國立臺灣大學電機資訊學院資訊工程學系
碩士論文
Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

應用霍氏模型記憶幾何規則

Finite Geometrical Relations Loading in Hopfield Model

施鴻逸
Hong-Yi Shih

指導教授：劉長遠 博士
Advisor: Cheng-Yuan Liou, Ph.D.

中華民國 101 年 9 月
September, 2012

致謝

首先我要感謝我的指導教授劉長遠老師，老師在我做研究的期間除了學問上的指導以外更給我許多方向，讓我有可以一直不斷努力的目標。也要感謝呂育道教授以及黃昭綺學姊與鄭為正學長在口試期間給我許多的建議，讓我可以繼續改進我的研究成果。

另外我要感謝跟我一起努力的同學曾聖翰，在研究期間互相幫忙解決問題，也互相鼓勵，謝謝你跟我一起討論很多程式上的問題。

在資工系也有不少的同學與學長姐、學弟妹，在我寫論文的期間給我許多支持。其中特別感謝佳慶學長給我的建議，以及感謝顯之學弟在我研究的時候常常空出週末跟我一起討論。希望大家不管正在工作還是仍在為論文奮鬥都能一切順利。

另外還要感謝我的社團星韻合唱團的許多好朋友，不論何時在生活上研究上碰上壓力，這邊總是充滿歡笑，和大家一起唱歌聚會能讓我放鬆心情然後繼續出發，星韻的好夥伴們是我最大的動力來源。

最後當然要感謝我的父母親，在我研究遇上瓶頸時給了我最大的支持與諒解，讓我可以轉換心情重新出發，才能在剩餘不多的學生生涯中做出今天的成果，謝謝你們一直以來的包容與鼓勵。

摘要

本論文為手寫字辨認設計兩項新的幾何規則，並將原有的特徵表示法做推廣，在應用霍氏模型的方法中得到較好的配對結果。除了手寫字辨認以外，我們也提出許多符合現今需求的新應用，本論文所提供的方法能夠繼續發展以解決這些更複雜的應用。

曲橢圓特徵模型適合用來表現複雜的幾何型態，這些特徵點在幾何上擁有許多規則，我們可以應用霍氏模型記憶這些規則以達成特徵點的配對。設計更多不同的幾何規則能提升識別率，也能解決許多不同類型的型態辨識問題。

關鍵字：手寫字辨識、樣式辨識、霍氏模型、幾何規則、曲橢圓特徵

Abstract

There are two similarities constructed in earlier works [1], [2], named inter-feature similarity and inter-link similarity. This work rewrites geometrical relations [3] and constructs them into Hopfield model to improve the matching result.

Constructing a set of bended-ellipse features is an efficient way for sampling cursive patterns. It converts a pattern into structural features and provides a natural way for handprinted characters recognition. The idea which called feature-to-feature adhesion constructs a topological configuration for a pattern and transforms it into an undirected graph. The compatibility associated with the graph is then formulated as an optimization problem and is solved by a devised Hopfield network.

Keyword : Handprinted character recognition, Pattern recognition, Hopfield model, Geometrical relation, Bended-ellipse feature

Contents

List of Figures

Chapter 1

Introduction

Handprinted character recognition is an important application in our life. Many of technology products nowadays such as smart phone and tablet PC allow users use handwriting as input. The recognition system for license plate or the postal codes are also the familiar applications of the handprinted character recognition system.

There are several different methods to accomplish handprinted 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 are relatively easy. However, most of those methods have a common defect that they usually require precise extraction of features and radicals [4], [5], [6]. In the feature-to-feature adhesion method developed by Liou and Yang [7], 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. Then use a Hopfield network [8] to solve a maximization problem collectively, and use neural network to complete the classification of characters finally. This method can be applied to not only character recognition, but also many other pattern recognition tasks while the pattern can be divided into small components.

In order to generate the features for template radicals and the patterns, Liou and Yang use the bended-ellipse features [9]. Each feature of a radical or a pattern is represented by a five dimensional vector which including 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 define 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 handprinted 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 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 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 applications of this method by reuse the calculated similarities.

The rest of the paper is organized as follows. In chapter 2 we introduce the feature-to-feature adhesion method, including bended-ellipse features, FTF order, and the details

of the classification. In chapter 3 we give new conditions and changes for classifying improvement. Chapter 4 discusses some new applications and shows our results. Chapter 5 provides summary and conclusion.

Chapter 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 [9]. 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 characters. Fig. 2.1 shows two examples 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 characters. The shapes of these radicals are relatively easy and we want to decompose them into small features.

We choose L template radicals $\mathbf{R}^1, \mathbf{R}^2, \ldots, \mathbf{R}^L$. Each radical is composed of a set of features. For a given radical $\mathbf{R}^j$, $1 \leq j \leq L$, we will generate L^j bended-ellipse features.

Figure 2.1: Chinese charaters with radicals in different locations or sizes.

So we can write $\mathbf{R}^j = \{\mathbf{r}_l^j | 1 \leq l \leq L^j\}$, where r_l^j is the lth feature vector of the jth template radical.

Now we discuss the method used to generate the bended-ellipse feature fields. Given a radical $\mathbf{R}^j$, we choose $\mathcal{L}^j$ seeds p_ℓ^j, $1 \leq \ell \leq \mathcal{L}^j$. p_ℓ^j 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 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. 2.2(a) shows an example of finding MSD lines of a given seed. The two bold lines in the example are the MSD lines. They are roughly along the strokes of the character.

Define $\bar{L}_\ell^j(\alpha)$ be the line segment begins from seed p_ℓ^j along direction α 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

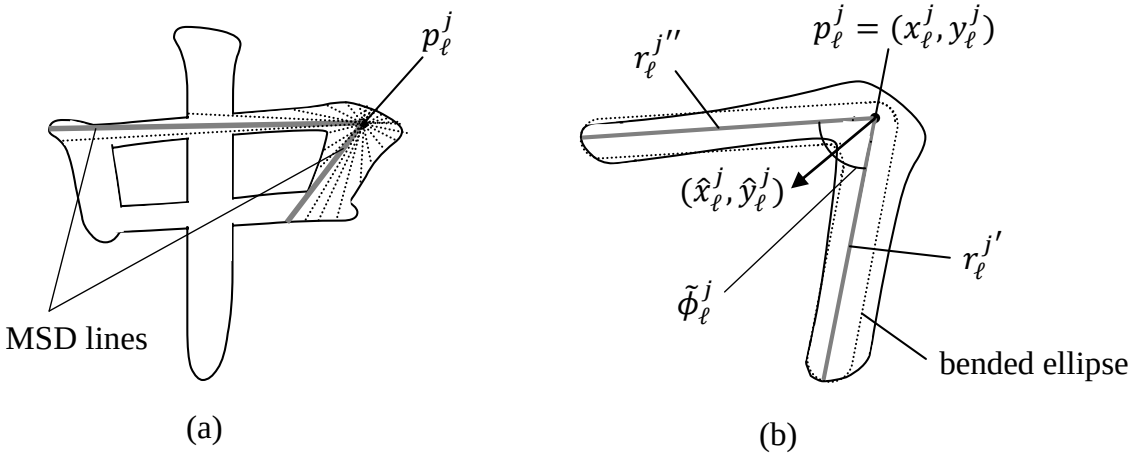


Figure 2.2: (a) The MSD lines. (b) The bended-ellipse feature (five dimensional vector).

find the MSDs of p_ℓ^j:

Direction α is a MSD if

$$\frac{1}{2w+1}\sum_{q=-w}^{w}|\bar{L}_\ell^j(\alpha+q\cdot\frac{\pi}{36})| > \frac{1}{2w+1}\sum_{q=-w}^{w}|\bar{L}_\ell^j(\alpha+(q+o)\cdot\frac{\pi}{36})| \quad \text{and}$$
$$\frac{1}{2w+1}\sum_{q=-w}^{w}|\bar{L}_\ell^j(\alpha+q\cdot\frac{\pi}{36})| > \frac{1}{2w+1}\sum_{q=-w}^{w}|\bar{L}_\ell^j(\alpha+(q-o)\cdot\frac{\pi}{36})|$$
$$\text{for} \quad o=1,...,O \tag{2.1}$$

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 α is an MSD, we call $\bar{L}_\ell^j(\alpha)$ an MSD line. Fig. 2.2(a) shows two MSD lines.

After obtaining the MSD lines, we are ready for generating bended-ellipse features. Recall that we picked $\mathcal{L}^j$ seeds from the jth template radical $\mathbf{R}^j$ and we will generate L^j bended-ellipse features $\{\mathbf{r}_l^j | 1 \le l \le L^j\}$. Note that each seed may create more than one feature. So we have $L^j \ge \mathcal{L}^j$. These bended-ellipse features can be represented by five-dimensional vectors as follows. Let $\mathbf{r}_l^j = [x_l^j, y_l^j, u_l^j, \phi_l^j\hat{x}_l^j, \phi_l^j\hat{y}_l^j]$ be the five-dimensional vector. Without loss of generality we may assume that $\mathbf{r}_l^j$ is generated from the seed p_ℓ^j.

The elements x_l^j and y_l^j in $\mathbf{r}_l^j$ are the coordinates of seed p_ℓ^j. Let $r_l^{j\prime}$ and $r_l^{j\prime\prime}$ be the lengths of the two MSD lines. We set

$$u_l^j = \frac{r_l^{j\prime} + r_l^{j\prime\prime}}{2}.$$

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

$$\phi_l^j = \frac{\pi - \tilde{\phi}_l^j}{\pi}.$$

Let $(\hat{x}_l^j, \hat{y}_l^j)$ be the unit vector extends from the seed p_ℓ^j and equally divides the angle $\tilde{\phi}_l^j$. Now every element in $\mathbf{r}_l^j$ is well-defined. Fig. 2.2(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 $\tilde{\phi}_l^j = 0$ is a degenerated ellipse that we will discard it. A feature is also defined as a defect if it has a small arm of the arm ratio is not uniform. More precisely, we will use the ellipses which satisfy $r_l^{j\prime} > r^*$, $r_l^{j\prime\prime} > r^*$ and $0.5 \leq r_l^{j\prime}/r_l^{j\prime\prime} \leq 2$, 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

$$\begin{pmatrix} n_a \\ 2 \end{pmatrix}$$

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. 2.3 gives some examples for indicating that. We will group the features generated from the same seed to simplify our later processes.

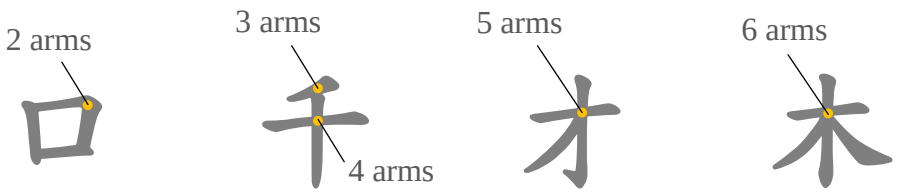


Figure 2.3: A seed with different arm number can generate more than 2 bended-ellipses.

We have discussed how to generate the bended-ellipse features. After applying the method to all the template radicals, we get $\mathbf{R}^j = \{\mathbf{r}_l^j | 1 \leq l \leq L^j\}$ for $1 \leq j \leq L$. We then apply the same process to every of N template patterns to obtain $\mathbf{S}^i = \{\mathbf{s}_n^i | 1 \leq n \leq N^i\}$ for $1 \leq i \leq N$, where $\mathbf{s}_n^i = [x_n^i, y_n^i, u_n^i, \phi_n^i \hat{x}_n^i, \phi_n^i \hat{y}_n^i]$ is the nth feature vector of the ith template pattern. For an unknown handprinted pattern input, we also apply the same process to it and we get $\mathbf{H} = \{\mathbf{h}_m | 1 \leq m \leq M\}$, where $\mathbf{h}_m = [x_m, y_m, u_m, \phi_m \hat{x}_m, \phi_m \hat{y}_m]$ is the mth 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 [10], [11]. Fig. 2.4 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 block. With these marks, the topological structure can be reconstructed correctly. Fig. 2.4(a) shows the reconstruction of the character and Fig. 2.4(b) gives the matrix representation of the feature-to-feature order.

The computation of the FTF order for bended-ellipse features is similar to the concept

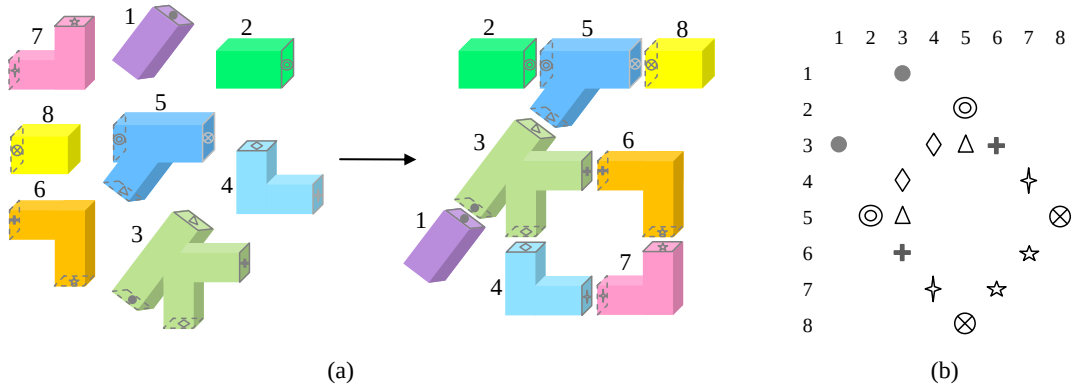
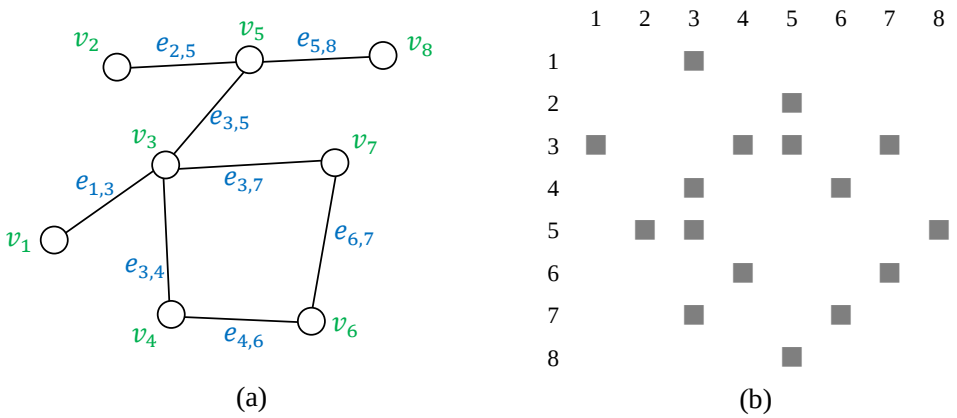


Figure 2.4: (a) The scattered blocks. (b) Matrix representation of the topological order.

of the cell-to-cell adhesion mechanism but simpler. In our method the features are similar to the blocks in Fig. 2.4(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 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. 2.5 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. 2.5(a) stands for a seed.

We apply this method to every template radical $\mathbf{R}^j$, $1 \leq j \leq L$. For each $\mathbf{R}^j$, the FTF order we obtained is represented by an $L^j \times L^j$ matrix Ψ^j. Formally, define $\Psi^j(l_1, l_2)$ to be the element in the l_1th row and l_2th column of matrix Ψ^j. And define

$$\Psi^j(l_1, l_2) = \begin{cases} 1, & \text{if there is a link between } \mathbf{r}^j_{l_1} \text{ and } \mathbf{r}^j_{l_2} \\ 0, & \text{otherwise.} \end{cases}$$



(a) (b)

Figure 2.5: (a) Graph representation of the FTF order. (b) Matrix representation of the FTF order.

The same process is applied to the template patterns and unknown handprinted pattern. We use an $N^i \times N^i$ matrix Γ^i to represent the FTF order of the ith template pattern $\mathbf{S}^i$ for $1 \leq i \leq N$ and an $M \times M$ matrix Υ to represent the FTF order of the unknown handprinted pattern $\mathbf{H}$. Fig. 2.5(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 handprinted character. Before we discuss the method of the classification, we list the information we have known.

- L template radicals $\mathbf{R}^j = \{\mathbf{r}_l^j | 1 \leq l \leq L^j\}$, $1 \leq j \leq L$ and Ψ^j for each $\mathbf{R}^j$.
- 1 unknown handprinted pattern $\mathbf{H} = \{\mathbf{h}_m | 1 \leq m \leq M\}$ and Υ for $\mathbf{H}$.
- N template patterns $\mathbf{S}^i = \{\mathbf{s}_n^i | 1 \leq n \leq N^i\}$, $1 \leq i \leq N$ and Γ^i for each $\mathbf{S}^i$.

We will compare the compatibility of every $\mathbf{R}^j$ with $\mathbf{H}$ and every $\mathbf{S}^i$, then choose the most likely template pattern to be the classification result. Formally speaking, let c_j^h be the compatibility between the jth radical $\mathbf{R}^j$ and the handprinted pattern $\mathbf{H}$. We will discuss the method of computing the compatibility c_j^h later. Let $\vec{\mathbf{c}}^h = [c_1^h \dots c_L^h]^T$ be the compatibility vector specifies the overall compatibility between $\mathbf{H}$ and all radicals $\mathbf{R}^1 \dots \mathbf{R}^L$. For each template pattern $\mathbf{S}^i$, $1 \leq i \leq N$, we apply the same process to compute the overall compatibility between $\mathbf{S}^i$ and all radicals $\mathbf{R}^1 \dots \mathbf{R}^L$. We represent this overall compatibility by a vector $\vec{\mathbf{c}}^i = [c_1^i \dots c_L^i]^T$.

Note that the only things we need to compute on-line is the vector $\vec{\mathbf{c}}^h$. The compatibilities between every $\mathbf{S}^i$ and every $\mathbf{R}^j$ can be pre-computed off-line, which means that the vectors $\vec{\mathbf{c}}^i$ for $1 \leq i \leq N$ can be prepared in advance. Once we obtained $\vec{\mathbf{c}}^h$, we can compute the dissimilarity between the handprinted pattern and every standard pattern to get the classification result. This dissimilarity, denoted by $\mathbf{D}(\vec{\mathbf{c}}^h, \vec{\mathbf{c}}^i)$, $1 \leq i \leq N$, is simply defined as

$$\mathbf{D}(\vec{\mathbf{c}}^h, \vec{\mathbf{c}}^i) = \|\vec{\mathbf{c}}^h - \vec{\mathbf{c}}^i\|.$$

We use the template pattern which minimizes the Euclidean distance $\mathbf{D}(\vec{\mathbf{c}}^h, \vec{\mathbf{c}}^i)$ as our classification result.

Now we introduce the method of computing the compatibility between a radical and a pattern. We begin this by discussing how to measure the compatibility between the 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 $(\mathbf{r}_{l_k}^j, \mathbf{h}_{m_k})$, where

$\mathbf{r}^j_{l_k} \in \mathbf{R}^j$ and $\mathbf{h}_{m_k} \in \mathbf{H}$, be one feature pair. We define the inter-feature similarity as

$$\mathcal{D}_1(\mathbf{r}^j_{l_k}, \mathbf{h}_{m_k}) = -\|\mathbf{r}^j_{l_k} - \mathbf{h}_{m_k}\|.$$

The Euclidean distance indicates the similarity of these two features. The bigger value of $\mathcal{D}_1$ 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 $(\mathbf{r}^j_{l_1}, \mathbf{h}_{m_1})$ and $(\mathbf{r}^j_{l_2}, \mathbf{h}_{m_2})$ be the two corresponding feature pairs. We say that these two feature pairs have high inter-link similarity if

$$\Psi^j(l_1, l_2) = 1 \text{ and } \Upsilon(m_1, m_2) = 1.$$

That is, there is a link between $\mathbf{r}^j_{l_1}, \mathbf{r}^j_{l_2}$ and there is a link between $\mathbf{h}_{m_1}, \mathbf{h}_{m_2}$. Together with these two similarities, we can define the compatibility between the two corresponding feature pairs $(\mathbf{r}^j_{l_1}, \mathbf{h}_{m_1})$ and $(\mathbf{r}^j_{l_2}, \mathbf{h}_{m_2})$ as follows:

$$\mathcal{D}_2(\mathbf{r}^j_{l_1}, \mathbf{r}^j_{l_2}, \mathbf{h}_{m_1}, \mathbf{h}_{m_2}) = \begin{cases} \frac{\mathcal{D}_1(\mathbf{r}^j_{l_1}, \mathbf{h}_{m_1}) + \mathcal{D}_1(\mathbf{r}^j_{l_2}, \mathbf{h}_{m_2})}{2}, & \text{if } \Psi^j(l_1, l_2) = 1 \text{ and } \Upsilon(m_1, m_2) = 1 \\ -\mu, & \text{otherwise.} \end{cases} \tag{2.2}$$

where μ 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,

$$\mathcal{D}_2(\mathbf{r}^j_{l_1}, \mathbf{r}^j_{l_2}, \mathbf{h}_{m_1}, \mathbf{h}_{m_2}) \neq \mathcal{D}_2(\mathbf{r}^j_{l_2}, \mathbf{r}^j_{l_1}, \mathbf{h}_{m_1}, \mathbf{h}_{m_2})$$

and

$$\mathcal{D}_2(\mathbf{r}^j_{l_1}, \mathbf{r}^j_{l_2}, \mathbf{h}_{m_1}, \mathbf{h}_{m_2}) \neq \mathcal{D}_2(\mathbf{r}^j_{l_1}, \mathbf{r}^j_{l_2}, \mathbf{h}_{m_2}, \mathbf{h}_{m_1}).$$

It is straightforward to compute the total compatibility between a radical and a pattern. The way is just summing 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. 2.5(a)). Let the graph of the jth radical $\mathbf{R}^j$ be G^j and the graph of the handprinted pattern $\mathbf{H}$ 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 η^j to denote the matched features. Formally we define $\eta^j = \{(\mathbf{r}_l^j, \mathbf{h}_{m_l}) | 1 \leq l \leq L^j\}$, where $\mathbf{h}_{m_l}$ is matched to $\mathbf{r}_l^j$. If the route η^j has been decided, the total compatibility between $\mathbf{R}^j$ and $\mathbf{H}$ can be computed using the equation in (2.2). 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:

$$\text{Maximize} \quad \sum_{\substack{1 \leq l_1, l_2 \leq L^j \\ 1 \leq m_1, m_2 \leq M}} \mathcal{D}_2(\mathbf{r}_{l_1}^j, \mathbf{r}_{l_2}^j, \mathbf{h}_{m_1}, \mathbf{h}_{m_2}) \tag{2.3}$$

This optimization problem can be solved by a Hopfield network [8]. There are some other works used a Hopfield network to solve the pattern recognition problem without involving the FTF order [12], [13]. The feature-to-feature adhesion method we used gives us a topological intuition to formulate the energy function.

We use a connectivity matrix V to represent a route in the Hopfield network. V is defined as an L^j by M matrix, where the lth row stands for the feature $\mathbf{r}_l^j$ of the radical $\mathbf{R}^j$, and the mth column stands for the feature $\mathbf{h}_m$ of the handprinted pattern $\mathbf{H}$. 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 lth row and the mth column. The value of V_{lm} is either 1 or 0. $V_{lm} = 1$ means that $\mathbf{h}_m$ is matched to $\mathbf{r}_l^j$; otherwise, $V_{lm} = 0$. We will check the state of V after the network converged and add $(\mathbf{r}_l^j, \mathbf{h}_m)$ to η^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 (2.3). 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 above constraints:

$$E = \frac{A}{2}\sum_{l}\sum_{m_1}\sum_{m_2\neq m_1} V_{lm_1}V_{lm_2} + \frac{B}{2}\sum_{m}\sum_{l_1}\sum_{l_2\neq l_1} V_{l_1m}V_{l_2m} + \frac{C}{2}\sum_{l}(\sum_{m} V_{lm} - 1)^2$$
$$-\frac{D}{2}\sum_{l}\sum_{l_1\neq l}\sum_{m}\sum_{m_1\neq m} \mathcal{D}_2(\mathbf{r}_l^j, \mathbf{r}_{l_1}^j, \mathbf{h}_m, \mathbf{h}_{m_1})V_{lm}V_{l_1m_1} \tag{2.4}$$

where A, B, C, and D are constants to be decided. Here the first three terms in (2.4) are designed to make sure that the matrix V satisfies the first and the second rules [14], and the last term in (2.4) 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

$$V_{lm}^{(t)} = \frac{1}{2}(1 + \tanh(\mathrm{v}_{lm}^{(t)}/\mathrm{v}_0))$$
$$\mathrm{v}_{lm}^{(t)} = \mathrm{v}_{lm}^{(t-1)} + \frac{\partial \mathrm{v}_{lm}^{(t)}}{\partial t} \tag{2.5}$$

where v_0 is a constant. The last term in (2.5) is the equation of motion which changes the state of the network. This motion equation is defined as

$$\frac{\partial \mathrm{v}_{lm}}{\partial t} = -\frac{\mathrm{v}_{lm}}{\tau} - A\sum_{m_1\neq m} V_{lm_1} - B\sum_{l_1\neq l} V_{l_1m} - C(\sum_{m} V_{lm} - 1)$$
$$+D\sum_{l_1\neq l}\sum_{m_1\neq m} \mathcal{D}_2(\mathbf{r}_l^j, \mathbf{r}_{l_1}^j, \mathbf{h}_m, \mathbf{h}_{m_1})V_{l_1m_1} \tag{2.6}$$

where τ is a constant.

We need set the initial state of V. A reasonable thought is that if two features are matched, they should have relatively higher inter-feature similarity. So we use the inter-feature similarity as the criterion to set the initial value of each V_{lm}. More precisely, we set $V_{lm}^{(0)} = 1$ if $\mathcal{D}_1(\mathbf{r}_l^j, \mathbf{h}_m) > \epsilon$, where ϵ is the threshold. Else, we set $V_{lm}^{(0)} = 0$. We use $\epsilon = 0.5$ in practice. Besides, refer to the work from Aiyer et al. [15], we set $A = 500$, $B = 500$, $C = 500/\widetilde{N}$, and $D = 500\widetilde{N}/80$, where $\widetilde{N} = L^j M$ in our work. Fig. 2.6 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 for checking 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.

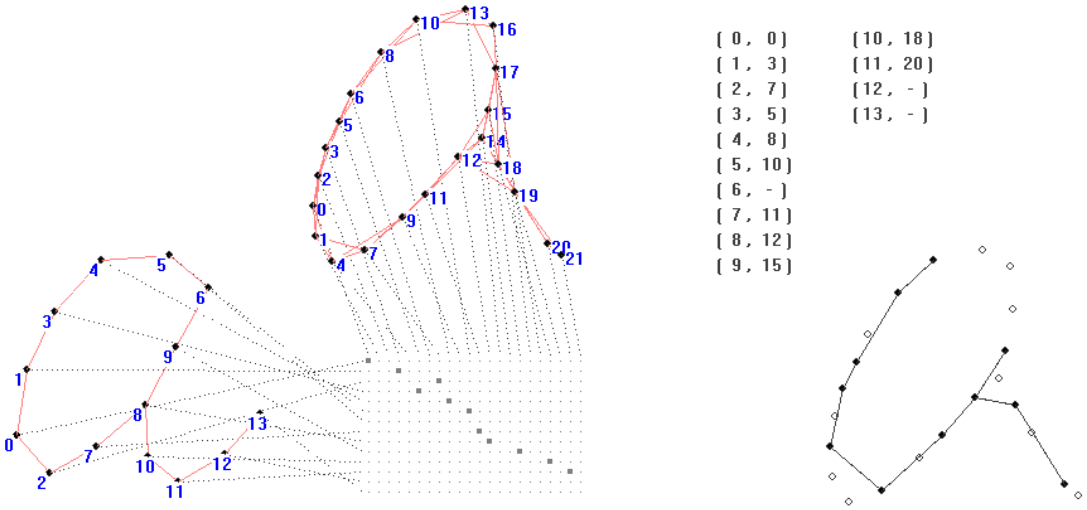


Figure 2.6: A converged result of the Hopfield network.

After running the Hopfield network for the unknown pattern and every template pattern, we get the overall compatibility vectors $\vec{\mathbf{c}}^h = [c_1^h \dots c_L^h]^T$ and $\vec{\mathbf{c}}^i = [c_1^i \dots c_L^i]^T$ for $1 \leq i \leq N$. For simpler pattern, we use the pattern $\mathbf{S}^i$ which minimizes $\mathbf{D}(\vec{\mathbf{c}}^h, \vec{\mathbf{c}}^i)$ as our classification result. If the patterns are more complex, we will use a four-layer back-

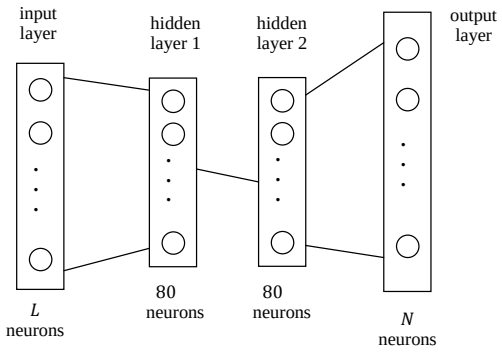


Figure 2.7: The concept of the four-layer backpropagation network.

propagation network for the classification [16]. Fig. 2.7 shows the network we use. This network can speed up by using the method by Liou and Yu [2]. We use $\vec{\mathbf{c}}^i$, $1 \leq i \leq N$ as input to train the network, and obtain the classification result when the network converged.

Chapter 3

Our improve

We have discussed the original feature-to-feature adhesion method. In the original method, Liou and Yang [7] 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

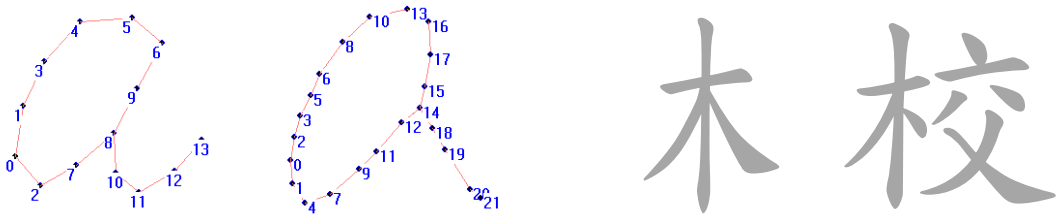
Liou and Yang [7] used only the inter-feature similarity and the inter-link similarity in their classification step. There are a lot of other relations among features that we can use for classifying. We now discuss two new similarity measurements. First we note that the inter-link similarity indicates whether there is a link between two features or not, but it can not 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 directs of the corresponding feature pairs. Recall that $(\mathbf{r}_{l_1}^j, \mathbf{h}_{m_1})$ and $(\mathbf{r}_{l_2}^j, \mathbf{h}_{m_2})$ are the

two corresponding feature pairs, where $\mathbf{r}_{l_k}^j \in \mathbf{R}^j$ and $\mathbf{h}_{m_k} \in \mathbf{H}$ for $k = 1, 2$. The concept of inter-direction similarity is easy. For example, if $\mathbf{r}_{l_2}^j$ is at the right of $\mathbf{r}_{l_1}^j$ and $\mathbf{h}_{m_2}$ is at the right of $\mathbf{h}_{m_1}$, we say they have high inter-direction similarity. Here we use the eight cardinal directions for our classification. Fig. 3.1 shows the eight directions we use. In this example $\mathbf{h}_{m_2}$ is at the north-east of $\mathbf{h}_{m_1}$. We will compute the inter-direction similarity for the corresponding feature pairs according to the angle difference of two directions.

Figure 3.1: The concept of the direction of two features.

For a given feature $\mathbf{r}_l^j$, the other features will be classified into three parts according to how far they are from $\mathbf{r}_l^j$. The first part are the features which there is a link between $\mathbf{r}_l^j$ and each of them. There is a link means that 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. 3.2(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 relative direction of them may have higher inaccuracies. We should decrease their influence in computing the inter-direction similarity.

For the feature $\mathbf{r}_l^j$, those features which are too far away from $\mathbf{r}_l^j$ are not good reference resources for measuring the inter-direction similarity either. This is because that these features may not belong to the same radical in the pattern in most cases. Fig. 3.2(b) is a common example in Chinese character. The template pattern at right is composed of the



(a) (b)

Figure 3.2: (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.

radical at left and another part. Considering the relative directions between the features at left and the features at right will provide us 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 $\mathbf{r}_l^j$. These features are good candidates for providing the direction information. Note that in fig. 3.2(a) the tails of two patterns are different, but for those features at the top-middle of the characters, 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 making up these ideas together, we can replace the equation (2.2) by the following equation:

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

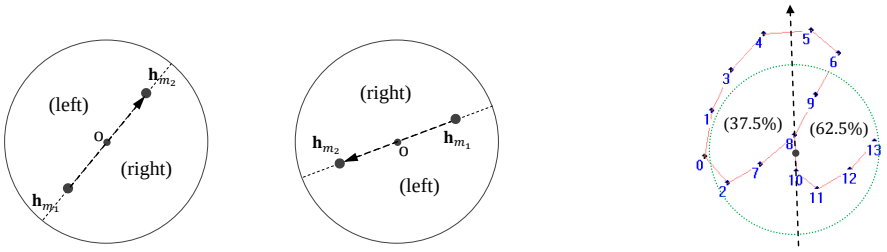
$$\bar{\mathcal{D}}_1 = \frac{\mathcal{D}_1(\mathbf{r}_{l_1}^j, \mathbf{h}_{m_1}) + \mathcal{D}_1(\mathbf{r}_{l_2}^j, \mathbf{h}_{m_2})}{2}. \tag{3.1}$$

where $\mathcal{D}_0(\mathbf{r}_{l_1}^j, \mathbf{r}_{l_2}^j) = \|(x_{l_1}^j, y_{l_1}^j) - (x_{l_2}^j, y_{l_2}^j)\|$, $\mathcal{D}_0(\mathbf{h}_{m_1}, \mathbf{h}_{m_2}) = \|(x_{m_1}, y_{m_1}) - (x_{m_2}, y_{m_2})\|$ are the Euclidean distances between the location of the features, δ is a positive constant. Here λ_1 and λ_2 specify the inter-direction similarity, we set $-0.3 \leq \lambda_1 \leq 0.3$ and $-0.5 \leq$

$\lambda_2 \leq 0.5$ in our simulations. The larger value means that the corresponding feature pairs are not similar in directions. Note that our design will reinforce the difference of the compatibility, so as to improve the result of running the Hopfield network.

Another similarity measurement we add 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 them. 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, $\mathbf{h}_{m_1}, \mathbf{h}_{m_2}$. We find their topological midpoint o, then count the number of features in a circle 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. 3.3(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. 3.3(b). Note that we also count $\mathbf{h}_{m_1}$ and $\mathbf{h}_{m_2}$ to avoid the situation that denominator may equals zero, we let $\mathbf{h}_{m_1}$ belongs to the "right-hand side" and $\mathbf{h}_{m_2}$ belongs to the "left-hand side" to remain the sum of the percentages is 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 above two differences. Together with



(a) (b)

Figure 3.3: (a) The concept of computing the "neighbor-rate." (b) An example of computing the "neighbor-rate."

other similarity measurements, we can further replace the equation (3.1) by the following equation:

$$\mathcal{D}_2(\mathbf{r}^j_{l_1}, \mathbf{r}^j_{l_2}, \mathbf{h}_{m_1}, \mathbf{h}_{m_2}) = \begin{cases} (1+\gamma)(1+\lambda_1)\bar{\mathcal{D}}_1, & \text{if } \Psi^j(l_1, l_2) = 1 \text{ and } \Upsilon(m_1, m_2) = 1 \\ -\mu, & \text{if } \mathcal{D}_0(\mathbf{r}^j_{l_1}, \mathbf{r}^j_{l_2}) > \delta \text{ or } \mathcal{D}_0(\mathbf{h}_{m_1}, \mathbf{h}_{m_2}) > \delta \\ -(\mu+\lambda_2), & \text{otherwise.} \end{cases} \tag{3.2}$$

where γ specify the inter-ratio similarity, 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 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 result we got is not good enough, so we choose to combine the inter-direction similarity and the inter-ratio similarity with the inter-link similarity. Fig. 3.4 gives the improved result of the example in fig. 2.6 after adding the two new similarity measurements.

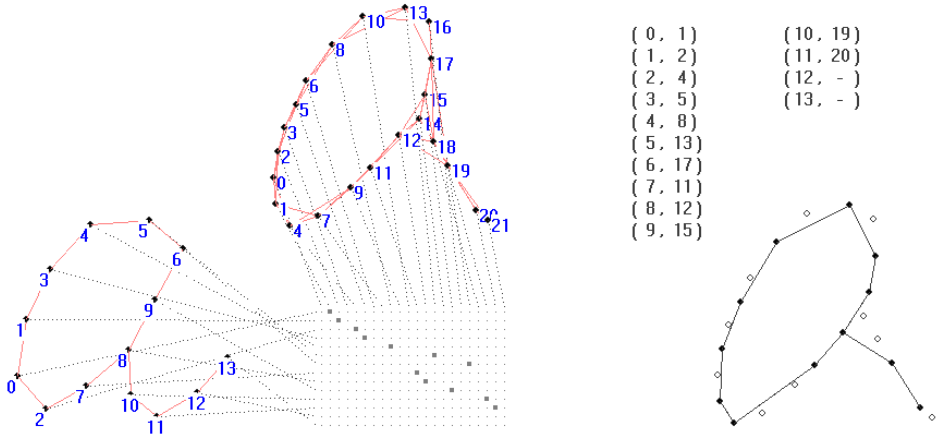


Figure 3.4: The improved result of running Hopfield network.

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 from 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 $\mathbf{r}_l^j = [x_l^j, y_l^j, u_l^j, \phi_l^j \hat{x}_l^j, \phi_l^j \hat{y}_l^j]$ of a radical, the first two elements x_l^j, y_l^j indicate the geomatric position of $\mathbf{r}_l^j$, the third element u_l^j indicate the size of $\mathbf{r}_l^j$ and the last two elements indicate the shape of $\mathbf{r}_l^j$.

We have discussed in the previous section that in Chinese characters some radicals may have similar shapes but different in sizes or locations. The original feature-to-feature adhesion method suggests that we regard them as different template radicals. Fig. 2.1 has shown some simple examples, but in fact Chinese characters are much more complex than these examples. Fig. 3.5 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 huge amount of template radicals.

口 圖噪靈器嘻嘔譴

Figure 3.5: More examples of Chinese charaters with the radical in different locations or sizes.

Our method for solving above problem is quite easy: for a chosen template radical, we only choose an appropriate amount for the variations of this radical. We observed that although a chosen radical may have large variations in sizes and locations in different patterns, their shape 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 $\hat{\mathbf{r}}_l^j = [\omega_1 x_l^j, \omega_1 y_l^j, \omega_2 u_l^j, \omega_3 \phi_l^j \hat{x}_l^j, \omega_3 \phi_l^j \hat{y}_l^j]$ to be the weighted bended-ellipse feature of the original feature $\mathbf{r}_l^j$. 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. 3.6 gives the result of using the weighted bended-ellipse features, 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 even looks better in shape.

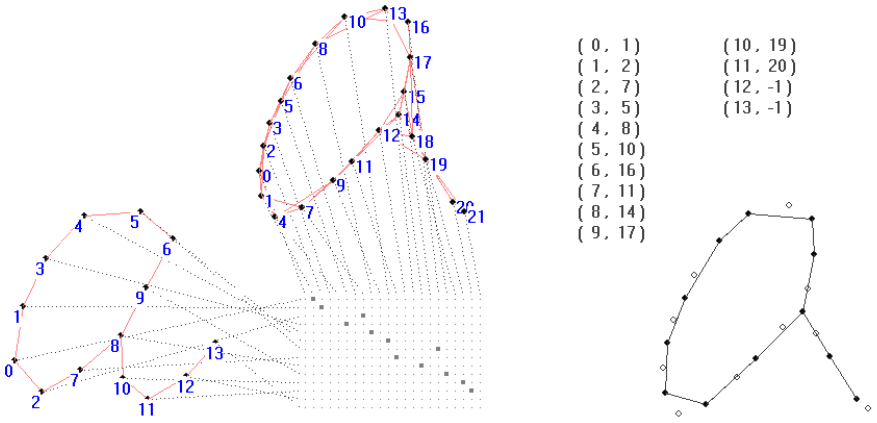


Figure 3.6: The matching result of using the weighted bended-ellipse features.

Figure 3.7: Weighted bended-ellipse features provide the "local match" effect.

In practice we set $\omega_1 = 1$, $\omega_2 = 1$, the value of ω_3 is between 1 and 4. We do not set an overgreat value to ω_3 because there are a lot of similar shape 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, let ω_3 be slightly larger than ω_1 and ω_2 is a balanced choice. Each feature will automatically match a similar feature in a local range. Fig. 3.7 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 the computation time) without loss the matching accuracy.

Chapter 4

Applications and discussion

We have introduced some new geometrical relations for improving the original feature-to-feature adhesion method in the last section. In this section we 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 is that we need to choose radicals in different viewpoints to ensure a correct matching. Fig. 4.1 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.

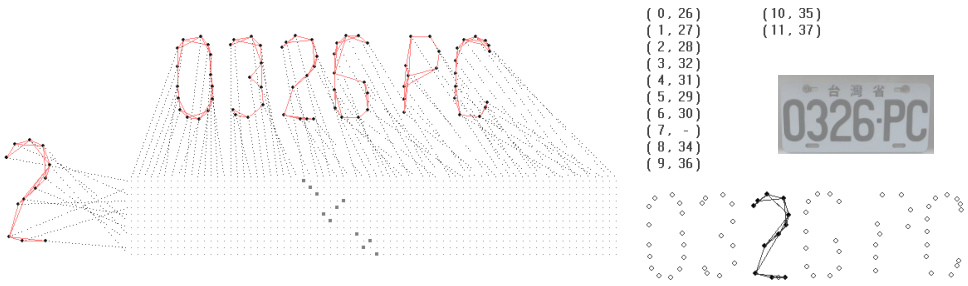


Figure 4.1: The matching result of a license plate.

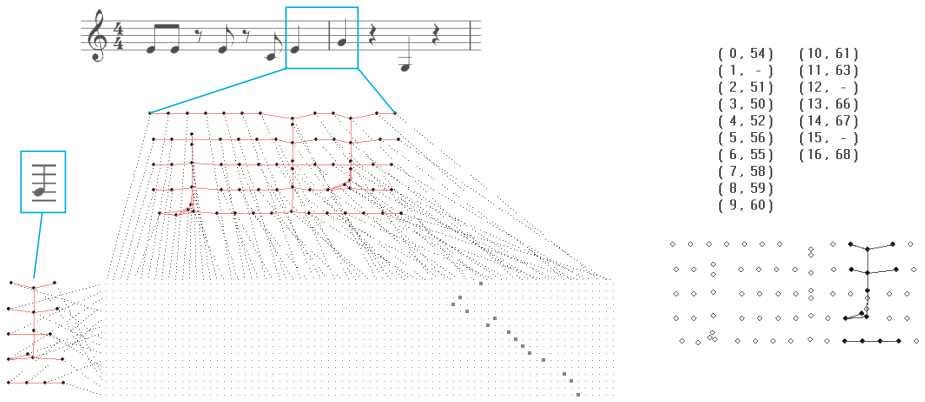


Figure 4.2: An example of the matching of the music score.

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. 4.2 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 pattern. 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, it can find the differences of two similar patterns also. The process is exactly the same with 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 previous process. Fig. 4.3 lists the similarity of each matched feature pair of previous example. The negative values at the right show the differences. The smaller value indicates the larger difference

between two features. If a feature does not match any feature, we set the inter-feature similarity value of this feature to be infinitely small. Fig. 4.4 gives another example that we compare two contours of drawing the map of Taiwan. The greenhouse effect causes the rising of the sea level nowadays. Using this method we can find the change of the contour of a coastline. The same process can also be used for finding the defects of a chip or other similar applications.

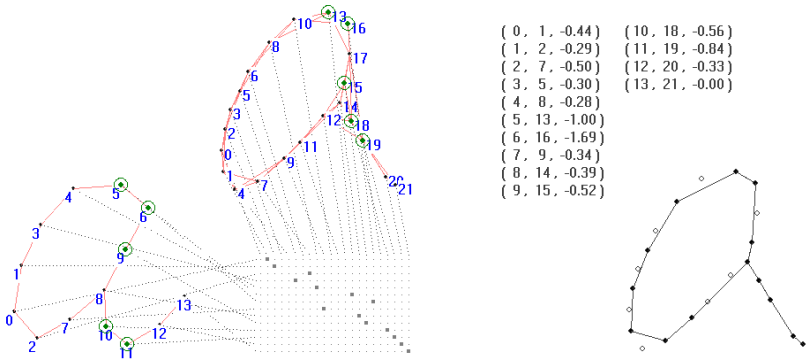


Figure 4.3: An example of the difference between two patterns. The circled place indicates the bigger difference between the features.

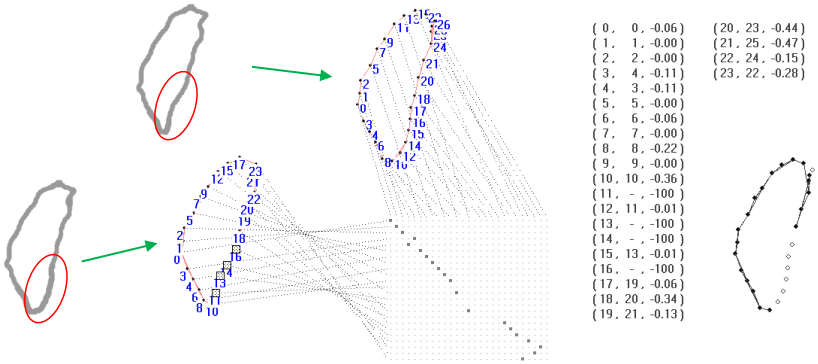


Figure 4.4: The difference of two similar contours is found by our method.

Chapter 5

Summary

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

Bibliography

[1] C. Y. Liou and H. C. Yang. Handprinted Character Recognition Based On Spatial Topology Distance Measurement. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(9): 941–945, 1996.

[2] C. Y. Liou and W. J. Yu. Ambiguous Binary Representation in Multilayer Neural Networks. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 1, pages 379–384. IEEE, 1995.

[3] D. R. Liou, C. C. Lin, and C. Y. Liou. Setting Shape Rules for Handprinted Character Recognition. In *ACIIDS (2)*, pages 245–252, 2012.

[4] J. Rocha and T. Pavlidis. A Shape Analysis Model with Applications to a Character Recognition System. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(4):393–404, 1994.

[5] J. Rocha and T. Pavlidis. Character Recognition Without Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(9):903–909, 1995.

[6] S. W. Lu, Y. Ren, and C. Y. Suen. Hierarchical Attributed Graph Representation And Recognition of Handwritten Chinese Characters. *Pattern Recognition*, 24(7): 617–632, 1991.

[7] C. Y. Liou and H. C. Yang. Selective Feature-to-Feature Adhesion for Recognition of Cursive Handprinted Characters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(2): 184–191, 1999.

[8] J.J. Hopfield and D.W. Tank. Neural Computation of Decisions in Optimization Problems. *Biological cybernetics*, 52(3):141–152, 1985.

[9] C. Y. Liou and H. C. Yang. Self-Organization of High-Order Receptive Fields in Recognition of Handprinted Characters. In *Neural Information Processing, 1999. Proceedings. ICONIP'99. 6th International Conference on*, volume 3, pages 1161–1166. IEEE, 1999.

[10] A. Moscona and H. Moscona. The Dissociation and Aggregation of Cells from Organ Rudiments of The Early Chick Embryo. *Journal of anatomy*, 86(3):287, 1952.

[11] P. L. Townes and J. Holtfreter. Directed Movements And Selective Adhesion of Embryonic Amphibian Cells. *Journal of experimental zoology*, 128(1):53–120, 1955.

[12] N. M. Nasrabadi, W. Li, and C. Y. Choo. Object Recognition by a Hopfield Neural Network. In *ICCV*, pages 325–328, 1990.

[13] Ponnuthurai N. Suganthan, Eam Khwang Teoh, and Dinesh P. Mital. Pattern Recognition by Homomorphic Graph Matching Using Hopfield Neural Networks. *Image Vision Comput.*, 13(1):45–60, 1995.

[14] H. Szu. Fast TSP Algorithm Based on Binary Neuron Output and Analog Neuron Input Using The Zero-Diagonal Interconnect Matrix and Necessary and Sufficient Constraints of the Permutation Matrix. *IEEE Trans. Int'l Conf Neural Networks*, 2: 259–266, 1988.

[15] S.V.B. Aiyer, M. Niranjan, and F. Fallside. A Theoretical Investigation Into the Performance of the Hopfield Model. *Neural Networks, IEEE Transactions on*, 1(2): 204–215, 1990.

[16] D. E. Rummelhart, G.E. Hinton, and J. L. McClelland. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1986.

本論文取材自國科會計畫 NSC 100-2221-E-002-234-MY3，題目解答為指導教授所授，全文為類神經網路實驗室學長與同學協力完成。