Graph Model Optimization based Historical Chinese Character Segmentation Method

Jingning Ji **Liangrui Peng** **Bohan Li**
Tsinghua National Laboratory for Information Science and Technology
Dept. of Electronic Engineering, Tsinghua University, Beijing, China
zjwljjn@gmail.com **plr@ocrserv.ee.tsinghua.edu.cn** **jim.jd.davis@gmail.com**

***Abstract*—**Historical Chinese document recognition technology is important for digital library. However, historical Chinese character segmentation remains a difficult problem due to the complex structure of Chinese characters and various writing styles. This paper presents a novel method for historical Chinese character segmentation based on graph model. After a preliminary over-segmentation stage, the system applies a merging process. The candidate segmentation positions are denoted by the nodes of a graph, and the merging process is regarded as selecting an optimal path of the graph. The weight of edge in the graph is calculated by the cost function which considers geometric features and recognition confidence. Experimental results show that the proposed method is effective with a detection rate of 94.6% and an accuracy rate of 96.1% on a test set of practical historical Chinese document samples.

Keywords—historical Chinese document; character segmentation; graph model

I. INTRODUCTION

Digitization of historical Chinese documents, e.g. Dunhuang manuscripts, is important for preservation, transmission and research of Chinese history. Transcribed during the 4th century to the 11th century, the Dunhuang manuscripts include more than 60,000 titles. Some of the historical documents are unique, which is of great significance to the complementary of Chinese history and Buddhist culture. With the effort of International Dunhuang Project (IDP), Dunhuang document images are shared via websites. However, most of these document images cannot be searched in full text due to the lack of recognition method of Chinese historical documents.

Character segmentation is an essential process in Chinese historical document recognition. However, the complex character structure, broken or touching characters, noise disturbance and different layouts in historical Chinese documents make it difficult to segment characters correctly. Fig. 1 shows an example of the Dunhuang documents which have the above-mentioned challenges in character segmentation.

Researchers have proposed a variety of character segmentation methods. As is discussed in [2], segmenting strategies can be categorized to "dissection", recognition-based methods, holistic methods and combinations of these three. The "dissection" method usually utilizes geometric features to find character segmentation results, which can be applied to historical Chinese character segmentation. However, purely recognition-based or holistic methods are not suitable for historical Chinese characters, as historical Chinese characters have a large character set up to tens of thousands of characters.

Most Chinese character segmentation algorithms include

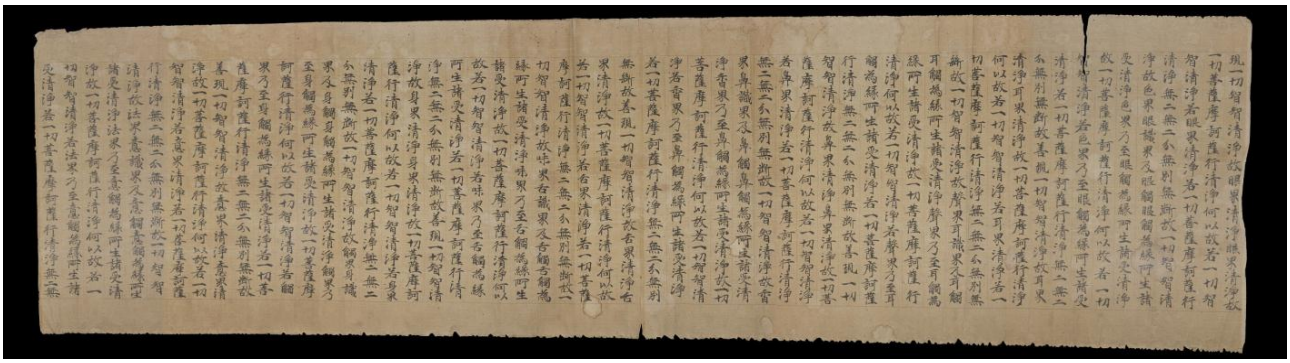


Fig. 1. An example of the Dunhuang documents transcribed during the 4th century to the 10th century.

978-1-4799-3243-6/14 $31.00 © 2014 IEEE
DOI 10.1109/DAS.2014.57

over-segmentation and merging stages. In over-segmentation stage, projection profiles, connected components, Viterbi algorithm [3, 4], Voronoi diagram [5] and stroke analysis [7, 8, 9] are mainly used.

In merging stage, a lot of algorithms have been proposed. Yi-Hong Tseng et al. [3] used the shortest path finding algorithm. Van Phan [5] proposed a criterion for removing Voronoi edges. Xiaolu Sun et al. [8] applied the Viterbi algorithm to achieve local optimization. Shuyan Zhao, et al. [9] extracted 15 geometric features for the decision tree generation to classify merging path. Several fuzzy decision rules were introduced as well to improve the performance. In the work of Lin Yu Tseng et al. [7], four kinds of merging operations are used to merge the stroke bounding boxes into candidate boxes. Then the candidate boxes are merged into characters by a dynamic programming method.

In this paper, a novel method for the merging process is proposed. The candidate segmentation positions are denoted by the nodes of a graph, and the merging process is regarded as selecting an optimal path of the graph. The weight of edge of the graph is calculated by the cost function considering component height, component gap, maximum component length, aspect ratio and recognition confidence. Different path selecting strategies is tested on practical Dunhuang historical Chinese document samples provided by National Library of China.

The rest of this paper is organized as follows: Section II presents our proposed methodology, including graph model of over-segmentation results, cost calculation and path selecting strategies. Experimental results and error analysis are discussed in Section III and conclusion is in Section IV.

II. METHOD

The overall OCR system is based on a prototype of machine-printed Chinese recognition system. The input RGB historical Chinese document image is binarized first. After binarization and noise removal, the system analyzes the layout of the image and segments the image into different regions. The regions are then segmented into text-line images. The text-line image is segmented into small components based on projection, connected component analysis and touching stroke analysis [8]. As these small components are over-segmented, a merging process is needed to find optimal segmentation results. Finally the merged characters are sent to a Chinese character recognizer to be recognized. Fig. 2 shows the process of the system.

The performance of the merging process is crucial to the overall performance of whole system. To solve the merging problem, the principle of graph theory is introduced as the following.

A. Graph model of over-segmentation results

The over-segmentation results of the text-line image can be treated as a directed graph where its segmentation positions can be treated as nodes and its small components can be treated as edges. The direction of edge is from the upper segmentation positions to the lower segmentation positions

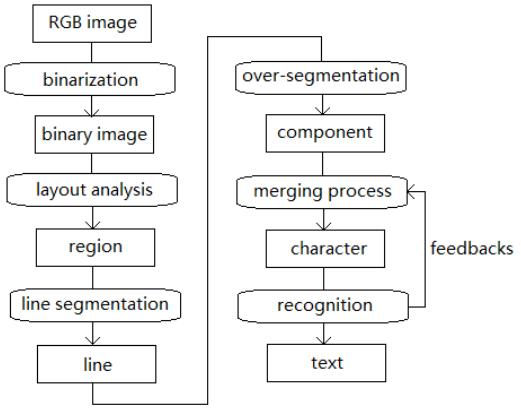
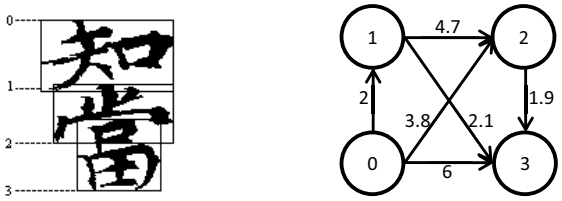


Fig. 2. The system flowchart.



(a) (b)

Fig. 3. Transform from a line image to a graph: (a) A typical line image. It is over-segmented to 3 small images, corresponding to 4 cuts including start and end. (b) The graph transformed by line image.

and the weight of edge is determined by component features. We call that the cost of a component, which describes the degree of difference between the component and a normal character. The cost should be non-negative in accordance with its meaning and for the convenience of application of graph theory algorithms. A typical transform from a part of a text-line image to a graph is shown in Fig. 3.

B. Cost calculation

Component height, component gap, maximum component length, aspect ratio and recognition feedback are considered to sum up the cost. The followings are specific costing methods.

1) Component height

The average height of all components is calculated first. Then the height of each component is divided by the average height. The normalized height h should approach to a certain value if the component is a real character. A function f(h) is designed to calculate the cost of height as shown in Fig. 4.

$$f(h)=\begin{cases}\alpha_1(h-th_1)^2 & (\mathrm{h}\le th_1)\\ 0 & (th_1<h<th_2)\\ \alpha_2(th_2-h)^2 & (\mathrm{h}\le th_2)\end{cases} \quad (1)$$

where th_1 and th_2 are lower and upper thresholds of a normal character height. α_1 and α_2 are used to adjust the proportion of cost generated by component height. The quadratic function is chosen indicating the mean square error.

For the convenience of analyzing, we set th_1=th_2=TH. As the value of TH is important for our method, it is discussed in Section III.

2) Component gap

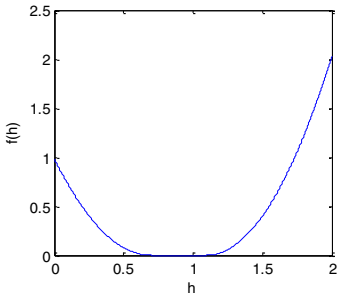


Fig. 4. An example of f(h) when th_1=0.7, th_2=1.1, α_1=2 and α_2=2.5.

It's obvious that a wider gap between components means greater possibility of a cut, indicating smaller costs of both components. As the cost should also be non-negative, the cost function of character gap S can be defined as

$$g(S_1, S_2) = \beta \times e^{-S_1/s} + \beta \times e^{-S_2/s} \quad (2)$$

where S_1 and S_2 are gaps between current component and its former and latter component, s the size coefficient and β the proportional coefficient. The parameter s relies on the character size and can be determined by the average height of all components. The parameter β relies on the intensity of characters.

It should be noticed that the gap should not be the space between bounding boxes, but the shortest distance between two black connected components.

3) Maximum component length

The maximum component length here refers to the greater value of component height and width, i.e. L=max(H,W) where H is image height and W is image width. It is used to add cost of small noises or dot strokes which have similar height to some special characters such as "—". The cost function of maximum component length can also be expressed by f(h) with a wider threshold and a smaller proportion. The parameter is decided empirically in our experiment.

4) Aspect ratio

The aspect ratio of a component is useful information when characters have different sizes but similar shapes. It is simply defined as the ratio of width to height and its cost function is similar to f(h).

5) Character recognition

Feedbacks of character recognition are of great help to describe the similarity between the component and a character. Our Chinese character recognizer is a MQDF classifier. As is discussed in [1], the following equation is an ideal estimation of recognition confidence.

$$e = 1 - \frac{d_i}{\min_{k \neq i}(d_k)} \quad (3)$$

$$d_j = \min_{1 \le k \neq K_j} \left\| y - y_k^{\,j} \right\| \quad (4)$$

where e is the confidence, i the classification result, K_j the number of prototypes in class j, y the features of image, y_k^j the features of prototype k in class j. The cost function can be

$$h(e) = \gamma(1-e) = \gamma \times \frac{d_i}{\min_{k \neq i}(d_k)} \quad (5)$$

where γ is the proportional coefficient depending on the performance of recognizer. If the recognition result is good enough, γ should be very large to reach the optimum.

The introduction of recognizer helps the system correctly merge most of characters which can be recognized, thus promoting the final recognition rate. In addition, recognizer can detect special characters such as "—" whose threshold in cost function should be adjusted to calculate a better cost.

C. Path selection

The merging process is equivalent to finding a proper path in the established graph model. The path starts from s_0 and ends at s_n and goes through edges which correspond to the merged components. If the costs of components are well calculated, the path with the minimum total cost is the optimal result. The shortest path and the minimum average cost path are considered.

Lots of shortest path algorithms have been proposed in graph theory and the Dijkstra algorithm is adopted in our method. The steps of Dijkstra algorithm are as follows.

Step 1. Let S denote the set of visited nodes and initialize S={s_0} where s_0 is the beginning node. Let T denote the set of unvisited nodes. Save the tentative distance from each node s_k to s_0 according to the equation $d(s_k)=C(s_0,s_k)$ where $C(s_0,s_k)$ means the weight of edge between s_0 and s_k, and $d(s_k)$ the tentative distance from s_0 to s_k. Set the beginning node s_0 as current node.

Step 2. For the current node s_c, calculate the tentative distance of its unvisited neighbor nodes using $d(s_k)=d(s_c)+C(s_c,s_k)$. Update $d(s_k)$ if it is less than the previously recorded tentative distance. Take the current node s_c from T to S. Choose node in T with minimum tentative distance as the current node.

Step 3. Repeat step 2 until the destination node s_n is visited. The current tentative distance $d(s_n)$ is the value of shortest path.

Sometimes the shortest path has its problem that multiple true characters have greater total cost than their merged component. The minimum average cost algorithm is proposed to overcome the disadvantage. Its process is as follows.

Step 1. Mark all the nodes unvisited except the beginning node s_0. Define $d(s_k,p)$ which means the tentative distance from s_0 to s_k passing p edges. Save the tentative distance $d(s_k,1)$ according to the equation $d(s_k,1)=C(s_0,s_k)$ where $C(s_0,s_k)$ means the weight of edge between s_0 and s_k. Save other tentative distances as infinite. Set s_1 as the current node.

Step 2. For the current node s_c, calculate all the tentative distances of its neighbor nodes using $d(s_k,p)= d(s_c,p-1)+ C(s_c,s_k)$. Update $d(s_k,p)$ if it is less than the previously recorded tentative distance passing p edges. Mark s_c as a

visited node and set the next node s_{c+1} as the current node.

Step 3. Repeat step 2 until all nodes are visited. The current tentative distance of destination node s_n passing p edges is saved in $d(s_n,p)$. Calculate the average cost of each path by dividing $d(s_n,p)$ by p and choose the minimum as the minimum average cost.

Fig. 5 shows a process of minimum average cost path selection. The original graph is Fig. 3(b). In Fig. 5(a) all tentative distances are initialized. In Fig. 5(b), s_2 and s_3 are visited and $d(s_2,2)$ and $d(s_3,2)$ are updated. In Fig. 5(c), s_3 is visited and $d(s_3,3)$ is updated. Notice that $d(s_3,2)$ is not updated because the calculated tentative distance is longer than the previous recorded distance. Finally, three average costs of $d(s_3,p)$ are compared and $d(s_3,2)/2$ is chosen as the minimum average cost.

The time and space complexity of minimum average cost path selection is a little bit higher than the Dijkstra algorithm, but it improves the performance in special cases when the characters are arranged intensively.

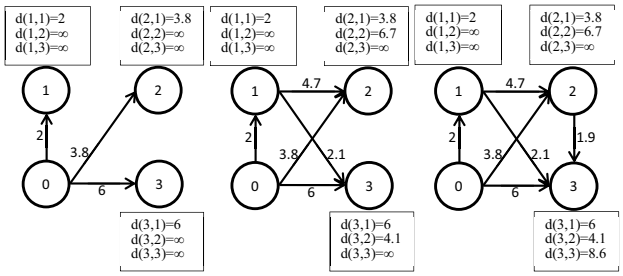


Fig. 5. Process of the minimum average cost algorithm: (a) step 1, (b) the first round of step 2 and (c) the second round of step 2.

III. EXPERIMENTAL RESULTS

A. Evaluation methods

The images are binarized and segmented into text-line images by the OCR system. Afterwards the text-line images are segmented to characters by our proposed algorithm with the minimum average cost path selection. A matching score is calculated for a segmented character and a character in the ground truth.

$$Score = \frac{S(g_i \cap d_j)}{S(g_i \cup d_j)} \qquad (6)$$

where g_i is the bounding box of a segmenting character, d_j the bounding box of a character in the ground truth and S the area of boxes. An acceptance threshold T is chosen to decide whether the two characters are matched. If Score>T, the two characters are matched, otherwise they are unmatched. The performance of segmenting result is evaluated by the detection rate and accuracy rate with different component height thresholds TH and various acceptance thresholds T.

B. Segmenting results

72 images taken from historical Dunhuang documents including 14,116 characters were tested in the experiment. The parameters α_1, α_2, β and γ in Equ. (1), (2) and (5) were set respectively according to estimation on a number of other similar historical documents. The result is shown in Fig. 6. As we can see, the detection rate reaches the maximum when component height threshold TH=0.8 and the accuracy rate reaches the maximum when TH=1. Finally the acceptance threshold is set to 0.6 due to manual evaluating experience and the height threshold TH is set to 1, where the detection rate was 94.6% and the accuracy rate was 96.1%. An example of segmenting results using our method is shown in Fig. 7. We also compared our method to a conventional projection based segmentation method, as well as our over-segmentation method without merging. Table 1 shows the result, which proves the effectiveness of our proposed method.

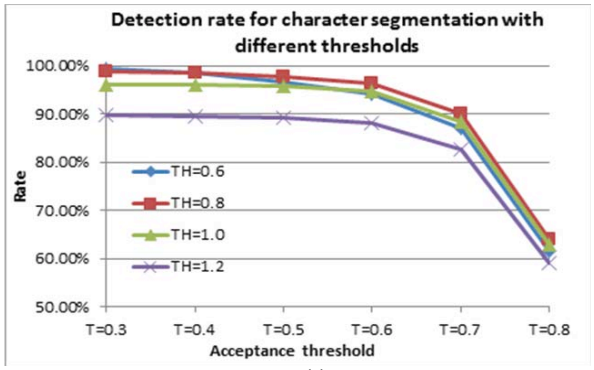


Fig. 6. Performance of our character segmentation algorithm with different thresholds: (a) detection rate and (b) accuracy rate.

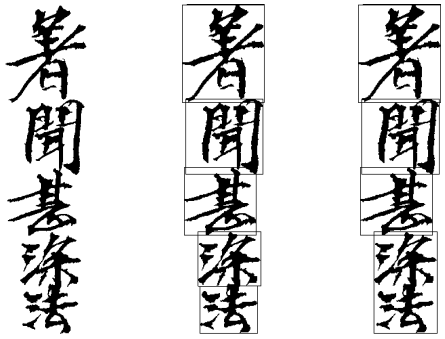
TABLE 1. COMPARISON OF OUR METHOD TO OTHERS

	Detection rate(%)	*Accuracy rate(%)*
Projection results	88.0	88.8
Over-segmentation results	90.2	71.9
Our method	94.6	96.1

Fig. 7. An example of the segmenting results of our method.

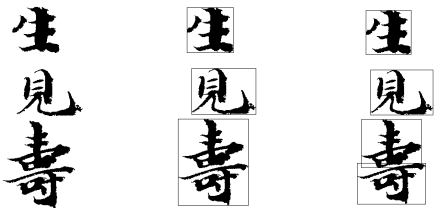
C. Error analysis

Most segmenting errors can be attributed to two reasons. The first error shown in Fig. 8 is caused by small characters connected together. The merging process merges them or the over-segmenting process overlooks them. The second error shown in Fig. 9 is caused by overlong characters. The system cuts that character to two components and decides to leave them separately. These errors are expected to be rectified by improvements of recognizer. Other errors caused by binarization or line segmentation are not presented here.



(a) (b) (c)

Fig. 8. An example of the first error: (a) a part of binary line image, (b) correct segmentation and (c) wrong segmentation.



(a) (b) (c)

Fig. 9. An example of the second error: (a) a part of binary line image, (b) correct segmentation and (c) wrong segmentation.

IV. Conclusion

In this paper, we propose a method for historical Chinese character segmentation. The proposed methodology has a remarkable performance in our samples. The experimental results prove the effectiveness of the algorithm. The method segments text-line image to characters with an over-segmenting process and a merging process. The merging process applies the minimum average cost path selection. Various geometric features and recognition feedbacks of characters are considered to calculate the costs.

Future research work will focus on two aspects, one is to develop a character recognize module with better performance on large scale historical Chinese characters. The other is improving the interacting mechanism between character segmentation and character recognition to segment touching characters.

Acknowledgment

This research is funded by National Natural Science Foundation of China (No. 61261130590, 61032008), 973 National Basic Research Program of China (No. 2014CB340500) and Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation. The authors would like to thank the National Library of China for providing historical Chinese document image samples.

References

[1] Xiaofan Lin, Xiaoqing Ding, Ming Chen, Rui Zhang, and Youshou Wu. "Adaptive confidence transform based classifier combination for Chinese character recognition." Pattern Recognition Letters19.10 (1998): 975-988.

[2] Casey, Richard G., and Eric Lecolinet. "A survey of methods and strategies in character segmentation." Pattern Analysis and Machine Intelligence, IEEE Transactions on 18.7 (1996): 690-706.

[3] Yi-Hong Tseng, and Hsi-Jian Lee. "Recognition-based handwritten Chinese character segmentation using a probabilistic Viterbi algorithm." Pattern Recognition Letters 20.8 (1999): 791-806.

[4] Min Soo Kim, Kyu Tae Cho, Hee Kue Kwag, and Jin Hyung Kim. "Segmentation of handwritten characters for digitalizing Korean historical documents." Document Analysis Systems VI. Springer Berlin Heidelberg, 2004. 114-124.

[5] Van Phan, Truyen, Bilan Zhu, and Masaki Nakagawa. "Development of nom character segmentation for collecting patterns from historical document pages." Proceedings of the 2011 Workshop on Historical Document Imaging and Processing. ACM, 2011.

[6] Nikos Nikolaou, Michael Makridis, Basilis Gatos, Nikolaos Stamatopoulos, and Nikos Papamarkos. "Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths." Image and Vision Computing 28.4 (2010): 590-604.

[7] Lin Yu Tseng, and Rung Ching Chen. "Segmenting handwritten Chinese characters based on heuristic merging of stroke bounding boxes and dynamic programming." Pattern Recognition Letters 19.10 (1998): 963-973.

[8] Xiaolu Sun, Liangrui Peng, and Xiaoqing Ding. "Touching character segmentation method for Chinese historical documents." IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2010.

[9] Shuyan Zhao, Zheru Chi, Pengfei Shi, Qing Wang. "Handwritten Chinese character segmentation using a two-stage approach." Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on. IEEE, 2001.