

A New Method of Recognizing Chinese Fonts *

Zihua Yang

School of Mathematical Sciences

South China Normal University, Guangzhou 510631, P. R. China

Lihua Yang

School of Mathematics and Computing Science

Sun Yat-sen University, Guangzhou 510275, P. R. China

Email: mcsylh@zsu.edu.cn

Ching Y. Suen

Center for Pattern Recognition and Machine Intelligence

Concordia University, Montreal, Canada H3G 1M8

Abstract

Chinese fonts are recognized by a new method based on Empirical Mode Decomposition. Five basic strokes have been selected to characterize the features of Chinese fonts. Based on them, stroke feature sequences of a given text block are calculated. Once decomposed by EMD, the first two Intrinsic Mode Functions corresponding to each stroke feature sequence are used to calculate the stroke energy of all the five basic strokes. These energies are combined with the five averages of the residues to produce a ten-dimensional feature vector. Finally, the minimum distance classifier is used to recognize the fonts. Experiments show encouraging recognition rates.

Keywords: *Empirical mode decomposition (EMD), Hilbert-Huang Transform (HHT), Font Recognition*

1. Introduction

Font recognition is an important and challenging topic in automatic document analysis and processing. Its goal is to recognize the font of a given text image. It is obviously very useful to recognize these attributes in many applications such as producing the re-editable text and achieving automatic typesetting in the output of an automatic document processing system. On the other hand, if the fonts of a

text can be recognized correctly, it can be used to enhance the recognition rate of an OCR system.

However, partially due to the difficulty of discriminating similar fonts, the issue has been ignored by researchers in spite of its clear importance. Existing studies consist of three main classes: (a) the methods based on local attributes such as serif, bold, etc. [1, 2]; (b) the methods based on local and/or global typographical features [3, 4, 5]; and (c) the methods based on texture analysis [6, 7]. Different methods employ different kinds of features to recognize the fonts. For a Chinese text, because of its structural complexity, font recognition is more difficult than those of western languages such as English, French, Russian etc. Only a few researcher have addressed this issue. Zhu and Tan [6] used Gabor filter to extract font features based on global texture analysis and obtained a high recognition rate. The disadvantage of this method is its high computational complexity due to the use of high dimensional features. Chen and Ding [8] presented an approach for font recognition of single Chinese characters based on wavelet feature, which produced a recognition rate of 97.35%. However the feature dimensions of 256 lead to a high computational complexity too.

This paper presents a novel approach to recognize Chinese fonts based on Empirical Mode Decomposition (EMD). By analyzing and comparing a large number of Chinese characters, five basic strokes have been selected to characterize the stroke attributes of Chinese fonts. Based on them, *stroke feature sequences* of a given text block are calculated. After EMD, the first two Intrinsic Mode Functions (IMFs) corresponding to each *stroke feature sequence*, which are of the highest frequencies, are used to produce the so called *stroke en-*

* This work was supported by NSFC(No. 60475042), the foundation of scientific and technological planning project of Guangzhou city (No. 2003J1-C0201) and the Scientific Foundation for Young Teachers of Sun Yat-sen University.

ergy, which is the average energy of the two IMFs over the length of the sequence. Calculating the *stroke energy* for all the five basic strokes and combining the five averages of the residues, a ten-dimensional feature vector is formed. Finally, based on the features, a minimum distance classifier is used to recognize the font as done by Tan and his co-workers[6]. Experiments show encouraging recognition rates.

This paper is organized as follows: the technique of font feature extraction is introduced in Section 2; the algorithm for Chinese font recognition based on EMD is given in Section 3; Section 4 contains the experimental results and analysis and finally, Section 5 gives the conclusion of the paper.

2. Font Feature Extraction

2.1. Preprocessing

For a given text image, the sizes of characters, the spaces between characters and text lines are usually not uniform. Sometimes, blank spaces also exist at the end of a paragraph. These phenomena will affect the recognition results seriously and must be suppressed by suitable normalizations before the font recognition.

In this paper, since the preprocessing is not our main concern, the approach presented in [6] is directly employed to preprocess the original text image. The readers are referred to [6] for details.

2.2. Stroke Feature Series

It is well known that Chinese characters consist of basic strokes. Some of them appear more frequently than the others. Each stroke has its own shape when printed in different fonts, as shown in Fig. 1. For example, a horizontal stroke (‘—’, Chinese name “Heng”) in a Chinese character is a completely horizontal segment if its font is HeiTi while it is a horizontal segment with a slight slant if its font is KaiTi, as shown in Fig. 1. Similarly, a vertical stroke (‘|’, Chinese name “Shu”) in a Chinese character is a completely vertical segment if its font is Regular SongTi while it is a vertical segment with some slant if its font is Italic SongTi, as shown in Fig. 1 too. The upper-right-to-lower-left stroke (‘/’, Chinese name “Pie”) and upper-left-to-lower-right stroke (‘\’, Chinese name “Na”) of font Lishu are more explanate than those of font SongTi. It has a roundish corner at the turning point of the horizontal-to-vertical stroke (‘┐’, Chinese name “HengZhe”) in characters of YouYuan font, instead of a right-angled corner as it is in those of HeiTi font. The above attributes are essential differences among different fonts and are used to recognize the font of a given text image in this paper. The five basic strokes selected to extract the

features for font recognition are “Heng”, “Shu”, “Pie”, “Na” and “HengZhe”, whose templates are illustrated graphically in Fig. 2. Each template of the first four basic strokes contains 8 pixels and its line width is set equal to one pixel. The template of the fifth stroke (“HengZhe”) contains seven pixels. To normalize the number of pixels in all selected basic strokes, we increase the weight of the pixel at the turning point of the stroke “HengZhe” by 2. In this sense, it can be said that each template of all the five basic strokes consists of eight pixels.

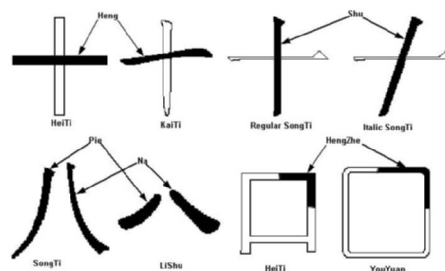


Figure 1. Some basic strokes of six kinds of frequently used Chinese fonts: HeiTi, KaiTi, Regular/Italic SongTi, FangSong, LiShu and YouYuan respectively.

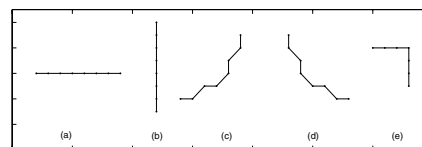


Figure 2. Templates of the five selected basic strokes

Each basic stroke plotted in Fig.2 is a 8×1 template of some geometric structure. For a given text image, we match these templates with the text at a random location and calculate the average of the gray levels of the eight pixels matched. It is easily understood that, the average is a measure of similarity between the stroke structure of the text at the location and the basic stroke. After doing this many times for random locations of the text block, many such averages are calculated for each basic stroke. If N av-

erages have been obtained for a given text block and a basic stroke, a series of N data is formed according to the order of calculation. The series is called *stroke(x) feature series*, in which x is a, b, c, d or e corresponding to the basic stroke used. When the calculations have been conducted for all the five basic strokes, five stroke feature series are obtained finally, which characterize the structural attributes of characters of different fonts. Generally, the larger N is, the better the statistical property of *stroke(x) feature series* is, consequently the more reliable it is to use the *stroke(x) feature series* to characterize the basic stroke information embedded in the text block. On the contrary, high computational complexity may be caused by a large N . In accordance with our experience, it is enough to set $N \approx 10 \max(W, H)$ for a text image of width W and height H .

2.3. Font Feature Extraction

The five stroke feature series have obviously different distributions if the text is of different fonts. For example, the amplitudes of the *stroke(a) feature series* of a text of SongTi or HeiTt are much larger than those of FangSong or KaiTi. It implies that the high-frequency energy of the *stroke(a) feature series* of SongTi or HeiTt text extracted from the first several IMFs of its EMD decomposition should be larger than those of FangSong and KaiTi text. To explain this phenomenon more clearly, let us observe the following experiment shown in Fig.3.

In Fig. 3, (I) is a Chinese text block of HeiTt, (II) is a Chinese text block of KaiTi, (III) is the *stroke(a) feature series* of 200 samples corresponding to (I) and its EMD decomposition, (IV) is the *stroke(a) feature series* of 200 samples corresponding to (II) and its EMD decomposition. The dotted lines are their corresponding instantaneous energies. To observe more clearly, in this experiment, the length of the *stroke feature series*, N , is set equal to 200 instead of $N \approx 10 \max(W, H)$ as described in Subsection 2.2. For simplicity, we denote the *stroke(x) feature series* of a text block of HeiTt and KaiTi by S_x^H and S_x^K , their i -th IMF by $imf_x^H(i)$ and $imf_x^K(i)$ respectively; similarly the residues are denoted by R_x^H and R_x^K respectively. In this example, the text blocks are gray-scale images with 256 gray levels: 0 for black pixel and 255 for white pixel. From the stroke feature series S_a^H , one can observe that S_a^H is 0 or close to 0 at some locations, which implies that the basic stroke(a) matches the text perfectly or almost perfectly at those locations. Contrarily, S_a^K is never equal or close to 0, which means that there are hardly any locations where the text matches the basic stroke(a) well for the KaiTi text. After they are decomposed by EMD, it can be seen easily that $imf_a^H(1)$ and $imf_a^H(2)$ are of higher amplitudes than those of $imf_a^K(1)$ and $imf_a^K(2)$, which shows that S_a^H contains a higher high-frequency energy than S_a^K . Usually, the differ-

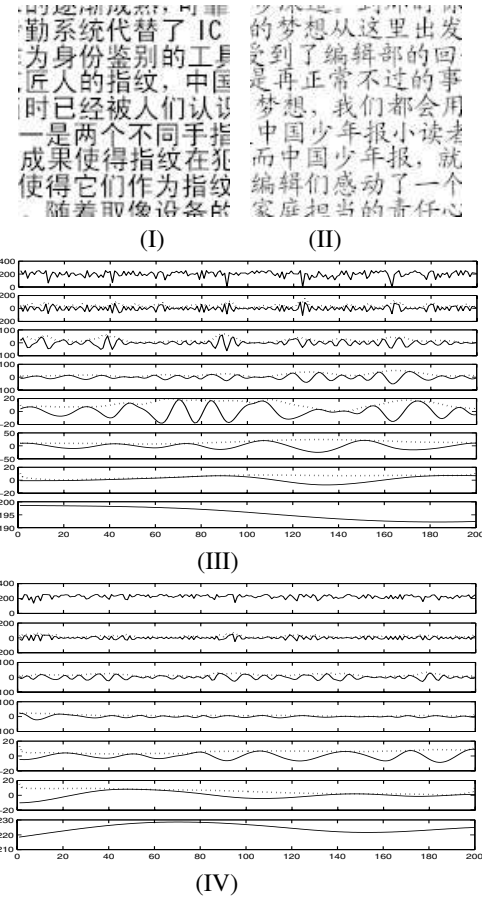


Figure 3. (I) is a Chinese text block of HeiTt; (II) is a Chinese text block of KaiTi; (III) is the *stroke (a) feature series* of 200 data corresponding to (I) and its EMD decomposition; (IV) is the *stroke (a) feature series* of 200 data corresponding to (II) and its EMD decomposition.

ences among the energies of the lower-frequency IMFs corresponding to texts of different fonts are so small that they give a negligible contribution to the font recognition. Therefore, in this paper, only the first two IMFs of each *stroke feature serial* are employed to design our algorithm for font recognition. From the residues of the EMD decomposition shown in Fig. 3, it is easy to see that R_x^H is around 195 while R_x^K is about 225. It can be interpreted without difficulty that a HeiTt text block contains many more "Heng" strokes, i.e., basic strokes (a), than a KaiTi text. This fact suggests that the residue should be utilized as a feature to classify different fonts too. The discussion above leads to the following algorithm for the extraction of font feature vector for a

text block.

Algorithm 2.1 For a text block of width W , height H and gray levels 256 (black pixels correspond to 0 and white pixels correspond to 255), the feature vector for font and font style recognition is produced according to the following steps:

Step 1 Conduct preprocessing as described in Subsection 2.1;

Step 2 Choose a suitable integer N which is approximately $10 \max(W, H)$ as discussed in Subsection 2.2. Then calculate the stroke(x) energies, e_x , and the stroke(x) residue energies, r_x , for $x = a, b, c, d$, and e as follows:

- (1) Generate stroke(x) feature series, S_x , by the method introduced in Subsection 2.2;
- (2) Based on the decomposition of S_x [9], calculate stroke(x) energy, e_x , and stroke(x) residue energy, r_x , as follows:

$$e_x = \frac{1}{2N} \sum_{j=1}^N [\text{imf1}_x(j) + \text{imf2}_x(j)],$$

$$r_x = \frac{1}{N} \sum_{j=1}^N \text{res}_x(j),$$

where, $\text{imf1}_x, \text{imf2}_x$ and res_x are the first two IMFs and the residue of the EMD decomposition of a stroke(x) feature series respectively.

Step 3 Let

$$V = [e_a, e_b, e_c, e_d, e_e, r_a, r_b, r_c, r_d, r_e],$$

which is the feature vector we want for font and font style recognition.

3. Font Recognition

Once the feature vector has been obtained, the next step is to design a classifier for font recognition. In this paper, since the design of the classifier is not our concern, to make our work more comparable with that of [6], a simple technique based on the weighted Euclidean distance (WED) is used as the classifier even though a better classifier can no doubt provide better recognition rates.

For convenience, we rewrite the feature vector

$$V = [e_a, e_b, e_c, e_d, e_e, r_a, r_b, r_c, r_d, r_e]$$

obtained by Algorithm 2.1 as

$$V = [v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}].$$

Suppose there are K classes of different fonts and font styles for classification. For each k from 0 to K , calculate the mean vector, denoted by

$$V^{(k)} = [v_1^{(k)}, v_2^{(k)}, v_3^{(k)}, v_4^{(k)}, v_5^{(k)}, v_6^{(k)}, v_7^{(k)}, v_8^{(k)}, v_9^{(k)}, v_{10}^{(k)}],$$

and the variance vector, denoted by

$$\Delta^{(k)} = [\delta_1^{(k)}, \delta_2^{(k)}, \delta_3^{(k)}, \delta_4^{(k)}, \delta_5^{(k)}, \delta_6^{(k)}, \delta_7^{(k)}, \delta_8^{(k)}, \delta_9^{(k)}, \delta_{10}^{(k)}],$$

based on the corresponding training samples. For a text block with an unknown font or font style, let its feature vector computed by Algorithm 2.1 be

$$V = [v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}].$$

Then the number of classes of fonts or font styles to be recognized is determined by the following equation:

$$k = \underset{1 \leq k \leq K}{\text{argmin}} \left\{ \sum_{x=1}^{10} \frac{(v_x - v_x^{(k)})^2}{(\delta_x^{(k)})^2} \right\}. \quad (1)$$

4. Experiments and Discussion

4.1. Samples for Training and Testing

To test our algorithm, experiments are conducted on six kinds of frequently used Chinese typefaces (SongTi, KaiTi, HeiTi, FangSong, LiShu and YouYuan), each of which includes four styles (Regular, Italic, Bold and Bold Italic), namely a total of 24 classes of fonts. All the samples are grayscale text images of 256 gray levels with a size of 128×128 pixels generated by a computer software and by scanner. The computer-generated text blocks are created by Photoshop 7.0 with a resolution of 72 pixels/inch. The scanner-generated text blocks are obtained by a scanner *HP scanjet 3670* with a resolution of 100dpi. For each font, 50 computer-generated text blocks (20 of them for training and 30 for testing) and another 50 scanner-generated text blocks (20 of them for training and 30 for testing too) are created. It means a total of 2400 samples for 24 fonts are employed in our experiment, 40 samples for training and 60 for testing each font or font style.

4.2. Experimental Results

For the samples given above, the experimental results are shown in Tab. 1, from which an average recognition rate of 97.2% is obtained. LiShu produced the highest average recognition rate among these six kinds of fonts, which is up to 99.1%, and FangSong has the lowest recognition rate, 94.5%. As for the four styles, Regular style has the highest and Bold style the lowest recognition rates, which are 98.6% and 95.7% respectively. The main factor causing Bold style the lowest recognition rate is that the Regular

	ST	KT	HT	FS	LS	YY	Ave.
Reg.	98.2	100	93.2	100	100	100	98.6
Bold	93.2	91.7	100	92.9	100	96.5	95.7
Ita.	100	100	96.5	93.3	100	100	98.3
BI	100	93.3	100	91.7	96.5	94.9	96.1
Ave.	97.9	96.3	97.4	94.5	99.1	97.9	97.2

Table 1. Recognition rate (%) of fonts and styles.

	ST	KT	HT	FS	LS	YY
ST	97.9	0	0.8	1.3	0	0
KT	0.2	96.3	0	2.8	0.7	0
HT	2.4	0	97.4	0	0	0.2
FS	2.1	3.4	0	94.5	0	0
LS	0	0	0.9	0	99.1	0
YY	0.4	0	1.7	0	0	97.9

Table 2. Font confusion rate (percent).

HeiTi are often confusable with the Bold SongTi or the Bold YouYuan, which makes the Regular HeiTi to have the lowest recognition rate of 93.2% among all the Regular fonts.

Some fonts and styles have similar shapes and are easily confused with one another. In the test samples used above, there are $4 \times 60 = 240$ samples for each font (including all the 4 styles). Let A be such a font, B be another font. If N samples within the set of 240 samples of A are recognized as B , we say the font confusion rate of A by B is $N/240 \times 100\%$. Tab. 2 is the Font Confusion Matrix (Percent), from which it can be seen that the most confusable pairs of fonts are FangSong and KaiTi. The font confusion rate of FangSong by KaiTi is 3.4% and that of KaiTi by FangSong is 2.8%.

4.3. Comparison with Other Existing Methods

In [6], Gabor filter is employed to extract font features based on global texture analysis and a high average recognition rate of 98.5% is achieved for the same 24 kinds of fonts as above, which is slightly higher than that of ours. Since there is no public database for Chinese font recognition now and we do not have the samples they experimented in [6], the comparison is not completely objective. In their experiments, only the computer-generated samples are used (see [6]), however, in our experiments samples including both the computer-generated and scanner-generated text blocks are tested. Moreover, the feature dimension in our algorithm is 10, which is much smaller than 16, the feature dimension used in [6].

5. Conclusion

This paper presents a novel approach to recognize Chinese fonts based on Empirical Mode Decomposition. The main advantages of the technique are listed below: (1) the feature dimension is very low; (2) the recognition rate is very high; (3) since the *stroke feature series* are extracted randomly from a text block, a text block can be reused many times, therefore less samples of text blocks are needed to train the classifier; (4) the sizes of each testing sample and training sample may be different.

It should be pointed out that since the *stroke feature series* depends largely on the selected basic strokes, the approach is not suitable for English font recognition now. However if some proper basic English strokes are selected as the substitutes for these strokes, we believe an algorithm for English font recognition can be developed without difficulty. In conclusion, this paper presents an interesting and worthy exploration on the applications of the theory of Hilbert-Huang Transform.

References

- [1] S.Khoubyari and J.J.Hull. Font and function word identification in document recognition. *Computer Vision and Image Understanding*, 63(1):66–74, 1996.
- [2] R. Cooperman. Producing good font attribute determination using error-prone information. *Int'l Society for Optical Eng. J.*, 3027:50–57, 1997.
- [3] H. Shi and T. Pavlidis. Font recognition and contextual processing for more accurate text recognition [a]. *ICDAR97 [C] . ULM, Germany : IEEE Computer Society Press*, pages 39–44, 1997.
- [4] A. Zramdini and R. Ingold. Optical font recognition using typographical features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):877–882, 1998.
- [5] MinChul Jung, YongChul Shin, and S N Srihari. Multifont classification using typographical attributes [a]. *ICDAR99 [C] . Bangalore , India : IEEE Computer Society Press*, pages 353–356, 1999.
- [6] Yong Zhu, Tieniu Tan, and Yunhong Wang. Font recognition based on global texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1192–1200, 2001.
- [7] Zeng Li, Yuanyan Tang, and Tinghuai Chen. Multi-scale wavelet texture-based script identification method. *Chinese Journal of Computers*, 23(7):12–18, 2000.
- [8] Li Chen and Xiaoqing Ding. Font recognition of single Chinese character based on wavelet feature. *ACTA ELECTRONICA SINICA*, 32(2):177–180, 2004.
- [9] N. E. Huang, Z. Shen, and S. R. Long et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London*, A(454):903–995, 1998.