ATLANTIS PRESS

Advances in Intelligent Systems Research, volume 133

2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE2016)

An Efficient Character Segmentation Algorithm for Offline Handwritten Uighur Scripts Based on Grapheme Analysis

Yamei Xu* and Panpan Du

School of Compute and Communication, Lanzhou University of Technology, Lanzhou, China

*Corresponding author

***Abstract*—Cursive offline handwritten Uighur scripts contain a lot of small and random writing strokes, which makes the character segmentation is more complicated. In view of this, a new efficient character segmentation algorithm based on grapheme (part of a character) analysis is proposed in this paper. Firstly, by dot strokes detection and Component analysis, a handwritten Uighur word is over-segmented into three types of strokes: dot, affix and main strokes. Secondly, the main strokes are over-segmented but the dot strokes are clustered, so that a main graphemes queue and an additional graphemes queue are constructed respectively. Finally, the best hypothesis of characters sequence is selected by analyses of the graphemes' shapes and recognition results. Experiment results with 93.09% character segmentation accuracy rate and 97.67% recall rate have verified the validity of the proposed algorithm.**

Keywords-computer application; uighur language; handwriting recognition; character segmentation; grapheme

I. INTRODUCTION

Uighur language has an official language status in the Xinjiang Uighur autonomous region of China with a population of about 12 million [1]. In the existing literature, most of handwriting recognition studies for cursive writings have been devoted to Arabic and Farsi [2-12]. But very less research efforts have been done for Uighur scripts [13-14].

The main available approaches for cursive handwriting recognition could be divided into holistic and segmentation-driven ones. The former approaches dealt with the word image as a whole unit for recognition [3-6], while the latter first segmented word into a sequence of characters and recognized them separately [7-12]. The segmentation-driven approaches were more favorable for large-scale classification. Thus our work adopted these approaches.

The typical cursive character segmentation approaches were usually based on image analysis, with several important features including projection histogram [7, 9-10], contour feature [8, 11] and skeleton feature [12]. Uighur characters have more complex dots and diacritics than Arabic, thus above techniques developed for Arabic handwriting could not be directly implemented in Uighur recognition. The segmentation results showed more false segmentations for especially those Uighur characters which have many small and random writing strokes.

In basis of the above analysis, this paper proposes an effective segmentation algorithm for offline handwritten Uighur words. The algorithm firstly over-segments a word image into three types of strokes: dot, affix and main strokes. Then, to avoid the false segmentations due to incorrect matching of the dot and affix strokes, the algorithm is designed to over-segment the main strokes and cluster the dot and affix ones respectively to get two types of graphemes(part of a character) queues. One of them is called the main graphemes queue, and another the additional one. Finally, the graphemes are matched and merged based on grapheme shapes and recognition results to obtain a segmented characters sequence.

The rest of this paper is organized as follows. Section II summarizes the important characteristics of Uighur handwritten scripts. Section III gives a flowchart of the proposed segmentation algorithm, and details the descriptions of several method modules implemented. Section IV presents the experiments performed and gives the result analysis. Finally, some concluding remarks and perspectives are presented in section V.

II. UIGHUR SCRIPT CHARACTERISTICS

The modern Uighur language was derived from the Arabic and Chagatai alphabets. The 32 basic shapes of Uighur letters are shown in Figure I, in which these characters indicated by '*' are absent in Arabic alphabet set. According to their different positions in the word, the 32 basic letters had been evolved into 128 characters.

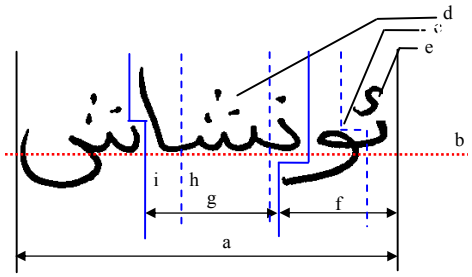
ن	م	س	ب	ز	ت	ج	ش
ي	ف	ه	ك	غ	خ	ق	ر
ل	ا	ە	ى	و	ﻗ*	چ*	پ*
گ*	ژ*	ې*	ھ*	ۋ*	ۇ*	ۆ*	ۇ*

FIGURE I. BASIC SHAPES OF UIGHUR LETTERS

Word is the smallest linguistic unit of Uighur script which can be independently used, between words there are obvious spaces. An example for structural rules of Uighur word is shown in Figure II, several most prominent characteristics are described as follows:

- Uighur scripts are inherently cursive and written from right to left at an imagined horizontal line, which is called "baseline".

- For each word, some strokes written along baseline are called “main strokes”, same very small strokes are called “dot strokes”, and other zigzags and diacritics are called “affix strokes”.
- Depending on where it appears in the word, each letter may have 2 to 8 variant forms. Plus to syllable-dividing mark “ئ” (“ـئـ”) and conjoined symbol “لا” (“ﻟﺎ”), 32 basic letters have been evolved into 128 characters.
- When characters join to each other the “ligatures” are made. Each word may include one or more ligatures, and each ligature may include one or more characters.
- Characters adjoined in the one ligature are connected by their main strokes. Characters without touching may overlap vertically.



a. word ئونشاش b. baseline
c. main stroke d. dot stroke e. affix stroke
f. ligature including 2 characters
g. ligature including 3 characters
h. segmentation boundary within ligature
i. segmentation boundary between ligatures

FIGURE II. AN EXAMPLE FOR STRUCTURAL RULES OF UIGHUR WORD

III. DESCRIPTION OF THE PROPOSED SEGMENTATION ALGORITHM

A. The Flowchart of the Proposed Algorithm

Character segmentation techniques available in the present literatures usually split the word image vertically along the cut-off points [7-12]. Uighur handwriting has many dot and affix strokes, and their writing positions more casual (for example characters: “ڭ”, “ۇ”, “ۆ”, “ج”, “پ”etc.). Thus the relative positions between main and additional strokes (dot and affix strokes) often vary a lot in the handwriting script, which may produce the stroke drift phenomenon. Hence, direct segmentation is likely to misclassify the additional strokes which belong to one same character into two different characters.

To overcome the mentioned deficiency, in this paper we proposes a new character segmentation algorithm for Uighur handwritten scripts. The algorithm first extracts the main, dot and affix strokes from a word through baseline region detection combined with component analysis. Then the main strokes are over-segmented and the dot and affix strokes are clustered to form two types of graphemes queues. Finally a segmented characters sequence will be obtained by the analyses of grapheme shapes and recognition results.

The processing flowchart of the proposed character segmentation algorithm is visualized in Figure III. The operations of strokes extraction and main strokes segmentation will be detailed in Figure IV and Figure V respectively.

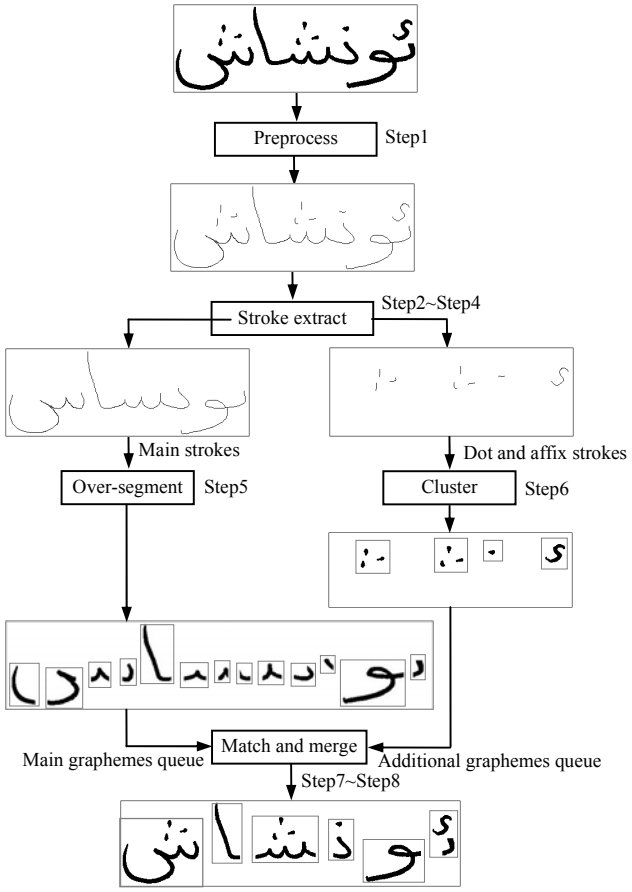


FIGURE III. FLOWCHART OF CHARACTER SEGMENTATION ALGORITHMWORD

B. Descriptions of the Proposed Algorithm

The concrete steps of the proposed character segmentation algorithm are described as follows:

1) Processing:

The step mainly includes slant correction, smoothing, thinning, normalization and broken connection, not go into details here.

2) Dot strokes detection:

Let $\boldsymbol{S} = (S_1, S_2, \ldots, S_r)$ be the strokes in a word, for the default dot threshold T, we determine one stroke S_i as a dot stroke when the component area of the stroke satisfies $A_C(S_i)<T$. All dot strokes are record as $\boldsymbol{P} = (P_1, P_2, \ldots, P_l)$.

3) Baseline region detection:

The baseline B and its baseline region $[B_u, B_l]$ are detected using the remained strokes $\boldsymbol{S}_R = \boldsymbol{S}–\boldsymbol{P}$. The computation is given by (1), where B_u and B_l are the upper and lower edges of the baseline region respectively.

Where $H(j)$ represents the horizontal projection of S_R, and W and L are the worth and height of S_R, known as in (2).

$$B = \arg_j \left[\max H(j)\right],$$
$$B_u = \arg_k \left[\sum_{j=k}^{B} H(j) = \sigma \sum_{j=0}^{B} H(j)\right] \quad (1)$$
$$B_l = \arg_k \left[\sum_{j=B}^{k} H(j) = \sigma \sum_{j=B}^{L-1} H(j)\right]$$

$$H(j) = \sum_{i=0}^{W-1} \boldsymbol{S}_R(i,j), \quad j = 0,\ldots,L-1 \quad (2)$$

Factor σ determines the area of the baseline region, and we have an experienced value: $\sigma = 0.8$.

4) Component analysis:

Let S_i and S_j be any two strokes in $\boldsymbol{S}_R$, and $W(S_i)$ and $W(S_j)$ are their widths of circumscribed rectangle box, if the overlap satisfied: $O_w(W(S_i), W(S_j)) > (2/3) \times min(W(S_i), W(S_j))$, then S_i and S_j are regarded as overlap each other. The relatively farther one of the two strokes from the baseline (assumed it is S_i) is detected. If S_i isn't within the baseline region, then it is classified into the affix strokes $\boldsymbol{S}_A$.

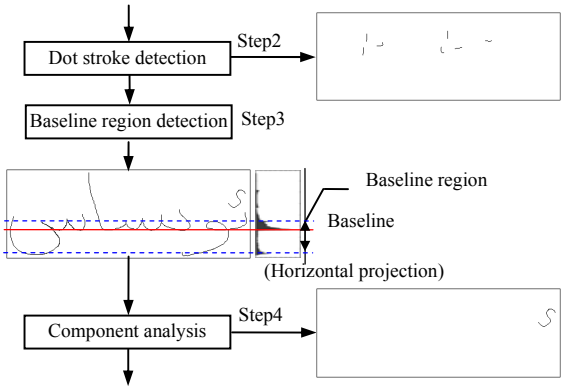


FIGURE IV. DIAGRAM OF STROKE EXTRACTION

5) Main strokes over-segmentation:

For the main strokes $\boldsymbol{S}_M = \boldsymbol{S}_R - \boldsymbol{S}_A$, we calculate their vertical differential projection [9] within the baseline region. Take the minimum points as cut-off points, $\boldsymbol{S}_M$ are over-segmented vertically to obtain the main graphemes queue $\boldsymbol{M} = (M_1, M_2, \ldots, M_n)$.

6) Dot and affix strokes clustering:

The dot strokes $\boldsymbol{P}$ are clustered by the max-min sequential clustering algorithm. Based on the rule that all dot strokes in one character lie in the same side of the baseline, the distance measure between one dot stroke P from the cluster $\boldsymbol{C}$ is defined as,

$$D(P,\boldsymbol{C}) = \begin{cases} +\infty, & [Y(P)-B][Y(\boldsymbol{C})-B] < 0 \\ |X(P)-X(\boldsymbol{C})|, & otherwise \end{cases} \quad (3)$$

Where $X(\cdot)$ and $Y(\cdot)$ respectively represent the x and y-axis coordinate of the centre of a dot stroke or a cluster, and B is the baseline position. The presentation ordering is from right to left along the x-axis. After clustering dot groups as a whole, we join the dot strokes $\boldsymbol{P}$ and the affix strokes $\boldsymbol{S}_A$ to obtain the additional graphemes queue $\boldsymbol{A} = (A_1, A_2, \ldots, A_m)$.

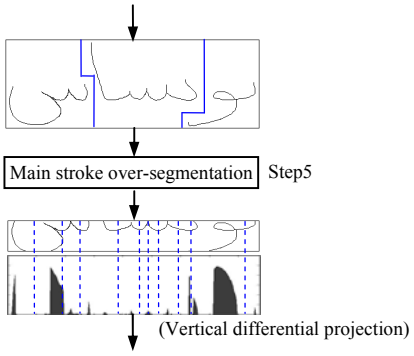


FIGURE V. DIAGRAM OF MAIN STROKE OVER-SEGMENTATION

7) Graphemes matching:

With the principle of nearest neighbor, a graphemes queue $\boldsymbol{G} = (G_1, \ldots, G_n)$ is obtained through matching the main graphemes queue $\boldsymbol{M} = (M_1, \ldots, M_n)$ with the additional graphemes queue $\boldsymbol{A} = (A_1, \ldots, A_m)$.

8) Graphemes merging:

First, the graphemes G_i and G_{i+1} are merged to construct a new grapheme when the shape of G_i satisfies the condition given by (4). Where $H(\cdot)$ and $W(\cdot)$ represent the height and width of the grapheme respectively, and $X_p(\cdot)$ and $X_c(\cdot)$ represent the x-axis coordinate of the peak and the center of the grapheme respectively, as shown in Figure VI. Factors λ_1, λ_2 get experienced values: $\lambda_1 = 0.3$, $\lambda_2 = 0.1$.

$$\left[\left|X_c(M_i) - X_p(M_i)\right| > \lambda_1 W(M_i)\right], \quad 1 \le i \le m-1 \quad (4)$$
$$and \left[H(M_i) < \lambda_2 H(\boldsymbol{M})\right]$$

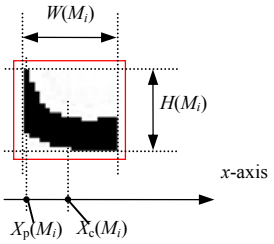


FIGURE VI. SHAPE ANALYSIS OF THE GRAPHEME M_i

Second, as Figure VII exhibits, the 8 combination types of Uighur characters are summarized. These characters may be over-segmented into the graphemes "ـد", "ـه", "ں" or "ـڈ". Thus, the final character sequence $\boldsymbol{C} = (C_1, \ldots, C_{n'})$, $n' \le n$ will be obtained by recognize and merge these special graphemes.

س س سه سه ش ش شه شه

سه = ـه + ـه + ـد شه = ـه + ـڈ + ـد
سه = ـه + ـه + ـه شه = ـه + ـڈ + ـه
س = ں + ـه + ـد ش = ں + ـڈ + ـد
س = ں + ـه + ـه ش = ں + ـڈ + ـه

FIGURE VII. THE 8 COMBING TYPES OF UIGHUR CHARACTERS

IV. EXPERIMENTS AND RESULTS DISCUSSION

Since that there were almost no articles about Uighur character segmentation in the present, we referred to some approaches for Arabic handwritten script segmentation. The database we used contained 500 Uighur word classes, each class had 25 samples, for a total of 12500 samples. The test method used was the leave-one-out cross-validation procedure. The experiments used Visual C++6.0 for programming. The operating environment was the PC of Intel i5-4300M CPU with 4.0GB memory.

In order to assess the character segmentation results, we used the following three measurement criteria: accuracy rate, recall rate and false positive rate. Among which the accuracy rate referred to the ratio of the correct segmentations to all segmentations detected. The recall rate referred to the ratio of the correct segmentations found to all real segmentations. And the False positive rate equaled 1 – Accuracy rate.

Three character segmentation algorithms had been compared in this experiment. Algorithm 1 was the proposed Uighur character segmentation method. Algorithm 2 presented in [8] first over-segmented the characters and then optimized segmentation path based on geometrical layout information. Algorithm 3 detailed in [9] was a character segmentation approach, which validated prospective segmentation points through fusing confidence values by the right and center character recognition outputs.

Performances of the above character segmentation algorithms are listed in Table I. As can be seen, our character segmentation approach (Algorithm 1) performs quite well with 93.09% accuracy rate and 97.67% recall rate. Analyzing and comparing with the other two algorithms, Algorithm 2 selects the graphemes merging path through combining the information of geometrical layout information. The accuracy rate of the algorithm is 88.79%. Information considered by the algorithm is more single, perhaps this is the reason for its little poor performance. Algorithm 3 validates prospective segmentations by fusing confidence values by the right and center character recognition outputs. The segmentation algorithm achieves 91.36% accuracy rate. But only involving the relevant recognition information may make the segmentation result depends too heavily on the character classifier.

In addition, sometimes Uighur character segmentation may encounter some irregular cases that our approach is still unable to solve. One case is when the characters "ـه" and "ـشه" are neighbors and the additional strokes of the character "ـشه" are written drifted. Its example is seen in the word "پـشـشـق". A false segmentation occurs between the characters "ـه" and "ـشه".

Another case happens when the character "ى" appears immediately after the character "ـلـ", and unluckily the "ـلـ" is easily written titled to the left according to the Uighur writing habits. Its example is seen in the word "ئەسلى". In which one valid segmentation point between the characters "ى" and "ـلـ" is missed. To further improve the performance of our algorithm, these special cases remained will be studied more comprehensively.

TABLE I. CHARACTER SEGMENTATION PERFORMANCES COMPARISON

Algorithm	Segmentation performances			
	Accuracy rate (%)	*Recall rate (%)*	*False positive rate (%)*	*Run time (ms/word)*
1	93.09	97.67	6.91	587
2	88.79	93.96	11.21	766
3	91.36	95.46	8.64	813

V. CONCLUSION

This paper presents an effective character segmentation algorithm for offline handwritten Uighur words. The most interesting contribution in this work is the path optimization strategy for character segmentation based on grapheme analysis. In the experiments the proposed algorithm performs with 93.09% character segmentation accuracy rate and 97.67% recall rate. The results are quite favorable and better than the other competing techniques. Moreover, the training data required by the system is limited to the fixed grapheme classes, so that the scalability of the proposed algorithm to large scale lexicon application is stronger.

Since the segmentation errors found in the experiment results are mostly due to the Uighur special structure rules and some bad writings, our future research works will include two main aspects. First, the effective correction of preprocessing technique will be enhanced for adhered or broken strokes. Second, the Uighur special rules and writing styles will be further investigated, and more structural information will be

mined and utilized accordingly to improve the algorithm performance.

ACKNOWLEDGMENT

The authors acknowledge the financial support of this work by grants from National Natural Science Foundation of China (No. 61562058).

REFERENCES

[1] Mamat Sadik, Basics of Uighur Language, 1st ed., Urumqi: Xinjiang People's Press, 1992.

[2] Rehman A, Saba T, "Off-line cursive script recognition: current advances, comparisons and remaining problems," Artificial Intelligence Review, vol. 37, no. 4, pp. 261–288, 2012.

[3] Kazim Fouladi, Babak N. Araabi, Ehsanollah Kabir, "A fast and accurate contour-based method for writer-dependent offline handwritten Farsi/Arabic subwords recognition," International Journal on Document Analysis and Recognition, vol. 17, pp: 181–203, 2014.

[4] Yousef Elarian, Irfan Ahmad, Sameh Awaida, et al, "An Arabic handwriting synthesis system," Pattern Recognition, vol. 48, pp: 849–861, 2015.

[5] Jafaar Al Abodi, Xue Li, "An effective approach to offline Arabic handwriting Recognition," Computers and Electrical Engineering, vol. 40, pp: 1883–1901, 2014.

[6] Saeeda Naz, Arif I. Umar, Syed H. Shirazi, et al, "Segmentation techniques for recognition of Arabic-like scripts: A comprehensive survey," Education and Information Technologies, vol. 21, pp: 1225-1241, 2016.

[7] Elzobi M, Al-Hamadi A, Al Aghbari Z, et al, "IESK-ArDB: a database for handwritten Arabic and an optimized topological segmentation approach," International Journal on Document Analysis and Recognition, 2012.

[8] Ding Xiaoqing, Liu Hailong, "Segmentation-driven offline handwritten Chinese and Arabic script recognition," Lecture Notes in Computer Science, Springer Press, 2008, vol. 4768, pp: 196-217.

[9] Al Hamad H A, Zitar R A, "Development of an efficient neural-based segmentation technique for Arabic handwriting recognition," Pattern Recognition, vol. 43, no. 8, pp: 2773-2798, 2010.

[10] Al Hamad H A, "Over-segmentation of handwriting Arabic scripts using an efficient heuristic technique," Proceedings of the 2012 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), IEEE Press, 2012, pp: 180-185.

[11] Parvez M T, Mahmoud S A, "Arabic handwriting recognition using structural and syntactic pattern attributes," Pattern Recognition, vol. 46, no.1, pp: 141-154, 2013.

[12] Abandah G A, Jamour F T, "Recognizing handwritten Arabic script through efficient skeleton-based grapheme segmentation algorithm," Proceedings 10th International Conference on Intelligent Systems Design and Applications (ISDA). IEEE Press, 2010, pp: 977-982.

[13] Zhao Hui, Han Linfeng, Su Hao, "An online recognition algorithm of handwritten Uighur characters," 2011 Seventh International Conference on Natural Computation (ICNC). IEEE Press, 2011, pp: 1616-1619.

[14] Wang Hua, Ding Xiaoqing, Halimurat, "Multi-font multi-size printed Uighur character recognition," Journal of Tsinghua University,vol.44, no. 7, pp: 946-949, 2004.