

Off-line Handwritten Chinese Character Stroke Extraction

Feng Lin and Xiaoou Tang

Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
{flin0, xtang}@ie.cuhk.edu.hk

Abstract

Stroke extraction is of great significance for an offline character recognition system. In this paper, we present an efficient stroke extraction method based on a combination of a simple feature point detection scheme and a novel stroke segment connecting method. The algorithm can fast and accurately extract the strokes from the thinned Chinese character images. Experimental results on a large data set with over eighteen thousand character strokes achieve over 99% accuracy.

1. Introduction

Chinese characters are two-dimensional pictographic patterns. Structural analysis of Chinese characters is a natural approach for optical handwritten Chinese character recognition. A key step for structural analysis is the extraction of individual strokes from the character image.

There are three basic approaches for character stroke extraction: gray-scale image based method, binary image based method, and thinned skeleton based method. For gray-scale based method, character stroke is extracted through detecting such 3D grey-level surface features as peak, pit, saddle, and ridge etc [7]. Computational cost for this approach is generally very high. The binary-image-based method uses a chain-code to describe the character boundary contour and extracts the feature regions, such as cross sections, junction regions and end regions, for stroke extraction [2][4]. This method also involves high computational complexity since all foreground pixels in the binary image need to be considered.

In order to further reduce the amount of data that need to be processed, a thinning process is generally used to obtain the skeleton of the binary image. Skeleton-based method extracts stroke information directly from such a skeleton image [5][6][7]. In this paper, we develop new algorithms for skeleton-based approach. Especially, we address the two key processing steps for stroke extraction: feature point extraction and broken stroke connection.

For feature point extraction, Rutoviz crossing number is generally used for skeleton images [3][6][7]. However, due to problems introduced at the thinning step, Rutoviz crossing number cannot detect all the fork points in a skeleton image. Several new measures are proposed by Liu et al [6] and Abuhaiba et al [1]. However both methods tend to detect more fork points than the actual numbers. In this paper, we propose a simple approach that can remove all bug pixels introduced at the thinning step.

To connect stroke segments at a fork point, several methods have been developed in recent years. Liao and Huang develop a curve fitting method [5]. All possible pairs of stroke segments connected at the same fork point are fit by the Bernstein-Bezier curve. The connections that produce large enough curvature radius at the fork point are selected. Liu et al propose a method using polygons to approximate skeleton segments [6]. They develop a complicated set of heuristic rules according to the number of stroke segments related to a fork point. Both methods are fairly complicated, thus may not be robust enough for the large variations of handwritten Chinese characters.

In this paper, we propose a new approach based on a novel bi-directional graph to connect Chinese character strokes. Experimental results on a large data set with over eighteen thousands strokes show that our approach can link the broken stroke segments with a very high accuracy.

2. Feature Point Extraction

Rutoviz's definition of crossing number for a pixel p is,

$$N_c(p) = \frac{1}{2} \cdot \sum_{i=1}^8 |x_{i+1} - x_i|$$

x_4	x_3	x_2
x_5	p	x_1
x_6	x_7	x_8

where $x_i (i=1, \dots, 9)$ are the adjacent points of pixel p and $x_1 = x_9$ [3][6][7]. The skeleton pixels in the thinning image can be classified into 3 classes:

- If $N_c = 1$, the analyzed point is an end point;
- If $N_c = 2$, the analyzed point is a connective point;
- If $N_c > 2$, the analyzed point is a fork point.

This taxonomy is based on the assumption that the width of the skeleton is strictly one pixel. This is not always true for a thinned skeleton. Figure 1 shows several exceptions, where a skeleton has two-pixel width in the fork region and pixels in the fork region have $N_c = 2$ instead of $N_c > 2$. We find that these exceptional (bug) points can be group into two categories. The first category is shown in Fig. 1(a), where we have a four-pixel block and each pixel in the block has an 8-connected neighbour at the corners of the block. We correct this case by moving two diagonal pixels outward by one step. An example after such a correction is shown in Fig. 1(b), where all pixels in the skeleton satisfy the three-class rules.

The second category can be defined as pixels with more than two 4-connected neighbours. Figure 1 (c-f) illustrate some examples, where bug pixels are marked by bold-italic font **D** and **X**. These pixels are in the fork section but has $N_c = 2$. They each has three or four 4-connected neighbours, so we can detect them easily by locating pixels with more than two 4-connected neighbours.

In order to remove these bug pixels while preserve the skeleton continuity at the fork section, we develop a scan-line ordered deletion scheme. Scanning the skeleton image from left to right for each horizontal line, line by line from top to bottom, we delete the first bug pixel encountered. Then the next bug pixel will be checked again for the number of 4-connected neighbours. If it still has more than two 4-connected neighbours, it will be removed. However, if it no longer has more than two 4-connected neighbours because of the removal of previous bug pixel, it will be kept in the skeleton. This process is continued until all bug pixels are processed. As shown in Fig. 1, all the removed bug pixels are marked as bold-italic **D**, and all the bug pixels that are switched to normal pixels are marked as bold-italic **X**. After the correction, all pixels in the skeleton satisfy the three-class rules. Thus all the feature points of a character, including end points and fork points, can be accurately extracted.

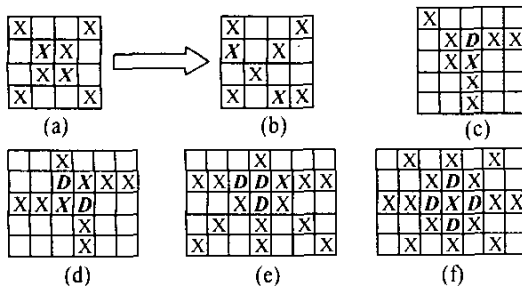


Figure 1. Example fork point bugs and their corrections.

3. Stroke Extraction

After the extraction of character feature points ($N_c = 1$ or > 2), we can trace the skeletons of the stroke segments from one feature point to the other. This simple tracing procedure is called regular stroke tracing.

Due to the thinning distortion, a fork point may be split into several fork points in a fork region, as shown in Fig. 2(a). To overcome this type of thinning distortion, we adopt the Maximum Circle Criterion [5] to group the fork points belonging to the same fork region.

At this point, we have extracted stroke segments bounded by end points and fork points. In order to obtain the complete strokes, we need to connect stroke segments that are disconnected by the fork points. To determine which pair of stroke segments needs to be connected at the fork point, we propose a bi-directional graph method.

First, we need to compute the incoming direction of all stroke segments intersecting with the fork point. Because of thinning distortion, the running direction of skeleton pixels around the fork point can be quite different from the actual orientation of the stroke segment, as shown in Fig. 2(b). To accurately estimate the stroke segment direction, only a section of skeleton pixels right outside the maximum circle of the fork point are used for direction computation, as marked in Fig. 2.

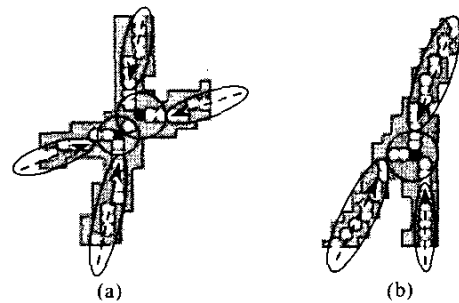


Figure 2. Thinning distortion and stroke direction estimation.

After the stroke segment direction computation, we connect stroke segments at a fork point based on a bi-directional graph. Figure 3 illustrates an example fork point with six intersecting stroke segments. The incoming directions of the six segments are shown in Fig. 3(a). We represent the directional relationship of the stroke segments by a graph shown in Fig. 3(b), where each node corresponds to a stroke segment. A directional edge is linked from one node to the other if the receiving node segment is closest to 180° from the starting node segment, among all the segments. An additional requirement is that

the angle between the two segments has to be above a threshold T_d (in our experiment, $T_d=135^\circ$).

For example, the node segment that intersects node-3 at an angle closest to 180° is node-1. Therefore, a directional link from node-3 to node-1 is established. However, for node-1, the closest opposite node is node-4, thus a directional link from node-1 to node-4 is set up. Since node-1 is also the closest opposite node for node-4, a directional link from node-4 to node-1 can also be set up. In this way, a complete relational graph for the six segments can be built as shown in Fig. 3(b). There are two bi-directional links 1-4 and 2-6, and two uni-directional links 3-1 and 5-1. Only segment pairs with bi-directional links are connected and the final stroke-connecting result is shown in Fig. 3(c).

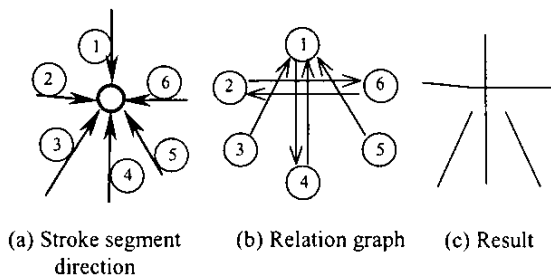


Figure 3. Stroke segment connection.

For some 3-fork points, the direction graph may fail to produce the correct connection because of the small angle differences among segment pairs, as shown in Fig. 4. To resolve this, we first detect the segment that is the closest opposite segment to both the other two segments. We call such a segment the handle of the 3-fork point. According to the orientation of the 3-fork handle, i.e. stroke A, we define four types of 3-fork structures for regular Chinese characters, as illustrated in Fig. 4. For these error-prone structures, we setup a set of heuristic rules derived from the general convention of Chinese writing to supplement the bi-directional graph method:

1. Segments B and C are on different sides of handle A.
2. The difference between the intersecting angle AC and angle AB is below a threshold.
3. Based on the orientation of the handle A, four types of solution are shown in Fig. 4 according to Chinese character convention

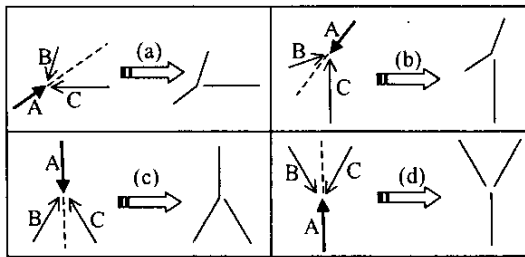


Figure 4. Heuristic connection rules for 3-fork point.

4. Experiments

To test the stroke extraction algorithm, ten students are invited to write a total of 3039 Chinese characters. The written characters are captured into character images by using a web-camera. There are 18398 strokes in these characters. For such a large data set, we achieve over 99% accuracy for the stroke extraction. Some stroke extraction results are shown in Fig 5.

The error statistics of the experimental results are summarized in Table 1. The errors can be divided into three categories:

–**Structural Error**: two independent strokes are connected because they are in line with each other and touched at the end points. In such a case, it is impossible to distinguish it from a single stroke without temporal information.

–**3-fork Error**: the heuristic rules cannot solve all the problems at a 3-fork point.

–**Broken Error**: two segments belonging to the same stroke are disconnected at a fork point. This is mainly due to that the maximum circle criteria fails to group two fork points in a fork region.

Table 1. Stroke extraction error statistics.

	Errors			
	Structural	3-fork	Broken	Total
Number	127	24	31	182
Error Percent(%)	0.690	0.130	0.199	0.989
Data	3039 Chinese characters with 18398 strokes			

5. Summary

In this paper, we present a simple and efficient scheme for off-line Chinese character stroke extraction. First, by removing two types of bug pixels in a fork region, we can directly use the Rutoviz number to detect all fork points in the character skeleton. Then using a bi-directional graph, we can connect the stroke segments at a fork point with high accuracy. Experiments on a very large data set (largest of its type to our knowledge) clearly show the efficiency of our approach. In fact, the simplicity of the new algorithm enables us to conduct experiments on such a large data set.

Acknowledgements

We thank Dr. Qiumei Yang for her comments. The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region. (Project no. CUHK4378/99E).

References

- [1] I.S. I. Abuhaiba, M. J. J. Holt, and S. Datta, "Processing of binary images of handwritten text documents," *Pattern Recognition*, Vol. 29, No. 7, pp. 1161-1177, 1996.
- [2] K. M. Ku and P. P. K. Chiu, "Fast stroke extraction method for handwritten Chinese character by cross region analysis," *Electronic Letters*, Vol. 30, No. 15, pp.1210-1212, 1994.
- [3] L. Lam, S. W. Lee, and C. Y. Suen, "Thinning methodologies - a comprehensive survey," *IEEE Trans. on PAMI*, Vol. 14, No. 9, pp. 869-885, 1992.
- [4] C. Lee and B. Wu, "A Chinese-character-stroke-extraction algorithm based on contour information," *Pattern Recognition*, Vol. 31, No. 6, pp. 651-663, 1998.
- [5] C. W. Liao and J. S. Huang, "Stroke Segmentation by Bernstein-Bezier Curve Fitting," *Pattern Recognition*, Vol.23, No. 5, pp. 475-484, 1990.
- [6] K. Liu, Y. S. Huang, and C. Y. Suen, "Robust stroke segmentation method for handwritten Chinese character recognition," *ICDAR'97*, Vol. 1, pp.211-215, 1997.
- [7] L. Wang and T.Pavlidis, "Direct gray-scale extraction of feature for character recognition," *IEEE Trans. on PAMI*, Vol. 15, No. 19, pp. 1053-1067, 1993.
- [8] J. J. Zou and H. Yan, "Extracting strokes from static line images based on selective searching," *Pattern Recognition*, 32(6), pp.935-946, 1999.

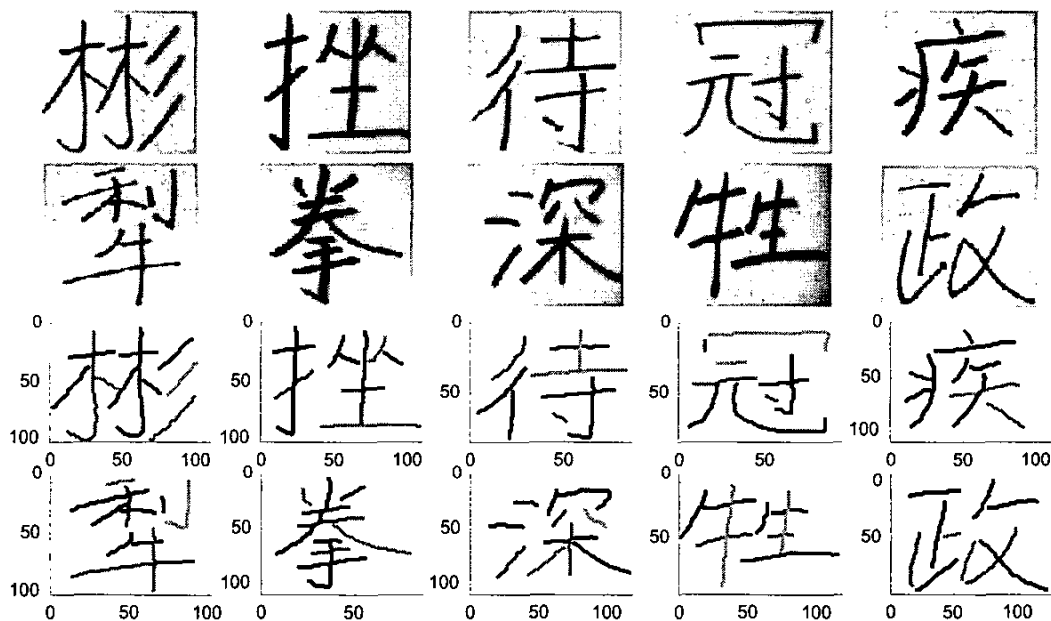


Figure 5. Example stroke extraction results and their original character images.