Skeleton-Based Recognition of Chinese Calligraphic Character Image

Kai Yu, Jiangqin Wu, and Yueting Zhuang

College of Computer Science, Zhejiang University
Hangzhou, 310027, P. R. China
{stephan,wujq,yzhuang}@zju.edu.cn

Abstract. The large amount of digitized Chinese calligraphic works in existence is a valuable part of the Chinese cultural heritage. But they can hardly be recognized by optical character recognition (OCR) which performs well on machine printed characters against clean background, because there are so different styles of shape complexity characters. So the approaches of automatic Chinese calligraphic character recognition become more and more important. A novel skeletonization algorithm called MFITS (morphology-fused index table skeletonization) is proposed and a skeleton-based Chinese calligraphic character recognition method is proposed too. The experiments show that MFITS can extract skeletons with only a few deformations and the skeleton-based Chinese calligraphic character image recognition method has a good performance.

Keywords: Chinese calligraphic character, MFITS, skeleton, recognition.

1 Introduction

Numerous collections of historical Chinese calligraphic works are valuable resources serving to historians as well as calligraphy lovers. Knowing what the calligraphic character images mean is very important. But due to the shape complexity of different styles of characters, it is difficult for people to recognize most of them. So finding an approach to recognize Chinese calligraphic characters written in different styles becomes a challenge now.

Calligraphic character recognition can be seen as a kind of offline handwriting recognition. Numerous promising research works have been done on recognizing manuscripts. Rath *et al.*[1] uses word-matching technology to recognize George Washington's manuscripts, and Yosef *et al.*[2] identifies ancient Hebrew manuscripts. Fuzzy technique[3] and SVM[4] were used to recognize cursive writing characters.

For offline Chinese character recognition, the calligraphic works are scanned and digital images are generated. These images are recognized by computers. There're many researches[5][6] on offline recognition for small set of Chinese characters but they can only be used in their specific areas. Neural network[7] and hidden Markov model[8] are imported to recognize handwriting Chinese, too.

Y.-M.R. Huang et al. (Eds.): PCM 2008, LNCS 5353, pp. 228–237, 2008.

English contains only 52 upper and lower case letters. However, thousands of Chinese characters are common used. So when recognizing Chinese characters, much more features must be analyzed and compared.

Till now, there're no published researches on calligraphic character recognition. And the available techniques such as OCR cannot be directly used to recognize calligraphic character, due to:

1) *Different width*: Ancients wrote with brush, which was soft and caused the widths of strokes vary in a large scope. Even in one stroke, the widths varied from here to there.

2) *Deformation*: The strokes in calligraphic characters are not regular. Some time lots of strokes are distorted and omitted.

3) *Complexity*: There are many different styles of calligraphic characters and even for one character there're different shapes. Strokes are connected or broken, too. As a result, the calligraphic characters are very different from their printed form.

4) *Degradation*: There are many noises on historical calligraphic works and the characters are not clear enough to recognize.

5) *Variable morphology*: The morphology of ancient Chinese is very different from modern Chinese, so the context cannot be used to help recognizing calligraphic characters.

We will propose a novel calligraphic character recognition approach based on the skeleton of the character. And for the skeleton extraction, we will also propose a novel calligraphic character skeletonization approach, which is called morphology-fused index table skeletonization (MFITS). Finally, we give some experiments.

2 Skeleton-Based Calligraphic Character Recognition

2.1 System Architecture

The Chinese calligraphic character recognition system is composed of two processes: candidate calligraphic character database building and recognition, which works as follow:

0) Input: The whole pages of digitized calligraphic works are inputted.

1) Binarization: The input pages are binarized and the noise is removed.

2) Page segmentation: The pages are segmented to individual calligraphic characters.

3) Normalization: The individual character is normalized to a certain size.

4) Skeletonization: The individual character is skeletonized by MFITS.

5) Global features generation: The features for filtering are generated.

6) Filtering: Query the database using the features generated above.

7) Skeleton comparison: The query results are compared to the input character image one by one and the similarities are calculated.

8) Output: The similarities are filtered by a gate value, and sorted. The probably recognition results are generated according to the annotated character for each calligraphic image in the database, and returned to the user.

When storing new data into the database, Step 0 to Step 5 are needed and the global features and skeleton generated are serialized and inserted into the database, as well as the meaning of the characters recognized manually.

The key techniques in the system are MFITS and character recognition based on skeletons, which are the proposed approaches in our paper.

2.2 MFITS: Morphology Fused Index Table Skeletonization

Chinese calligraphic character skeletonization is a thinning procedure. There are many methods, of which index table and mathematic morphology are two typical ways. Index table is the quickest, but it may cause some deformations at the cross positions; while mathematic morphology may cause small branches both at the end of strokes as well as the middle of the strokes where it is not very smooth. However, they won't cause deformations at the cross point in some special conditions according to our experiments. When the character is in *Hei* style whose strokes look like rectangles, skeletonizing with mathematic morphology may cause few deformations.

Comparing the advantages and disadvantages of the two methods, and observing the situations when the deformations occur, a new approach called MFITS (Morphology-Fused Index Table Skeletonization) is proposed. Firstly, the binarized calligraphic character image is thinned with a proper index table, which only removes the raised pixels. Then thinned result, *Hei* style like character, is skeletonized by the mathematic morphology skeletonization approach. Finally, the small branches are removed and the skeleton is obtained.

For thinning with index table, whether a pixel is to be removed is decided by the pixels around it. Suppose $a_1, a_2, \ldots, a_8$ are the 8 pixels around a_0. The center pixel a_{0final} can be calculated by the following equation.

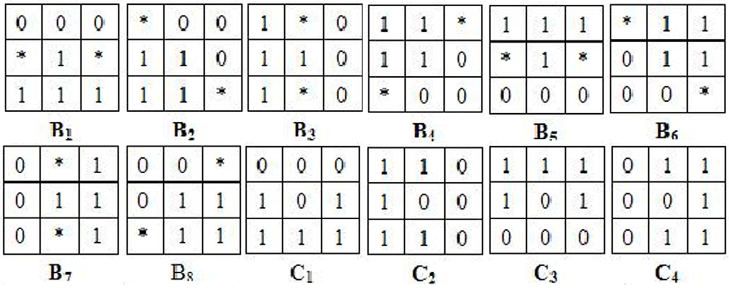
$$a_{0final} = \begin{cases} 0 & (a_0 = 0) \\ f(a_1, a_2, \ldots, a_8) & (a_0 \neq 0) \end{cases} \tag{1}$$

where f is just the index table, which is the mapping from each group of a_i $(i = 1, 2, \ldots, 8)$ to a_{0final}. This table contains $2^8 = 256$ rows. The matrix of the binarized image needs to be scanned repeatedly until no pixel removed.

To skeletonize with mathematic morphology[9][10][11], a set of mask matrixes are used to dilate and erode the binarized images of individual calligraphic characters with the following equation:

$$\mathbf{A} \otimes \{\mathbf{D}\} = (((\ldots(\mathbf{A} \otimes \mathbf{B}_1) \oplus \mathbf{C}_1) \otimes \mathbf{B}_2) \ldots) \otimes \mathbf{B}_7) \oplus \mathbf{C}_4) \otimes \mathbf{B}_8 \tag{2}$$

Here $\{\mathbf{D}\} = \{\mathbf{B}_1, \mathbf{C}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{C}_2, \mathbf{B}_4, \mathbf{B}_5, \mathbf{C}_3, \mathbf{B}_6, \mathbf{B}_7, \mathbf{C}_4, \mathbf{B}_8\}$ is the set of mask matrixes shown in Fig. 1, $\oplus$ is dilation operation and $\otimes$ is erosion operation. Mask matrixes $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4$ are imported to delete the pixels on both sides of strokes symmetrically so that there're fewer deformations in the skeleton.



$$\underset{B_1}{\begin{bmatrix}0&0&0\\ *&1&*\\ 1&1&1\end{bmatrix}}\quad \underset{B_2}{\begin{bmatrix}*&0&0\\ 1&1&0\\ 1&1&*\end{bmatrix}}\quad \underset{B_3}{\begin{bmatrix}1&*&0\\ 1&1&0\\ 1&*&0\end{bmatrix}}\quad \underset{B_4}{\begin{bmatrix}1&1&*\\ 1&1&0\\ *&0&0\end{bmatrix}}\quad \underset{B_5}{\begin{bmatrix}1&1&1\\ *&1&*\\ 0&0&0\end{bmatrix}}\quad \underset{B_6}{\begin{bmatrix}*&1&1\\ 0&1&1\\ 0&0&*\end{bmatrix}}$$

$$\underset{B_7}{\begin{bmatrix}0&*&1\\ 0&1&1\\ 0&*&1\end{bmatrix}}\quad \underset{B_8}{\begin{bmatrix}0&0&*\\ 0&1&1\\ *&1&1\end{bmatrix}}\quad \underset{C_1}{\begin{bmatrix}0&0&0\\ 1&0&1\\ 1&1&1\end{bmatrix}}\quad \underset{C_2}{\begin{bmatrix}1&1&0\\ 1&0&0\\ 1&1&0\end{bmatrix}}\quad \underset{C_3}{\begin{bmatrix}1&1&1\\ 1&0&1\\ 0&0&0\end{bmatrix}}\quad \underset{C_4}{\begin{bmatrix}0&1&1\\ 0&0&1\\ 0&1&1\end{bmatrix}}$$

Fig. 1. Mask matrixes for mathematic morphology

2.3 Global Features Generation and Filtering

When searching large scale Chinese calligraphic character database, the characters should be filtered firstly in some quick ways in order to speed up the calligraphic character recognition. In our paper, the candidate characters are filtered according to some global features.

Total number of black pixels in the skeletons is the first global feature. The candidate characters with a large difference can be rejected.

Stroke density is another important feature to see whether it is a possible candidate or not. Here the stroke density is defined as the number of times a scanning beam drills through a stroke.

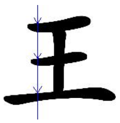


Fig. 2. The scanning beam

A group of scanning beams are used to scan the normalized character both in horizontal and vertical. The scanning value will increase when a scanning beam find the color changes from white to black. For example, the scanning value of the beam shown in Fig. 2 is 3. Thus, the horizontal and vertical stroke densities can be defined as the follows.

$$density_{horizontal} = \sum_{i=1}^{size} \frac{h_i}{size} \quad density_{vertical} = \sum_{i=1}^{size} \frac{v_i}{size} \tag{3}$$

where h_i and v_i are the ith horizontal and vertical scanning values, and $size$ is 64 because the calligraphic characters have been normalized to 64 × 64.

The candidate character cannot be similar to the input character if its stroke density is very different from the input character, so it should be rejected. The proportion between the horizontal and the vertical stroke density is independent of the character size and the writing style. So some impossible candidate characters can be rejected by comparing these two values.

Some more impossible candidate characters can also be rejected because of the large difference between the horizontal and vertical histograms.

2.4 Database Building

Till now, all the data and features which need to be stored in the database for recognition are obtained. The calligraphic character database is created.

All the scanned calligraphic works are segmented, binarized and skeletonized. Then the global features are generated and the calligraphic characters segmented are annotated manually one by one. All the data above are stored in the database.

2.5 Similarity Calculating and Skeleton Comparison

After filtering, the rest candidate characters are all similar to the input calligraphic character globally. Now the local features will be generated and similarities between the input character image and the left candidate character images are calculated one by one to find the most similar candidate character images.

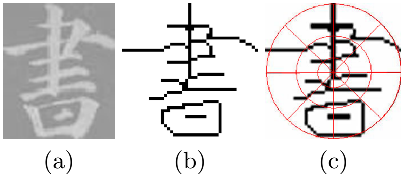


Fig. 3. Features generated for a calligraphy. (a) An individual calligraphic character image; (b) The skeleton (c) The corresponding log-polar bins for context computing.

The input character is skeletonized with MFITS described above. For each pixel in the skeleton, the around areas are divided as follows: first, four concentric circles whose radii are $size/2$, $size/4$, $size/8$ and $size/16$; then four lines with included angel 45°. Thus the around pixels of a certain pixel are put into 32 bins, which is shown in Fig. 3(c). These bins are not in the same size because the pixels nearby is much more important than the faraway pixels. Then the pixels in each bin are counted and a vector $\mathbf{N} = (n_1, n_2, \ldots, n_{32})^{\mathrm{T}}$ is formed, which is the eigenvector of pixel loaded in the center of the circles.

Now the distance between pixel a in skeleton $\mathbf{A}$ and pixel b in skeleton $\mathbf{B}$ can be defined. The eigenmatrix can be formed as the follows.

$$\mathbf{C} = \begin{pmatrix} x_1 & y_1 & \mathbf{N}_1 \\ x_2 & y_2 & \mathbf{N}_2 \\ \vdots & \vdots & \vdots \\ x_p & y_p & \mathbf{N}_p \end{pmatrix} \tag{4}$$

Each row in the matrix contains the position of a pixel in the skeleton of the calligraphic character and the corresponding eigenvector. Suppose $a \in \mathbf{A}$

and $b \in \mathbf{B}$, whose eigenvectors are $\mathbf{N}_a = (n_{a1}, n_{a2}, \ldots, n_{a32})^{\mathrm{T}}$ and $\mathbf{N}_b = (n_{b1}, n_{b2}, \ldots, n_{b32})^{\mathrm{T}}$, the distance between pixels a and b is defined as the following equation.

$$d(a,b) = \sum_{i=1}^{32} \frac{(n_{ai} - n_{bi})^2}{n_{ai} + n_{bi}} \tag{5}$$

Suppose **A** and **B** are skeletons of two same calligraphic characters; pixel b in skeleton **B** is the corresponding pixel of pixel a in skeleton **A**, we call pixel b the similar pixel of pixel a.

The similar pixel in skeleton **B** of the pixel a_{ij} in skeleton **A** can be found as follows:

$$(i', j') = \arg\min(d(a_{ij}, b_{kl})) \tag{6}$$

where (k, l) must satisfy

$$\begin{cases} i - \theta \cdot size < k < i + \theta \cdot size \\ j - \theta \cdot size < l < j + \theta \cdot size \\ 0 < k \le m \\ 0 < l \le n \end{cases} \tag{7}$$

Here θ is an experienced value.

Because the similar pixel of a certain pixel in the skeleton of a similar calligraphic character should be in a neighborhood of the certain pixel corresponding to their whole skeletons, the constraint in Equation (7) is defined.

When giving two skeletons **A** and **B**, the difference between pixel a_{ij} in **A** and its similar pixel $b_{i'j'}$ in skeleton **B** can be calculated with Equation (8).

$$diff(a_{ij}, \mathbf{B}) = \begin{cases} d(a_{ij}, b_{i'j'}) + \alpha L(a_{ij}, b_{i'j'}) & (a_{ij} \text{ is in the skeleton}) \\ 0 & (a_{ij} \text{ is not in the skeleton}) \end{cases} \tag{8}$$

Here $L(a, b)$ is the Euclidean distance between pixel a and b when skeleton **A** is overlapped with skeleton **B** , and α is a constant. Thus the distance between two skeletons **A** and **B** can be obtained as Equation (9).

$$Dist(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^{m} \sum_{j=1}^{n} diff(a_{ij}, \mathbf{B}) \tag{9}$$

This is just the similarity between the two calligraphic characters.

Finally, the candidate calligraphic characters whose similarities are smaller than a gate value are sorted by the similarity.

2.6 Recognition

The similarities of the possible candidate character images have been calculated and sorted with the approach described above. Suppose the first N character images in the sorted list are $I_1, I_2, \ldots, I_N$, the weight $\delta_i (i = 1, 2, \ldots, N)$ of each candidate character image can be defined as the follows.

$$\delta_i = \frac{1}{i^2} + \frac{1}{Dist(\mathbf{C}_i, \mathbf{C}_0)} \tag{10}$$

where $\mathbf{C}_i$ is the skeleton of the candidate character image I_i and $\mathbf{C}_0$ is the skeleton of the input character image I_0.

Suppose there are M annotated characters $A_1, A_2, \ldots, A_M$ for the N candidate character images. And q_i images $I_{i1}, I_{i2}, \ldots, I_{iq_i}$ are the calligraphic character images whose annotated character is A_i. Thus, the probability that I_0's meaning is A_i is

$$prob_i = \frac{\sum_{k=1}^{q_i} \delta_{ik}}{\sum_{j=1}^{M} \sum_{k=1}^{q_j} \delta_{jk}} \tag{11}$$

Finally the M annotated characters are sorted again by the probabilities and the recognized results are returned to the user.

3 Experiments and Evaluation

We implemented the whole system described above with C#. In this section, we compare the skeletonization results of MFITS with skeletons extracted with index table and mathematic morphology. Finally we show the results of both single calligraphic character recognition and full page recognition in order to prove that our approach is effective and efficient.

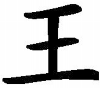
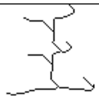
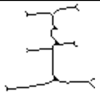
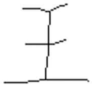
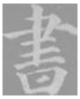
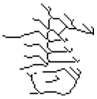
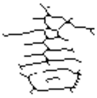
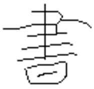
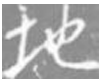
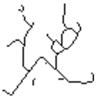
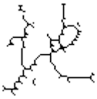
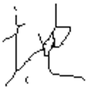
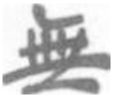
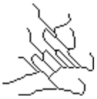
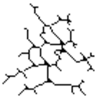
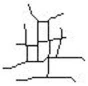
3.1 Database Building

Firstly, we scanned hundreds of calligraphic books, segmented all the pages and collected about 12000 calligraphic character images. We also collected the 3500 printed *Kai* style characters that were commonly used because they were similar as *Kai* style calligraphic characters. So the database contain about 15500 character samples. Then the skeletons were extracted with MFITS and the features for filtering were calculated. Finally, all the calligraphic characters were annotated manually and the features as well as the skeletons were serialized and stored in a database. Thus, the candidate character database was built.

3.2 Skeleton Extraction

Here we skeletonized all the calligraphic characters in our database with MFITS and compared them with the skeleton extracted with index table and morphology. Some randomly selected results are shown in Table 1. It can be seen that there's only a few deformations in the skeletons extracted by MFITS, so MFITS is much more effective than index table or mathematic morphology.

Table 1. Skeletonization with three different approaches

Original calligraphic characters	Skeletonized with index table	Skeletonized with mathematic morphology	Skeletonized with MFITS
			
			
			
			

3.3 Individual Chinese Calligraphic Character Recognition

300 calligraphic character samples in different styles were chosen randomly from Internet, which were not included in the candidate database. Our proposed system was used to recognize them. Table 2 shows some of the results. 289 samples were recognized correctly. The recognition rate is about 96.3%.

Table 2. Recognition results of some calligraphic character samples

Input Sample						
Result	其	月	之	者	铭	德

From the result, you can see that the proposed approach is effective in recognizing individual Chinese calligraphic characters.

The first row of Fig. 4 shows three random selected calligraphic character samples with incorrect results. Each of them has a correct result returned as a second choice. The second row of Fig.4 shows the most similar calligraphic character samples in our database whose annotations were the first recognition results. You can see that each group of samples is all very similar in both shape and construction. And in our experiments, the difference between the probabilities of the first and second results was less than 3 percents.

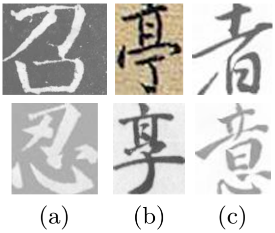


Fig. 4. Three samples recognized incorrectly with the most similar calligraphic characters in database

3.4 Full Page Calligraphic Character Recognition

The input of the system here was a famous works ” *The Orchid Pavilion*” written by Wang Xizhi. The first four columns are shown in Fig.5(a). You can see that there were noises such as red seals and some grey areas.

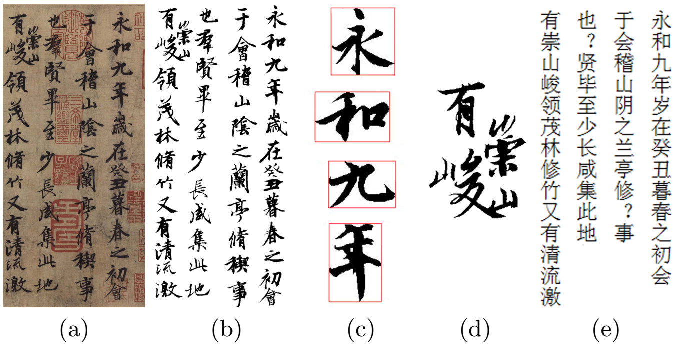


Fig. 5. A part of example of full page calligraphic character recognition. (a) The original calligraphic image; (b) The binarized calligraphic image; (c) The segmented result; (d) An area which cannot be segmented automatically; (e) The recognition result.

The seals as well as the grey areas and some noise pixels were removed. The binarized image of the first four columns is shown in Fig.5(b). Fig.5(c) shows the segmented results. There is an area that cannot be segmented automatically, as shown in Fig.5(d). This area contains four characters in which two on the right are inserted. So there are no blank areas in the histogram between them. However, this area is too large to be one character. So it needs to be segmented manually. There are some characters marked as ”?” in the result of the first four columns shown in Fig.5(e). It means the calligraphic character here cannot be recognized. The rest calligraphic characters in the first four columns are all recognized correctly. For the whole works, there are 323 characters and 308 are recognized correctly. So the recognition rate is about 95.4%.

This experiment shows that our recognition approach is effective and efficient.

4 Conclusion

This paper proposed a novel approach to recognize Chinese calligraphic characters based on their skeletons. This paper also gave a morphology fused method to skeletonize calligraphic characters with index table, called MFITS. Based on the approaches above, this paper implemented a complete Chinese calligraphic character recognition system, which can recognize individual calligraphic characters and full pages of calligraphic works. Our experiments show that MFITS can extract calligraphic character's skeletons with only a few deformations, and the recognition approach can get a high recognition rate when there's a database with enough annotated Chinese calligraphic character samples.

Acknowledgments. Thanks for the support of National Natural Science Foundation of China (No.60773176, No.60533090, No.60525108) and Program for Changjiang Scholars and Innovative Research Team in University(IRT0652).

References

1. Rath, T.M., Kane, S., Lehman, A., Partridge, E., Manmatha, R.: Indexing for a Digital Library of George Washington's Manuscripts: A Study of Word Matching Techniques. CIIR Technical Report MM-36 (2002)
2. Yosef, I.B., Kedem, K., Dinstein, I., Beit-Arie, M., Engel, E.: Classification of Hebrew Calligraphic Handwriting Styles: Preliminary Results. In: Proceedings of First International Workshop on Document Image Analysis for Libraries, pp. 299–305 (2004)
3. Suresh, R.M., Arumugam, S.: Fuzzy technique based recognition of handwritten characters. Image and Vision Computing 25(2), 230–239 (2007)
4. Camastra, F.: A SVM-based cursive character recognizer. Pattern Recognition 40(12), 3721–3727 (2007)
5. Yan, J.: Study on Handwritten Chinese Character Recognition Based on Multi-Structure Information Fusion. PhD thesis, Chongqing University (2002)
6. Shi, D., Damper, R.I., Gunn, S.R.: Offline handwritten Chinese character recognition by radical decomposition. ACM Transactions on Asian Language Information Processing 2(1), 27–48 (2003)
7. Jing, C., Zhichun, M., Xing, F., Dapeng, D.: Simulation research of Chinese character recognition based on self-organizing neural network. Computer Engineering 33(11), 170–172 (2007)
8. He, Z., You, X., Tang, Y.: Writer identification of Chinese handwriting documents using hidden Markov tree model. Pattern Recognition 41(4), 1295–1307 (2008)
9. Gonzalez, W.: Digital Image Process, 2nd edn., pp. 420–453. Prentice Hall, Lebanon (2002)
10. Yuan, W., Yang, Y., Bin, X., Hong, W.: Off-line recognition method for handwritten Chinese character based on the mathematic morphology. Journal of Computer Application 26(3), 622–623, 626 (2006)
11. Jianping, W., Zituo, Q., Jinling, W., Guojun, L.: Chinese character stroke thinning and extraction based on mathematic morphology. Journal of Heifei University of Technology (Natural Science) 28(11), 1431–1435 (2005)