-to-feature adhesion method uses two similarity measurements to match features to recognize of the handprinted characters. Through this work, we have constructed two new rules based on the geometrical relations of the bended-ellipse features. We add them into the Hopfield model to improve the matching result. In addition, we introduced the concept of the weighted bended-ellipse features, which allow us to focus on specific characteristics of the features. With this generalized definition, we can further improve the matching result and apply our method to more applications. We also provided new kinds of applications in this paper. The same approach can be applied to many other pattern recognition tasks, as long as the pattern can be divided into small components.

References


