

On-line Handwritten Uyghur Word Recognition Using Segmentation-Based Techniques

Mayire Ibrayim¹ and Askar Hamdulla²

¹*Postdoctoral Research Station of Computer Science and Technology, Xinjiang University, Urumqi, China 830046,
mayire401@gmail.com*

²*School of Software, Xinjiang University, Urumqi, China 830046,
askarhamdulla@gmail.com*

Abstract

An approach to online handwriting word recognition using segmentation-based techniques is presented in this paper. This approach is referred to as lexicon-driven approach because an optimal segmentation is generated for each string in the lexicon. Word recognition problem is transformed into matching optimization problems between the dictionary entry and the handwritten word image. The segmentation processes use these steps such as removing delayed strokes, shape analysis of the stroke trajectory, reconstructing delayed strokes and combining adjacent fragments. Dynamic matching is used to ranking the lexicon entries in order to get best match. A match score is assigned to a segmentation and string by matching each segment to the corresponding character in the string with a character recognition algorithm that returns confidence value for each character class. As a result the performance for lexicons of size 10, 100, 500 and 1000 are 93.17%, 70.33%, 59.79%, 51.20% and 94.85%, 79.75%, 74.42%, 62.19% for adding distance and normalizing distance respectively.

Keywords: *Uyghur word, Online handwritten recognition, Integrated segmentation and recognition, Path matching, lexicon*

1. Introduction

Handwritten word recognition is a challenging problem encountered in many real-world applications. Recently, devices such as digital tablets, hand-held computers, and mobile technology, provide significant opportunities for alternative interfaces that work in forms smaller than the traditional keyboard and mouse. A variety of approaches for handwritten word recognition have been reported since 1990 [1-7]. But they can be grouped into two major approaches: word-based and segmentation-based. In the word-based approach [8-11], individual letters are not recognized as such but a global feature vector is extracted from the input word. This recognition method is usually faster and avoids problems associated with segmentation. In the segmentation-based approach [12-16], each word is segmented into its component letters and a recognition technique is then used to identify each letter. There are two approaches to word recognition. The one is context-free recognition where a lexicon is used as a post processor to select the best match. The other is Lexicon directed recognition, where a pre-segmented word image is matched against all the words of the lexicon to obtain the best match [17-21]. In this paper, lexicon directed word recognition algorithms based on segmentation of a word image into primitive components will be described.

Uyghur 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 language. Uyghur character written in a cursive style from right to left and

it no upper or lower case exists. Its alphabet contains 32 characters, each has between two and four shapes (128 characters) and the choice of which shape to use depends on the position of the letter 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 is a sequence of disjoint connected components called pseudo-words. It has a main stroke that includes its basic shape, and complementary strokes which include dots or complementary parts. A Uyghur word can have one or more pseudo-word. Each pseudo-word can be a group of characters or one character.

This paper is organized as follow. In next section, describes the overall methodology of the handwritten word recognition system. Section 3 contains the discussion about over segmentation method of words image into primitives. Section 4 describes character confidence assignment. Section 5 describes the recognition and dynamic programming matching. Section 6 is about handwritten Uyghur word data, experimental result and error analysis. Section 7 provides a summary of the work presented in this paper.

2. Word Recognition System Overview

The block diagram of our Uyghur handwriting word recognition system is shown in Figure 1. A handwritten word recognition system uses a lexicon of candidate strings to provide context for a word image in the handwritten word recognition process. Our word recognition algorithms take two inputs: word image and a lexicon representing possible hypotheses for the word image. The goal is to assign a matching score to each lexicon entry or to select the best lexicon entry among the set.

In segmentation stage, the input handwritten word image is first over segmented into primitive segments, then construct candidate segmentation network. A word image is segmented into sub-images called primitives. Each primitive ideally corresponds to a character or part of a character. A segment is either a primitive or a union of primitives.

In matching stage, the sequence of primitive segments is matched with the character string of lexicon. This is a dynamic programming (DP) problem, which minimizes an edit distance, and the result largely depends on the cost defined for segment-to-character match.

3. Segmentation

3.1. Over Segmentation

The segmentation algorithm includes rules that are crucial to the performance of the system. In this process, an input image segmented into a sequence of primitive images. Each primitive is a sub-image of the original image and ideally consists of a single character or a sub-image of a single character. A stroke is defined as all-data point samples written between a certain pen-down action and the following pen-up action. 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.

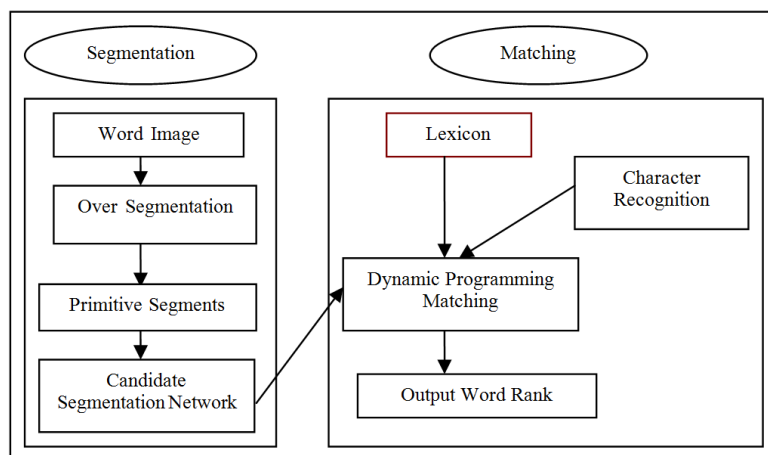


Figure 1. Overview of the Word Recognition System

For character separation we found that it's related with writing style. The connecting type of different writing style of Uyghur characters is as follows. ①Ligature: 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. ③Overlap: 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.

Our over-segmentation process as follows [22-23]: 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 by more than two segmentation points. The over segmentation follows the steps below:

A. Removing Delayed-Strokes: Delayed strokes are detected using a holistic approach. In order to examining the states of successively written Uygur strokes, the geometry features are calculated for each stroke.

B. Pre-Segmentation: Firstly, the initial separation points are formed. Secondly, the overlapping issues are handled. This stage will eliminate some of the elements in the group of initial separation point. Thirdly, the separation sections are generated. Compute the horizontal distance between each two initial separation point. Finally, the segmentation points are located.

C. Rule-based Filtering. The suggested segmentation point is passed through the rule based filtering to discard the incorrect segmentation point.

D. Reconstruct Delayed Stroke. A bounding box of a segmented block is specified by the coordinates of left, right, top and bottom boundaries. In accordance with the written order of each stroke, compute the overlapping degree between the segmented primitive and the delayed stroke.

3.2. Construct Segmentation Network

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 $\{P_0, P_1, \dots, P_N\}$. One or more consecutive segmented blocks may be 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 Figure 2. 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. If the match function is additive, a dynamic programming approach can be used to efficiently find the best segmentation.

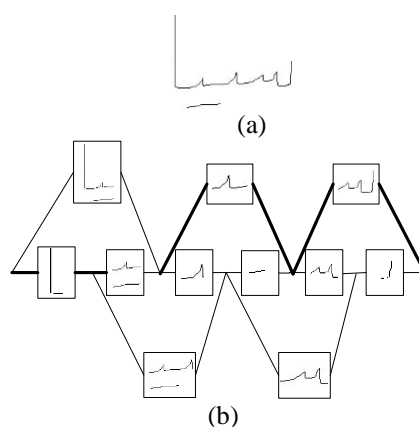


Figure 2. (a) Original Word, (b) An Example of Candidate Segmentation Network

4. Character Confidence Assignment

The core of the matching algorithm is the module that takes a letter from the word image and returns a confidence value. In matching algorithm based on dynamic programming, the matching distance metric between the word images and the dictionary letters will directly determine the matching result. The system finds the best sequence of segments to match a given word from the lexicon under the criterion defined by the match function. To arrive at word level matching scores, the system uses a match function to aggregate the matching scores of segments to characters. In the Uyghur word recognition system which present in this paper, the output of a Uyghur character classifier to represent character confidence, so a high-precision Uighur character classifier can improve the performance of the system.

The character recognizer measures the shape confidence distance between a candidate pattern and a character. In this system, the character recognition is a classifier on character shape features extracted from Uyghur characters [24]. The character classifier (MQDF) [25] is trained on the Uyghur character database. Each character pattern, with the pseudo-two-dimensional moment normalization (P2DMN) method [26] for normalization and a normalization-cooperated method for 8-direction feature extraction (NCFE) [27], is represented by a 200-dimensional feature vector. The feature vector is reduced 120D by fisher linear discriminant analysis (FLDA) for accelerating classification. On the 120D feature vector, the MQDF uses 50 principal eigenvectors for

each of 128 classes (128 Uyghur characters). The output of the MQDF is a quadratic distance (cost).

5. Recognition

In this paper approach, word recognition problem is transformed into matching optimization problems between the dictionary entry and the handwritten word image. The word recognition algorithm has two inputs: a digital image, assumed to be an image of a word, and a list of strings called a lexicon, representing possible identities for the word image. This approach is referred to as lexicon-driven approach because an optimal segmentation is generated for each string in the lexicon.

Word recognition scheme, uses a matching technique designed to handle the large dimensional feature vectors which represent shape description of characters in a word. The recognizer employs a lexicon-driven approach. The lexicon is introduced in the early stage of the recognition process—— an input word image is compared with only those words present in the lexicon. In the matching procedure, comparisons between feature vectors of several possible combination of segments and reference feature vectors of codewords are made to find the best match. For each match, the minimum distance value of each comparison is retained.

5.1. Problem Formulation

Let an input word image denoted as

$$T = \{s_0, s_1, \dots, s_{S_N-1}\} \quad (1)$$

in which S_N is the number of segments in the input image and S_k represents the kth segment. Let the lexicon entries be referred to as R_i .

$$R_i = \{c_i(0), c_i(1), \dots, c_i(C_N(i)-1)\} \quad 1 \leq i \leq L_N \quad (2)$$

where $C_N(i)$ is the number of characters in the ith lexicon entry, L_N is the number of lexicon entry, and $c_i(j)$ is represent the jth character of the ith lexicon entry.

Code words of all characters C_m can be obtained:

$$C_m = \{X(i)\}_{i=1}^m \quad (3)$$

Where $m=1, \dots, 128$ is number of code words of 128 characters, and $X(i)$ represents the ith code word.

Let us define the notion of merged segments as follows:

$$S(b, e) = \{s_b \oplus s_{b+1} \oplus \dots \oplus s_e\} = \{s_v\}_{v=b}^e \quad 0 \leq b \leq S_N, 0 \leq e \leq S_N, b \leq e \quad (4)$$

Where b is a beginning segment and e is an ending segment, and \oplus is a merging operator (Fig.3).

$$\Psi = \left\{ (c_i(0), s_0 \oplus s_r), (c_i(1), s_{r+1} \oplus s_k), \dots, (c_i(C_N(i)-1), s_q \oplus s_{S_N-1}) \right\} \quad (5)$$

Therefore, the basic problem of matching can be described as follows: For a given input word image, T, find a lexicon entry, R_i , which has the minimum distance from T.

$$D^* = \min_{1 \leq i \leq L_N} D(R_i, T) \quad (6)$$

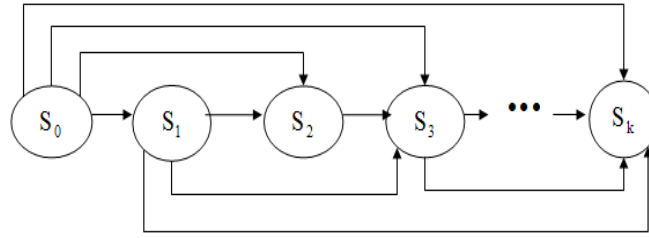


Figure 3. Segmentation Merged

5.2. Dynamic Programming Matching

The core process in handwritten word recognition system is a matching algorithm that finds the best segmentation of a word image matching to each word string in the lexicon under the criterion defined by a match function. Dynamic programming is used to find the segmentation that yields that best match to a given string. In the matching procedure, comparisons between feature vectors of several possible combinations of segments and reference feature vectors of code words are made to find the best match.

In the first phase of the matching, matching between each character of a given lexicon entry $(c_i(j), 0 < j < c_N(i))$ and arbitrary portion of the segment(s), specified by b and e , of the input T is performed.

$$\hat{D}(v, b, e) = d(F(S(b, e)), X(v)) \quad , 0 \leq v \leq c_i(j) \quad (7)$$

Where $F(\cdot)$ represents feature extraction operation for segment(s) and $d(\cdot, \cdot)$ is distance between two feature vectors. This equation(7) gives the distance between each code word of characters $c_i(j)$ and all possible pairs of segments starting at b and ending at e . the best match for all v is obtained by (8).

$$\bar{D}(b, e) = \min_{1 \leq v \leq c_i(j)} \left[\frac{1}{c_i(j)} \hat{D}(v, b, e) \right] \quad (8)$$

In the second phase of the matching, the individual matching scores of the first stage are combined to compute the accumulated cost over the entire lexicon entry. This is accomplished by using dynamic programming. Distance of the best path ending at segment e for each j th character is computed by (9).

$$\bar{D}_j(e) = \min_{1 \leq b \leq e} [\bar{D}(b, e) + \bar{D}_j(b-1)] \quad (9)$$

Based on the recursion of (9), we can formulate the second stage of this dynamic matching procedure for determining the overall best path as follows.

Step 1: Initialization

$$\bar{D}_0(e) = \bar{D}(0, e) \quad 0 \leq e \leq M_d - 1 \quad (10)$$

Step 2: Recursion

$$\bar{D}_1(e) = \min_b [\bar{D}(b, e) + \bar{D}_0(b-1)] \quad e_{\min}^1 \leq e \leq e_{\max}^1$$

$$\bar{D}_2(e) = \min_b [\bar{D}(b, e) + \bar{D}_1(b-1)] \quad e_{\min}^2 \leq e \leq e_{\max}^2$$

...

$$\bar{D}_j(e) = \min_b [\bar{D}(b, e) + \bar{D}_{j-1}(b-1)] \quad e_{\min}^j \leq e \leq e_{\max}^j$$

...

$$\bar{D}_{C_N-1}(e) = \min_b [\bar{D}(b, e) + \bar{D}_{C_N-2}(b-1)] \quad e_{\min}^{C_N-1} \leq e \leq e_{\max}^{C_N-1} \quad (11)$$

Step 3: Final Minimum Distance

$$D^* - \bar{D}_{c_{N-1}}(S_N - 1) \quad (12)$$

6. Experimental Result

In this section, we provide experimental results. We first describe the data sets, and then provide recognition rates and error analysis.

6.1. Acquisition of Online Word Image Data

Acquisition of on-line handwriting word image data is the first step of Uyghur online handwritten word recognition system. The lack of comprehensive data sets is a limiting issue for online handwritten recognition methods. There is no publicly available online Uyghur database. In this paper, the database we used for testing consists of 1460 words collected from different people using a Han Wang writing tablets. A few samples of this data are shown in Fig.4. Handwriting input captured as a stream of position data in the form of "x" and "y" coordinates. The input is a list of points of positions in the stroke written by the user. For a sequence of N data points $X = \{x_i\}$, $i=1, 2, \dots, N$. X is an ordered sequence of (x, y) pen coordinates. The information of the word image is saved as a format of binary file.

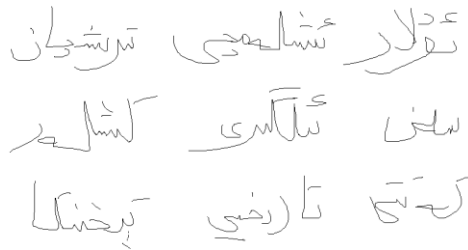


Figure 4. A Few Samples of Handwritten Word

6.2. Experimental Result

By using in this paper approach, word recognition problem is transformed into matching optimization problems between the dictionary entry and the handwritten word image. The inputs to the system are an image of a handwritten word and a lexicon of candidate word strings. The output is a sorted lexicon in which each word is assigned a matching score. The word with the highest matching score is the recognition result. The approach is based on the use of generating multiple possible segmentations of a word image into characters and matching these segmentations to a lexicon of candidate strings. By driving the dictionary information, based on candidate segmentation network, obtaining confidence score of character recognition by the character classifier. The matching of test word images with lexicon entry is a crucial step of handwritten word recognition system.

The match cost function is a more important factor in dynamic programming algorithm, which will influence the selection of least cost path of the search results. In order to merge the primitive components into character and find the optimum character segmentation, the dynamic programming is applied using the summation of character likelihood as the objective function. The match function aggregates the character matching scores to obtain the word matching score.

In this paper, we use two match functions; there are the adding match function and the normalizing match function. And develop a fast approximate matching algorithm using a dynamic programming technique. In Table1, gives online Uyghur handwritten word recognition rate. In the test, we used two kinds of distance measure, namely the adding edit distance and normalized edit distance. Given a test word image, corresponding

dictionaries are randomly generated with size of 10,100,500 and 1000 words. The true word is always present in the lexicon. As a result the performance for lexicons of size 10 , 100, 500 and 1000 are 93.17%, 70.33%, 59.79%, 51.20% and 94.85%, 79.75%, 74.42%, 62.19% for adding minimum edit distance and normalizing edit distance respectively.

Table 1. Top n Correct Word Recognition Rate (%)

Lexicon size		10	100	500	1000
Top n choices					
Adding distance	1	93.17	70.33	59.79	51.20
	2	95.66	76.21	67.28	59.52
	5	98.16	79.32	73.43	62.46
	10		82.48	75.51	63.60
Normalizing distance	1	94.85	79.75	74.72	62.19
	2	96.52	83.34	80.26	68.64
	5	98.43	88.28	83.37	71.83
	10		92.87	86.54	74.42

6.3. Error Analysis

The merits of the path are determined by the path scores for each character. Character score is calculated by NCFE feature value, it reflects the similarity between the combination of primitive segments and the character corresponding in the dictionary. Although get a better online Uyghur handwriting word recognition rate, but there was still a lot of errors exist, it is inevitable.

From the results of current word recognition, four main source of error have been identified:

- 1) High computational complexity of this method requires a longer recognition time.
- 2) Recognized words must exist in the dictionary.
- 3) Segmentation error is the most important reason which caused wrong word recognition.
- 4) With increasing word length, the recognition rate decreases exponentially.

In addition to the above, in the word segmentation process, the detection of multi split point and missed point, incomplete and noise of characters, unreliable of character recognition are the important reasons for the current low recognition rate.

7. Conclusion

In this paper, we describe an algorithm for handwriting Uyghur word recognition algorithm. On-line handwriting input as a natural, convenient has been attached great importance to and has been widely used. However, little work has been done in this area. Through analysis of the unique shape and writing styles of Uyghur characters, this project research an effective approach for online handwritten Uyghur word recognition based on a lexicon-driven, integrated segmentation and recognition have been presented. Word recognition problem is transformed into matching optimization problems between the dictionary entry and the handwritten word image. Dynamic matching between characters in the lexicon entry and segment(s) of the input word image is used to ranking the lexicon entries in order to get best match. We are not very satisfied with our preliminary experimental results of Uyghur word recognition rate. So, the further improvements using better character classifiers and applying the geometric models to character string recognition are needed to be undergoing in the near future.

Acknowledgements

This work is supported by Natural Science Foundation of China (No. 61462081 and No. 61263038) and Natural Science Foundation of Xinjiang (No.2015211C285) and PHD Start Foundation of Xinjiang University (No.BS130117).

References

- [1] J. Simon, "Off-line cursive word recognition", Proceedings of the IEEE, vol. 7, no. 80, (1992).
- [2] F. Kimura, M. Shridhar and Z. Chen, "Improvements of a lexicon directed algorithm for recognition of unconstrained handwritten words", Proceedings of the Second International Conference on Document Analysis and Recognition, (1993) October 20-22; Tokyo, Japan
- [3] M. Chen, A. Kundu and J. Zhou, "Off-line handwritten word recognition using hidden Markov model type stochastic networks", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 5, no. 16, (1994)
- [4] J. Mao, P. Sinha and M. Moidin, "A system for cursive handwritten address recognition", International Conference on Pattern Recognition, (1998) August 16-20; Brisbane, Australia.
- [5] T. Steinherz, E. Rivlin and N. Intrator, "Online cursive script word recognition - a survey", International Journal on Document Analysis and Recognition, vol. 2, no. 2, (1999).
- [6] A.Vinciarelli, A survey on on-line cursive word recognition. Pattern Recognition, 7, 35 (2002)
- [7] K. C. Santosh and C. Nattee, "A Comprehensive Survey on Online Handwriting Recognition Technology and Its Real Application to the Natural Handwriting", Kathmandu University Journal of Science, Engineering and Technology, vol. 1, no. 5, (2009).
- [8] S. Al-Maadeed, C. Higgins and D. Elliman, "Recognition of off-line handwritten Arabic words using hidden Markov model approach", Proceedings of the 16th International Conference on Pattern Recognition, (2002) August 11-15; Quebec, Canada.
- [9] S. Abdelazeem and H. M. Eraqi, "On-line Arabic Handwritten Personal Names Recognition System based on HMM. Proceedings of the 11th International Conference on Document Analysis and Recognition, (2011) September 18-21; Beijing, China.
- [10] A. Benouareth, A. Ennaji and M. Sellami, "Arabic handwritten word recognition using HMMs with explicit state duration", Journal on Advances in Signal Processing, (2008).
- [11] W. Chunheng, Y. Hotta and M. Suwa, "Handwritten Chinese address recognition", In Frontiers in Handwriting Recognition, Proceedings of the 9th International Workshop on, (2004) October 26-29; Tokyo, Japan.
- [12] V. Margner and H. E. Abed, "Arabic handwriting recognition competition", Proceedings of the 11th International Conference on Document Analysis and Recognition, (2011) September 18-21; Beijing, China.
- [13] M. B. Menhaj and M. Adab, "Simultaneous segmentation and recognition of Farsi Latin printed texts with MLP", Proceedings of the International Joint Conference on Neural Networks, (2002) May 12-17; Honolulu, Hawaii.
- [14] C. L. Liu, H. Sako and H. Fujisawa, "Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 26, (2004).
- [15] M. Cheriet, N. Khurma, C. L. Liu and C. Y. Suen, "Character Recognition Systems: A Guide for Students and Practitioners", JOHN WILEY & SONS, INC., (2007).
- [16] X. Q. Ding and H. L. Liu, "Segmentation-driven offline handwritten Chinese and Arabic script recognition", Proceedings of the Summit on Arabic and Chinese Handwriting, (2006) September 27-28; College Park, USA.
- [17] C. L. Liu, M. Koga and H. Fujisawa, "Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 24, (2002).
- [18] F. Kimura, M. Shridhar and Z. Chen, "Improvements of a lexicon directed algorithm for recognition of unconstrained handwritten words", Proceedings of the 2nd International Conference on Document Analysis and Recognition, (1993) October 20-22; Tokyo, Japan.
- [19] M. Shridhar, G. Houles and F. Kimura, "Handwritten word recognition using lexicon free and lexicon directed word recognition algorithms", Proceeding of the 4th International Conference on Document Analysis and Recognition, (1997) August 18-20; Ulm Germany.
- [20] G. Kim and V. Govindaraju, "A lexicon driven approach to handwritten word recognition for real-time applications", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 4, no. 19, (1997).
- [21] P. Gader, M. Mohamed and J. Chiang, "Handwritten word recognition with character and inter-character neural networks", IEEE Trans. Syst Man Cybern B Cybern, vol. 1, no. 27, (1997).
- [22] I. Mayire, H. Zhang and C. L. Liu, "An Effective Character Separation Method for Online Cursive Uyghur Handwriting, Chinese Conference on Pattern Recognition", (2012) September 24-26; Beijing, China.

- [23] I. Mayire, H. Askar and T. Dilmurat, "A Dynamic Programming Method for Segmentation of Online Cursive Uyghur Handwritten Words into Basic Recognizable Units", *Journal of Software*, vol. 8, no. 10, (2013).
- [24] I. Mayire, H. Zhang and C. L. Liu, "A Method for Online Handwritten Uyghur Character Recognition", *Pattern Recognition and Artificial Intelligence*, vol. 6, no. 25, (2012).
- [25] F. Kimura, K. Takashina, S. Tsuruoka and Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 1, no. 9, (1987).
- [26] C. L. Liu, M. Koga, H. Sako and H. Fujisawa, "Aspect ratio adaptive normalization for handwritten character recognition", *Advances in Multimodal Interfaces—ICMI*, (2000) October 14-17; Beijing, China.
- [27] C. L. Liu and X. D. Zhou, "Online Japanese character recognition using trajectory-based normalization and direction feature extraction", *Proceeding of the 10th International Workshop on Frontiers in Handwriting Recognition*, (2006) October 23-26; La Baule, France.

Authors



Mayire Ibrayim, she received B.S. degree from Sichuan University, Chengdu, China, in 2005, and received Ph.D. degree in communication and information systems at the institute of Electronic Information from Wuhan University, Wuhan, China, in 2013. She is a lecturer at Institute of Information Science and Engineering of Xinjiang University since Sept. 2013. Currently, she is working at the Postdoctoral Research Station of Computer Science and Technology, Xinjiang University, Urumqi, China. His research interests include image processing and handwriting recognition.



Askar Hamdulla, he 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 120 technical papers on speech synthesis, natural language processing and image processing. He is a senior member of CCF and an affiliate member of IEEE.