A Dynamic Programming Method for Segmentation of Online Cursive Uyghur Handwritten Words into Basic Recognizable Units

Mayire Ibrayim
Institute of Information Science and Engineering, Xinjiang University, Urumqi, 830046, P.R. China.
Email: mayire401@gmail.com

Askar Hamdulla*
College of Software, Xinjiang University.Urumqi, 830046, P.R. China.
Email: askar@xju.edu.cn

Dilmurat Tursun
Institute of Information Science and Engineering, Xinjiang University, Urumqi, 830046, P.R. China.
Email: tursundilmurat@gmail.com

***Abstract*—Correct and efficient segmentation of Uyghur words into characters is crucial to the successful recognition. However, little work has been done in this area. There are many connected characters in cursive Uyghur handwriting, which makes the segmentation and recognition of Uyghur words very difficult. To enable large vocabulary Uyghur word recognition using character models, we propose a character segmentation method using dynamic programming in online cursive Uyghur handwriting. Firstly, after removing delayed strokes from the handwritten words, potential breakpoints are detected from concavities and ligatures by temporal and shape analysis of the stroke trajectory. Then, a dynamic programming method is applied to find the best segmentation point for each character. Our preliminary experiments on an online Uyghur word dataset demonstrate that the proposed method can achieve good performance in segmenting cursive handwritten Uyghur characters.**

***Index Terms*—Online Uyghur Handwriting, Character Over-Segmentation, Character Recognition, Dynamic Programming**

I. INTRODUCTION

With the widespread use of computing devices during the last decades, the need for fast and efficient text input measures is increasing. Handwriting recognition offers an important component in building such interfaces [1]. There has been considerable attention in the area of handwriting recognition for Latin-based and Oriental languages [2-5]. Despite that Uyghur script is widely used in regions of minority nationalities and the automatic recognition of handwritten Uyghur characters has many potential applications, the research on this problem has received little attention. The difficulties inherent in segmenting Uyghur words include character connectivity, position-depended character shaping and delayed strokes. This work considers online handwritten Uyghur word recognition and designs a character separation algorithm, which is an important step for word recognition.

Correct and efficient segmentation of words into characters is considered to be a fundamental problem in handwriting recognition. There are two approaches for handwritten word recognition: global approach and analytical approach [6]. The global approach treats the word as a whole [7], while the analytical approach decomposes the word into smaller units or characters. Analytical approach is suitable for large word vocabularies because the number of character classes is limited [8]. Dynamic programming algorithm has been widely used in many researches with recognition-based segmentation of Chinese and English character. The basic idea is to use a dynamic programming algorithm to find a globally optimal path of segments. References [9-16] proposed segmentation approaches based on a projection analysis and dynamic programming (DP). In a first step, the projection analysis is used to find candidate segmentation points. And then, by using recognition and language model information, a DP algorithm is applied to find most reliable segmentation paths. We adopt the analytical approach for handwritten Uyghur word recognition and need an effective algorithm for separating the characters in cursively written words. Many works have been done in handwriting segmentation [17-19] but very few were for the segmentation of cursive Uyghur words [20]. To overcome the uncertainty of character segmentation before recognition, we adopt the strategy of

Manuscript received September 12, 2012; revised December 21, 2012; accepted April 25, 2013.

*Corresponding author: Askar Hamdulla (askar@xju.edu.cn)

doi:10.4304/jsw.8.10.2535-2540

over-segmentation for cursive word recognition: separate the word image into primitive segments, with each segment being a character or a part of character. After over-segmentation, correct characters can be formed by concatenating consecutive primitive segments. The segmentation of Uyghur words is difficult due to the characteristics of the language [21] and the variable styles of writing. Particularly, Uyghur characters have variable character sizes and gaps, and some characters have many variations of shapes depending on the position in words. In cursive writing, the characters in a word are mostly connected.

Based on the characteristics of cursive handwritten Uyghur words, our method first removes delayed strokes from the words, then potential breakpoints are detected from concavities and ligatures by temporal and shape analysis of the stroke trajectory. Afterwards, some redundant segmentation points are removed in a heuristic filtering step. Finally, a dynamic programming method is applied to find the best segmentation path. Our preliminary experiments on online handwritten Uyghur words demonstrate that the proposed method can give a high recall rate of segmentation point detection.

The rest of this paper is organized as follows. Section 2 reviews the characteristics of Uyghur script. Section 3 describes the overview of segmentation system. Section 4 describes the proposed character over-segmentation method. Section 5 describes a dynamic programming method applied to find the best segmentation past. Experimental results are given in Section 6 and we conclude the paper in Section 7.

II. The Characteristics of Uyghur Script

Uygur language is a Turkish language used in the Xinjiang Uygur autonomous region in China. Uyghur writing is based on an alphabet and rules different from those of Chinese and Latin languages. Uyghur characters are written in a cursive style from right to left and no upper or lower case exists. Its alphabet contains 32 kinds of characters, and each character has two to four shapes and the choice of which shape to use depends on the position of the character (within its word or pseudo-word). Start form: only the suffix connects with the next character; Middle form: initial and suffix connect with adjacent characters; End form: only the initial connects with the above character; Isolated form: initial and suffix does not connect with adjacent character. Many characters have a similar shape. The position or number of these secondary strokes makes the only difference.

The word consisting of a sequence of disjoint connected components is called pseudo-words. It has a main stroke that includes its basic shape, and complementary stokes which include dots or complementary parts. A Uyghur word can have one or more pseudo-words. Each pseudo-word can be a group of characters or one character.

For character separation, we found that it's related with the writing style. The connecting types of different writing styles of Uyghur characters are as follows: ① Ligature: In order to make a word, all characters connect directly to the characters which immediately follow along a writing line or baseline. Some combination of two characters has special shapes called "ligatures". ② Concavity: Even for the same word, the different writers may have different writing style, resulting in different word shape such as concavity. ③ Overlap: Overlap refers to writing style of the points which have above or below writings in the same stroke. Characters in a word often overlap due to the writing styles.

III. Segmentation System Overview

The block diagram of our character segmentation system is shown in Figure1. Firstly, after removing delayed strokes from the handwritten words, potential breakpoints are detected and extra breakpoints are filtered using knowledge of character shapes. Then, delayed strokes are reconstructed. In over-segmentation stage, this module over-segments the word pattern into a sequence of primitive segments. Finally, a dynamic programming method is applied to find the best segmentation path.

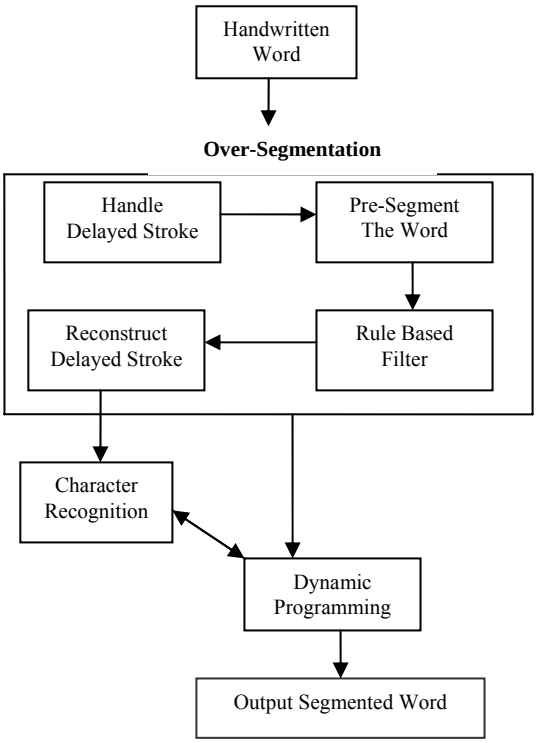


Figure1. System diagram of handwritten Uyghur word segmentation.

IV. Over-Segmentation

A. Removing Delayed-Strokes

A stroke is defined as all-data point samples written between a certain pen-down action and the following pen-up action. Important information from the digitizing hardware other than the point coordinate pairs is the pen-down and the pen-up signals. Each group of strokes contain the primary strokes and secondary strokes which we call in this paper" delayed strokes", regardless the

order by which they were written. Thus, a stroke may represent a pseudo-word or a character, or sometimes even a dot.

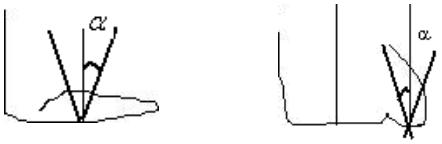
Detecting and removing delayed stroke is an important step. Delayed strokes are detected using a holistic approach. In order to examining the states of successively written Uygur strokes (either primary stroke or delayed stroke like dots for example), the following geometry features are calculated for each stroke such as width and height of bounding box and distance of overlapping between two strokes. If the value of the feature is less than predefined threshold, then the stroke is delayed-stroke.

B. Pre-Segmentation

In this section, after removing delayed strokes from the handwritten words, potential breakpoints are detected from concavities and ligatures by temporal and shape analysis of the stroke trajectory. Over segmentation is denoted when the character is segmented into several primitive segments. The over segmentation follows the steps below:

Step1: Formation of Initial Separation Point. For every right-to-left line $\overline{P_iP_{i+1}}$ in the stroke, P_i will be considered as an initial separation point between characters if the angle between $\overline{P_iP_{i+1}}$ and the horizontal axis is smaller than $\frac{\pi}{6}$.

Step2: Handling Space Overlapping. This stage will eliminate some of the elements in the group of initial separation points. If one of initial separation points has no any above writings across the whole range defined by the angle(90- α)o to (90+ α)o, then this point is accepted, else this point is rejected from the group of initial separation points (α is 25oempirically), as shown in Figure2.



(a) (b)

Figure2. (a) Rejected segment point; (b) Accepted segment point.

Step3: Generating Separation Section. We compute the horizontal distance between every two initial separation points. If the distance is smaller than a predefined threshold (10 empirically), these points will form a bigger segmentation section.

Step4: Locating Segmentation Points. In this stage our algorithm finds K possible separation sections in the main stroke. The middle of every separation sections will be located as segmentation points S_i (i=1, 2,⋯, k).

C. Rule-based Filtering

In this section, the suggested segmentation point is passed through the rule-based filtering to discard the incorrect segmentation point.

Rule1: baseline is computed with the respect to the horizontal pixel density. The baseline corresponds to the original writing line in which all the connection between the successive characters take place. For handwriting, the base line is an ideal concept and simplification of actual writing. In practice connections occur near, but not necessarily on a baseline. Compute the vertical distance D between suggested segment point S_i and the y-value of the baseline on a word. If the distance is less than a predefined threshold (10 empirically), then filter the segmentation point.

Rule2: If the distance between two suggested segmentation points is less than a predefined threshold (5 empirically), remove the segmentation points.

D. Reconstruct Delayed Stroke

The last step left in this stage is reconstruction of delayed strokes. Delayed strokes are essential to distinguishing among various Uyghur characters. Thus, handling delayed strokes correctly is vital for recognition of the segmented character. Generally, delayed stroke is written after completing the main stroke in each connected component. In the Uyghur script, delayed strokes are written above or blew the segmented character block and could appear before, after, or within the segmented character block with respect to the horizontal axis. A bounding box of a segmented block is specified by the coordinates of left, right, top and bottom boundaries. Therefore, in accordance with the written order of each stroke, compute the overlapping degree between the segmented primitive and the delayed stroke. Denote the leftmost and the rightmost of segmented primitive block and delayed stroke block as (x_s^L, x_s^R) and (x_d^L, x_d^R), respectively. If $x_s^L < x_d^R$ and $x_s^R > x_d^L$, they are overlapping and the delayed stroke is belong to the segmented primitive block.

V. SEGMENTATION USING DYNAMIC PROGRAMMING

After the above over-segmentation processes, we obtained a sequence of primitive segments. Some primitive segments contain a single Uyghur character; some contains only components of character which need to do a further consolidation. In this section, the dynamic programming method is used to further merge the candidate segmented block that are bounding boxes of components of characters.

The handwritten word image is represented in a sequence of segmented blocks, which are ordered from right to left. The segmentation points that separate the segmented block sequence is denoted as $\{P_0, P_1, \ldots, P_N\}$. One or more consecutive segmented blocks may was combined to generate one candidate Uyghur character pattern. So a candidate character pattern composed of T

consecutive segmented blocks between the segmentation points P_{i-1} and P_{i+T-1}. To reduce the complexity of Uyghur character structure, the number of consecutive segmented blocks is considered to prohibit some neighboring blocks from combining. One to at most three of consecutive segmented blocks can be combined into a candidate character pattern. These candidate segmented patterns compose a candidate segmentation path and these candidate segmentation paths are represented in a candidate segmentation network. A candidate segmentation network is constructed based on the result from the combination of over-segmentation as shown in Figure3. Each segmentation point can be seen as a node and each group node represents a candidate path. Each edge which connects two nodes can be seen as a candidate character pattern.

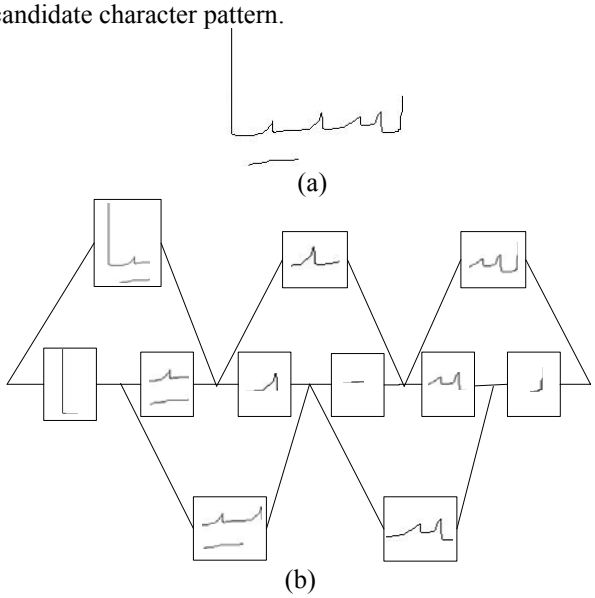


Figure3. (a) Original word; (b) An example of candidate segmentation network

We are applying the dynamic programming method to find the globally optimal segmentation boundaries based on the given candidate segmentation network. optimal segmentation boundaries refer to the minimum cost path from one of the start nodes to one of the terminal nodes. Its core idea of dynamic programming is to calculate the cost through the input word which minimizes the defined cost function for different segmentation paths, in order to find the minimal cost for each possible segmentation paths. The segmentation path is evaluated by the classification scores of their constituent candidate character pattern. The cost in each edge is a function of character recognition distances. In the context of character segmentation, character classifier must be trained with a variety of characters. The character classifier (MQDF) [22] is trained on the Uyghur character database. Each character pattern, with the pseudo-two-dimensional Moment normalization (P2DMN) method [23] for normalization and a normalization-cooperated method for 8-direction feature extraction [24], is represented by a 200-dimensional feature vector. The feature vector is reduced 120D by fisher linear discriminant analysis for accelerating classification.

The optimal path, in the sense of minimum cost, gives the segmentation result. Each possible group of candidate segmented block will be sent to classifier. After recognized with single handwritten Uyghur character recognizer, the recognition distance of each candidate pattern becomes the cost of the corresponding edge of this candidate pattern. A set of candidate segmentation paths in the candidate segmentation network can be represented as $\{Sp_0, Sp_1, \ldots, Sp_K\}$, where Sp_i is a candidate segmentation paths and K is the number of segmentation paths. The optimal segmentation path from the start nodes to the terminal nodes in candidate segmentation network can be obtained by checking these candidate segmentation paths.

VI. Experimental Results

A. Acquisition of Online Handwritten Word Data

Acquisition of online handwriting data is the first step of Uyghur online handwritten recognition system. As we know, there is no publicly available online Uyghur database. In this paper, the database we used for testing consists of 900 words collected from different people (300 words written by each writer) using Han Wang writing tablets. The handwriting input is captured as a stream of positions in the form of "x" and "y" coordinates. Depending on the type of a digitizer it may be able to provide more information such as pen-pressure and pen-tilt and with programming one can also compute the pen movement speed. Most systems including ours, however, use only the coordinates and pen-up/down signals. The information of the word image is saved as a format of binary file.

The number of characters of in a word varies from 2 to 15 as summarized in table 1. Set 1 is written in a more regularly style and set 2 is written in a normal while the third is written more free.

TABLE I.
THE NUMBER OF TEST WORD (ONE SET)

Number of characters	2—5	6—10	11—18	Total Word	Total character
Number of test word	65	203	32	300	1898

The algorithms were implemented in C++ in platform of Microsoft visual C++ 6.0.

B. Experimental Results of Segmentation

To examine the utility of our approach, we have carried out experiments on the word data sets explained in the previous section. For the small sample test, the character segmentation algorithm is usually evaluated with the help of manual statistics; for the large sample image test, the final recognition accuracy of the system applied is often used to evaluate the performance of the segmentation algorithm. In this paper, the first method is adopted. For the over-segmentation stage, we evaluate the

performance of character detection in terms of the rates of Recall (R) and Precision (P) as well as the harmonic average (F-measure), which are defined as:

$$\text{Recall}=\frac{\text{number of correctly detected seperation point}}{\text{number of true seperation point}}*100\%$$

$$\text{Precision}=\frac{\text{number of correctly detected seperation point}}{\text{number of detected seperation point}}*100\%$$

$$\text{F-measure}=\frac{2*\text{Recall}*\text{Precision}}{\text{Recall}+\text{Precision}}*100\%$$

Experimental results of our over-segmentation approach and segmentation using DP approach on different sets for online handwritten Uyghur word are listed in Table2 respectively. The results shows that comparing with over-segmentation approach, segmentation using DP approach improves P(56.13%→71.97%, 52.87%→67.72%, 49.73%→63.41%), with the high recall rate (R). The improvement of P will decrease the number of primitive segments, which helps to improve the performance of word recognition. The main reason of wrong segmentation is the consequence of two problems. One is problem of over-segmentation; another is wrong assignment of delayed strokes.

TABLE II.
PERFORMANCE OF ONLINE HANDWRITTEN UYGHUR WORD SEGMENTING ON DIFFERENT DATASET

Method		Set1 (%)	Set2 (%)	Set3 (%)
over-segmentation	Recall	98.78	98.26	96.52
	Precision	56.13	52.87	49.73
	F-measure	71.54	68.81	65.75
segmentation using DP	Recall	90.94	89.68	86.15
	Precision	71.97	67.72	63.41
	F-measure	80.31	77.36	72.84

VII. CONCLUSIONS

This paper presented a new approach for character separation of online Uyghur cursive handwriting based on dynamic programming which can lay the foundation for the word recognition task. The cursive nature of Uyghur, delayed strokes and characters overlap are some of the key problems that make Uyghur word segmentation more difficult than other languages such as Latin or Chinese. In this paper, the module of over-segmentation cuts the word pattern into a sequence of primitive segments and the dynamic programming method will find the globally optimal segmentation boundaries based on the given candidate segmented block. Our preliminary experiments on Uyghur word data demonstrate that the proposed method can achieve both good performance and efficiency in segmenting cursive handwritten Uyghur characters. The proposal for future work is to refine the filtering of separating point for improving the precision and to apply a recognition system for testing the separating performance.

ACKNOWLEDGMENT

This work is supported by Natural Science Foundation of China (No. 61263038), Program for New Century Excellent Talents in University (NCET-10-0969) and Key Technologies R&D Program of China (2009BAH41B03).

REFERENCES

[1] R. Plamondon, S.N. Srihari, "On-line and off-line handwriting recognition - a comprehensive survey", IEEE Trans. Pattern Anal. Mach. Intell., 22(1), 2000, pp. 63-85 .

[2] S. Jaeger, S. Manke, J. Reichert, etc, "Online hand-writing recognition: The NPen++ recognizer", *Int. J. Document Analysis and Recognition*, 3(3), 2001, pp. 169-180 .

[3] M.S. Khorsheed, "Off-line Arabic character recognition - a review", *Pattern Analysis and Applications*, 5 (1), 2002, pp.31-45.

[4] C.L. Liu, M. Koga, H. Fujisawa, "Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading", *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(11), 2002, pp. 1425-1437.

[5] Q.F. Wang, F. Yin, C.L. Liu, "Handwritten Chinese text recognition by integrating multiple contexts", *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(8), 2012, pp. 1469-181.

[6] A. Amin, "Off-line Arabic character recognition: the state of art", *Pattern Recognition*, 31 (5), 1998, pp.517-530.

[7] A. Benouareth, A. Ennaji, M. Sellami, "Arabic handwritten word recognition using HMMs with explicit state duration", *Journal on Advances in Signal Processing*, 2008, pp.1-13.

[8] Y. Lu and M. Shridhar, "Character segmentation in handwritten words-an overview", Pattern Recognition, 29(1), 1996, pp.77-96.

[9] Y.H. Tseng, H.J. Lee, "Recognition-based Handwritten Chinese Character Segmentation Using Probabilistic Viterbi Algorithm", *Pattern Recognition Letters*, 20(8), 1999, pp.791-806.

[10] C. Hong, L. Gareth, Y.M. Wu, "Segmentation and Recognition of Continuous Handwriting Chinese Text", *Int. J. Pattern Recognition and Artificial Intelligence*, 12(2), 1998, pp.223-232.

[11] L.Y. Tseng, R.C. Chen, "Segmenting Handwritten Chinese Characters based on Heuristic Merging of Stroke Bounding Boxes and Dynamic Programming", Pattern *Recognition Letters*, 19(8), 1998, pp.963-973.

[12] M. Mohamed, P. Gader, "Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation -based dynamic programming techniques", .*IEEE Transactions Pattern Analysis and Machine Intelligence*, 18(5), 1996, pp.548-554.

[13] X. Gao, P.M. Lallican, C.V. Giard-Gaudin, "A two-stage on line handwritten Chinese character segmentation algorithm based on dynamic programming", *Proc of the 8th ICDAR. Washington DC:IEEE Computer Society*, 2005, pp.735-739.

[14] Z. Han, C.P. Liu, "A two-stage handwritten character segmentation approach in mail addresses recognition", *Proc of the 8th ICDAR. Washington DC : IEEE Computer Society*, 2005, pp.111-115.

[15] H. Murase, "Online Recognition of Free-format Japanese Handwritings", *Proc. 9th Int. Conf. Pattern Recognition* , 1988, pp.1143-1147.

[16] T. Fukushima, M. Nakagawa, "On-line Writing-box-free Recognition of Handwritten Japanese Text Considering Character Size Variations", *Proc. 15th Int. Conf. Pattern Recognition*, 2000, pp.359-363.

[17] R.G. Casey, E. Lecolinet, "A survey of methods and strategies in character segmentation", IEEE Trans. Pattern Anal. Mach. Intell., 18(7), 1996, pp.690-706.

[18] A. Cheung, M. Bennamoun, N.W. Bergmann, "A new word segmentation algorithm for Arabic script", DICTA: Digital Imaging Comput. Tech. Appl., 1997, pp.431-435.

[19] Y. Lu, M. Sridhar, "Character segmentation in handwritten words: an overview", *Pattern Recognition*, 29(1), 1996, pp.77-96.

[20] Halmurat, Aziguli: "Research and development of a multifont printed Uyghur character recognition system", *Chinese Journal of Computers*, 27(11), 2004,pp.1480-1484

[21] Mamat Sadik, *Basics of Uyghur Language.*Urumqi, Xinjiang People's Press, 1992 (in Chinese)

[22] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition", *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(1), 1987, pp. 149-153.

[23] C.L. Liu, M. Koga, H. Sako, H. Fujisawa, "Aspect ratio adaptive normalization for handwritten character recognition", in: Tan, T., Shi, Y., Gao, W. Editors, *Advances in Multimodal Interfaces—ICMI 2000*, Lecture Notes in Computer Science, Vol. 1948 , Springer, Berlin, 2000, pp.418–425.

[24] C.L. Liu, X.D. Zhou, "Online Japanese character recognition using trajectory-based normalization and direction feature extraction", *Proc. 10th Int. Workshop on Frontiers in Handwriting Recognition* (*IWFHR*), La Baule, France, 2006, pp.217-222.

Mayire Ibrayim, born in1981, She is currently pursuing a Ph.D. degree at the university of Wuhan,Wuhan, China. Her main research interests include image processing, and handwriting recognition .

Dilmurat Tursun received B.E. in 1983 in Electrical Engineering, from Xinjiang University of China. Currently, he is a professor in the Institute of Information Science and Engineering of Xinjiang University. He has published more than 40 technical papers on experimental phonetics and image processing.

Askar Hamdulla received B.E. in 1996, M.E. in 1999, and Ph.D. in 2003, all in Information Science and Engineering, from University of Electronic Science and Technology of China. In 2010, he was a visiting scholar at Center for Signal and Image Processing, Georgia Institute of Technology, GA, USA, tutored by Professor Biing-Hwang(Fred) Juang. Currently, he is a professor in the School of Software Engineering, Xinjiang University. He has published more than 80 technical papers on speech synthesis, natural language processing and image processing. He is an affiliate member of IEEE.