Hierarchical On-line Arabic Handwriting Recognition

Raid Saabni
Department of Computer Science
Ben-Gurion University of the Negev, Israel
Triangle Research&Development Center, Israel
saabni@cs.bgu.c.il

Jihad El-Sana
Department of Computer Science
Ben-Gurion University of the Negev, Israel
Negev Research&Development Center, Israel
el-sana@cs.bgu.c.il

Abstract

In this paper, we present a multi-level recognizer for on-line Arabic handwriting. In Arabic script (handwritten and printed), cursive writing – is not a style – it is an inherent part of the script. In addition, the connection between letters is done with almost no ligatures, which complicates segmenting a word into individual letters. In this work, we have adopted the holistic approach and avoided segmenting words into individual letters. To reduce the search space, we apply a series of filters in a hierarchical manner. The earlier filters perform light processing on a large number of candidates, and the later filters perform heavy processing on a small number of candidates. In the first filter, global features and delayed strokes patterns are used to reduce candidate word-part models. In the second filter, local features are used to guide a dynamic time warping (DTW) classification. The resulting k top ranked candidates are sent for shape-context based classifier, which determines the recognized word-part. In this work, we have modified the classic DTW to enable different costs for the different operations and control their behavior. We have performed several experimental tests and have received encouraging results.

1 Introduction

Keyboards and mice have become the prevalent human-computer interaction devices as they allow people to quickly and efficiently form letters and words, which are the building blocks of literate communication. Nevertheless, keyboards are too cumbersome to utilize in miniaturized computing devices, such as PDAs and mobile phones. For that reason, some of these devices are equipped with small touch screens or pads that enable handwriting interaction. Recognizing handwriting, is still a difficult task because of the huge variance of written word/letter shapes. The challenge is especially daunting when it comes to cursive scripts, such as Arabic.

In Latin and Cyrillic scripts, cursive writing is a style of handwriting that is designed for writing down by hand. In this style, the letters in a word are connected, making a word one single complex stroke. Other scripts, such as Arabic, cursive writing – is not a style – it is an inherent part of the script. The connection between consecutive letters in a word depends on the letter. Some do not connect to the following letter and interrupt the continuity of the stroke. As a result, a word in Arabic script is composed of multiple complex strokes.

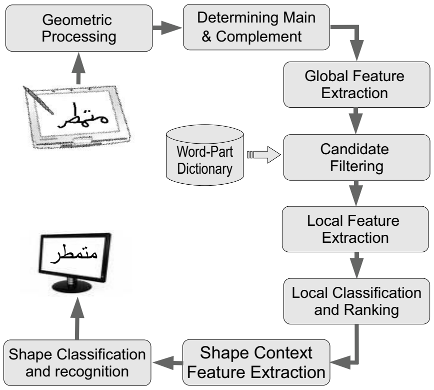
Geometric Processing
Determining Main & Complement
Global Feature Extraction
Word-Part Dictionary
Candidate Filtering
Local Feature Extraction
Local Classification and Ranking
Shape Context Feature Extraction
Shape Classification and recognition
متمطر
متمطر

Figure 1. The flow of our system

The research on cursive script recognition has established two main approaches. The first approach segments an input curve (that represents a word) into individual characters, which are recognized and then assembled to identify the written word. Such an approach is required to maintain only a small set of trained models – one for each letter shape – to handle large vocabulary. However, the absence of consistent baselines, large variations in writing styles, and seamless connection between letters (connection is done with almost no ligatures) make segmentation into individual letters almost impossible. The second approach recognizes the input word shape as a whole and avoids the error-prone segmentation process. Nevertheless, it is required to maintain and train models for each word in the dictionary and

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.263

compare the tested shape against large number of models.

In this paper, we present a new online recognition algorithm for handwritten Arabic script (as shown in Figure 1). We have adopted the holistic approach to avoid segmenting words into letters. Nevertheless, we segment words into connected components, which will be called *word-parts*. We also perform the recognition on the word-part level instead of the whole word level and ignore the additional strokes. Such an approach dramatically reduces the search space as many words share common word-parts and some differ only by the additional strokes. To reduce the search space, we apply a series of filters in a hierarchical manner. The earlier filters perform light processing on a large number of candidates and the later filters perform heavy processing on a small number of candidates. In the first filter, global features and delayed strokes patterns are used to reduce candidate word-part models. In the second filter, local features are used to guide a dynamic time warping (DTW) classification. The resulting k top ranked candidates are sent for shape-context based classifier, which determines the recognized word-part. In this work we have modified the classic DTW to enable different costs for the different operations and control their behavior.

2 Related Work

A lot of research had been done on recognizing isolated forms of Arabic characters as an alternative way of writing. Some of these approaches extract features from the boundaries or the skeleton of the letters to define Fourier descriptors [11, 16, 4] or rely on the Fourier spectrum of the characters [4, 17]. Other approaches extract features from the character strokes, which are fed to a Bayes classifier to recognize the input character [18].

Segmentation-based approaches try to segment the input word into characters or constituent strokes based on the geometric features of letter curves [5, 1, 3]. Other approaches rely on vertical projection and histogram techniques to segment words into characters [6, 10, 14, 2, 9]. Segmentation approaches based on HMM-models [13] or morphological rules [21] were also developed. We refer the interested reader to the survey in character segmentation [19].

Recently the segmentation-free methods became the leading methods for recognizing cursive script. Trinkle [12] uses a hybrid system for printed Arabic script recognition. Global and local features were used by Maddouri and Amiri [15] to recognize words for check verification using transparent neural networks. Saabni [8] proposed a method for on-line Arabic script recognition using HMM.

3 Arab Script Characteristics

The Arabic script is used as the alphabet for several languages such as Farsi, Urdu, Malay, Swahili, Hausa, and Ottoman Turkish. It is written from right to left in a semi-cursive manner in handwriting as well as machine printing. On one hand, Arabic script is similar to western scripts in that it has a strict alphabet consisting of letters, numerals, punctuation marks, spaces, and special marks. On the other hand, it is different in the way it combines letters into words and the way it treats vowels.

The Arabic script consists of 28 basic letters, 12 additional special letters, and 8 diacritics. A letter in Arabic usually has several (2 to 4) different shapes – initial, medial, final, and isolated – according to its adjacent letters and its position within the word. As a result, the 28 basic letters in Arabic script have 120 different shapes. Some letters interrupt the cursiveness of a word by prohibiting a connection to the following letters and splitting words into connected groups of letters called components. Each component includes one or more letters, and with its additional strokes, forms a part of word, which we call *word-part*.

An Arabic word-part, ω, has a *main part*, which is totally cursive, and a *complementary part*, which includes all the additional strokes of the letters within ω (The complementary part could be empty). Several letters share the same body part and differ by the complementary parts. For example, the word-parts *bayt* (بيت) and *tabeth* (تبث) share the same main body and differ in the complementary part.

4 Our Approach

In this section, we discuss the various modules of our online recognizer and its general flow, which is shown in Figure 1. Our system accepts an ordered sequence of samples directly from the digitizer. The input sequence then goes through the following stages in order to recognize the corresponding word.

1. The input sequence goes through several *geometric processing steps* to minimize handwriting variations and reduce noise.
2. The points on the input sequence are classified into body and complementary parts; then the delayed strokes, which belong to the complementary part, are extracted and classified into points and strokes.
3. The global features and delayed strokes patterns are used to determine the set of candidates, which is usually a small fraction of entire dataset [20].
4. Local features are extracted from the point sequence that represents the main body part.
5. The extracted features are fed to a dynamic time warping (DTW) recognizer, which uses the extracted features to determine and order the trained models (candidates) that match the input sequence.
6. The top ranked k candidates are sent to a *shape context* based classifier that determines the recognized word.

In the following subsections we discuss in detail each of these stages.

4.1 Geometric Preprocessing

Most digitizers perform uniform temporal sampling, which often results in an oversampling of slow pen motion regions and under-sampling of fast pen motion regions. This stage performs writing-speed normalization by re-sampling the point sequences and distributing the points uniformly over the sampled curve. The point sequence (polyline) is then smoothed, using a low pass filter – to minimize handwriting variations, reduce noise, and remove imperfections caused by acquisition devices.

The number of edges/vertices representing a polyline usually influences the number of features used to characterize it. The running time of most statistical recognizers is affected by the number of features. For that reason, it is desirable to reduce the number of points in the sequence (polyline) while maintaining the shape of the input model.

In this work, we have adopted the Dynamic Time Warping (DTW) statistical recognizer, which tends to produce better results when the edges of the polyline are of similar length. Therefore, our simplification algorithm reduces the number of vertices that represent the polyline, while maintaining almost the same length for its edges. We chose to simplify a polyline $p = v_0, v_1, \cdots, v_{n-1}$ by applying the vertex-removal operator. This operator removes a vertex v_i based on its distance from the segment $\overline{v_{i-1}, v_{i+1}}$ and the distance to its two adjacent vertices.

4.2 Features Extraction

The detection of delayed strokes is performed based on their sequential order, location, and size. Delayed strokes are detected based on the size and shape of their bounding box with respect to the word-part.

We extract two types of features from the body part – global and local features. The global features include loops, ascenders, and descenders. The local features characterize local relation between adjacent or nearby points on the polyline.

The global features are easy to extract in on-line handwriting recognition systems. Loops are detected by inspecting the self-intersection within the curve. Ascenders and descenders are defined with respect to lower and upper baselines. The existence of these baselines and respecting a constrained writing style simplify the extraction of reliable ascender and descender features; otherwise, it is hard to rely on these features. In online handwriting, it is easy to define and draw upper and lower baselines and respect the first-grade Arabic writing rules.

The local features are extracted from the point sequence and quantify the relation between neighboring strokes. Let $P_s = \{p_0, \ldots, p_{n-1}\}$ denote the input sequence after applying geometric processing. From this sequence we extract the following two features.

- For each point p_i, $i > 0$, we determine the angle between the segment $\overline{p_{i-1}p_i}$ and the x-axis (the horizontal line). We will refer to this feature as $\alpha(p_i)$. This feature quantifies the relation between adjacent segments, but does not provide any information concerning the point's environment.
- To quantify the relation between a point and its environment, we extract a semi-global feature, similar to the one introduced by Belongie and Malik [7]. It is defined as the angle between the segment $\overline{p_{i-1}p_{i+\delta}}$ and the x-axis, where δ determines the width of the considered environment. We will refer to this feature as $\beta(p_i, \delta)$, where $\delta > 2$.

The two features are interpolated linearly using Equation 1, where w is a normalized positive weight that controls the blending of the two features and δ.

$$f(p_i) = (1 - w)\alpha(p_i) + w\beta(p_i, \delta) \qquad (1)$$

4.3 Shape Context

The second recognition phase utilizes the *shape context* feature, introduced by Belongie and Malik [7]. The shape context feature vectors scheme considers the set of n points on the contour, C, of the shape. For each point $p_i \in C$ it assigns $n - 1$ vectors, one for each point $p_j \in (C - p_i)$. This set is very rich description vectors, however, it is too detailed. Therefore, the relative position distribution is used as a robust, compact, and highly discriminative descriptor. For each point p_i, the scheme defines the shape context to be the coarse histogram of the relative coordinates of the remaining $n - 1$ points.

We use the shape context feature on the stroke of the body part as it was used on the closed contours. We use the n points taken uniformly from the given stroke.

5 Word-Part Recognition

Matching algorithms are the core process of any recognition system. The recognition and classification algorithms rely on matching techniques to determine the similarity between two point sequences. The feature-based techniques extract and compare a set of feature vectors from the two strokes (polyline). In this paper we use a feature-based technique as it provides flexible comparison, which is essential to handling varying handwriting styles.

We had avoided segmenting word-parts into letters and considers the continuous word-parts as the basic alphabet of the Arabic language. As a result, the recognition for a written word is performed by recognizing its word-parts in the right order and combining them while consulting the dictionary [20]. For that reason, the basic matching procedure compares word-parts, i.e., computes the match between two word-parts.

5.1 Dynamic Time Warping

Dynamic Time Warping (DTW) is an algorithm for measuring similarity between two polylines which may vary in time or speed. This technique suits matching sequences with nonlinear warping. For one-dimensional sequences, DTW runs at polynomial time complexity and is usually computed by dynamic programming using Equation 2.

$$D(i,j) = \min\{D(i,j-1), D(i-1,j), D(i,j)\} + cost \tag{2}$$

In this research, we have slightly adjusted the classic DTW to include different costs for insertion, deletion, and substitution. In addition, we have adopted an extra-cost for consecutive insertion and deletion to avoid introducing long segments that disturb the recognition accuracy. The DTW is computed by taking the minimum of the three options including the cost of each operation, as shown in Equation 6. We assign different cost functions for deletion, insertion, and substitution based on the introduced change. In all handwriting, including Arabic, the difference between two point sequences that represent two different words is very small, i.e., inserting/deleting just a few consecutive elements can change the sequence to represent a different word-part.

The match between the shapes of two word-parts is estimated by computing the feature vectors, mentioned in section 4.2. Let S_a and S_b be the sequences of the feature vectors calculated from the two word-parts. We define the $cost_{ins}(i)$, $cost_{del}(i)$, and $cost_{sub}(i,j)$ as the cost of inserting a new element at i into the sequence S_a, the deletion of the element i from the sequence S_a, and the substituting of the element i in the sequence S_a by the element j in sequence S_b, respectively. Equation 3, 4, and 5 define the cost of each operation, Where del_i and ins_i are the numbers of consequent operation of deletion or insertion until point i, respectively.

$$\begin{align} cost_{sub} &= (S_a(i) - S_b(j))^2 \tag{3} \\ cost_{del} &= ((S_a(i+1) - S_a(i)) * ins_i)^2 \tag{4} \\ cost_{ins} &= ((S_b(i+1) - S_b(i)) * del_i)^2 \tag{5} \end{align}$$

In order to embed the influence of consequent deletion or insertion into the minimization problem of the DTW, we use Equation 6 to define the dynamic programming.

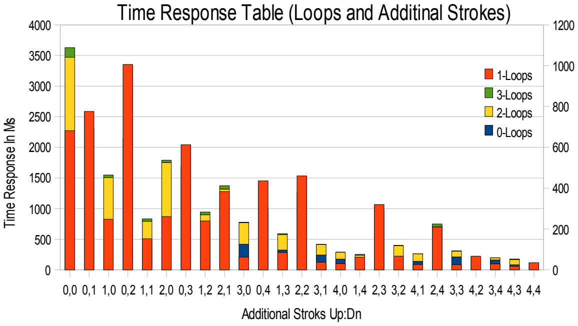

Figure 2. The response time of our system

$$\begin{align} D(i,j) = \min\{&D(i,j-1) + cost_{ins}, \\ &D(i-1,j) + cost_{del}, \\ &D(i-1,j-1) + cost_{sub}\} \tag{6} \end{align}$$

As can be seen, this rule costs consequent operations of deletion and insertions in a quadratic factor to the number of these consequent operations. This scheme forces the spread of these operations over all the fitting process and thus, forbids consequent operations of deletion or insertion.

Several stages are performed to reach the final recognition of a written word-part. In the first stage, the system filters a class of all candidate word-parts from the dictionary using the global features and the complementary part as explained in section 4.2. In the second stage, a DTW algorithm is applies to measure and score the similarity between the input word-part and each candidate word-part using the extracted local features. In the third stage, the k top ranked word-parts are selected and compared against the written word-part using shape context features. The closest word-part is reported as the recognized word-part.

6 Experimental Results

In this project, we focus on testing the feasibility of the online recognition of Arabic script using the holistic approach in a reasonable response time. We have implemented our system and performed several tests on various datasets using $2.1GHz$ Pentium Dual-Core with $1024GB$.

The average response of our unoptimized system for recognizing a written word-part on the open vocabulary system was $954\ ms$ and the longest was close to $2800\ ms$. We consider this time response as reasonable and focused on the recognition precision. The graph in Figure 2 shows the response time with respect to various configuration.

To evaluate our system we generated the shapes of the words in the database by using a group of 10 writers. Each writer wrote a compact set of Arabic words that include all the Arabic letters in their different shapes. A semi-automatic system was used to generate, for each writer, the

User Type	GM.Hit	GM.5	SCM.Hit
Tester1(Trainer)	88%	98%	90%
Tester2(Trainer)	83%	96%	89%
Tester3(Trainer)	85%	95%	87%
Tester4	85%	94%	87%
Tester5	83%	92%	86%
Tester6	86%	94%	88%

Table 1. The recognition behavior of the various stages of our system for each tester

shapes of all the words in the database from the written compact set.

For evaluating the recognition rate, we asked each user to write 100 word-parts retrieved randomly from the database. Six students participated in our experiment, where each performed the test 10 times, with different sets of random word-parts. Three of the six students participated in generating the shapes of the word-parts (trained the system). Such separation enables evaluating the writer dependency of the system. Table 1 summaries these results. The column *GM.Hit* reports the recognition rate after the geometric filter; the column *GM.5* reports the rate of finding the correct word in the top 5 candidates; and the column *SCM.Hit* reports of the success rate of the shape-context filter using top 5 candidates.

7 Conclusion and Future Work

We have presented a multi-level recognizer for online Arabic handwriting. The multi-level recognition is performed through a series of filters that aim to reduce the search space. At each phase, the number of candidates is reduced. The core of the system is based on modified dynamic time warping, which is followed by a shape context classifier applied on the resulting top k candidates. We have performed several tests on various datasets and received encouraging results.

References

[1] G. M. A. Amin and J. Haton. Recognition of handwritten arabic words and sentences. *Proc. of 7th Int. Conference on Pattern Recognition, Canada*, pages 1055–1057, 1984.

[2] H. Al-Yousefi and S. Udpa. Recognition of arabic characters. *IEEE Trans. Pattern Analysis Machine Intell*, 14(8):853–857., 1992.

[3] H. Almuallim and S. Yamaguchi. A method of recognition of arabic cursive handwriting. *IEEE Trans. Pattern Analysis Machine Intell*, pages 715–722., 1987.

[4] S. Alshebeili, A. Nabawi, and S. Mahmoud. Arabic character-recognition using 1-d slices of the character spectrum. *SP*, 56(1):59–75, January 1997.

[5] A. Amin, A. Kaced, J. Haton, and R. Mohr. Handwritten arabic character recognition by the irac system. In *5th Int. Conf. Pattern Recognition, Miami, FL.*, pages 729–73, 1980.

[6] A. Amin and J. Mari. Machine recognition and correction of printed arabic text. *IEEE Trans. Syst. Man Cybern*, 19(5):1300–1306., 1989.

[7] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

[8] F. Biadsy, R. Saabni, and J. El-Sana. Segmentation-free online arabic handwriting recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, page to appear, 2009.

[9] B. Bushofa and M. Spann. Segmentation and recognition of arabic characters by structural classification. *IVC*, 15(3):167–179, March 1997.

[10] S. El-Emami and M. Usher. On-line recognition of handwritten arabic characters. *IEEE Trans. Pattern Analysis Machine Intell*, 12(7):704–710., 1990.

[11] T. El-sheikh and R. Guindi. Automatic recognition of isolated arabic characters. *Signal Processing*, 14(2):177– 184., 1988.

[12] A. Gillies, E. Erl, J. Trenkle, and S. Schlosser. Arabic text recognition system. In *Proceedings of the Symposium on Document Image Understanding Technology*, 1999.

[13] A. Gouda and M. Rashwan. Segmentation of connected arabic characters using hidden markov models. In *Computational Intelligence for Measurement Systems and Applications,*, volume 14-16, pages 115 – 119, July 2004.

[14] . E.-D. K. El-Gowely and A. Nazif. Multi-phase recognition of multi-font photoscript arabic text. In *Proc. 10th Conf. on Pattern Recognition*, pages 700–702., 1990.

[15] S. Maddouri and H. Amiri. Combination of local and global vision modelling for arabic handwritten words recognition. In *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pages 128–135, 2002.

[16] S. Mahmoud. Arabic character recognition using fourier descriptors and character contour encoding. *Pattern Recognition*, 27(6):815–824., 1994.

[17] N. Mezghani, A. Mitiche, and M. Cheriet. On-line recognition of handwritten arabic characters using a kohonen neural network. In *Proceedings of the International Workshop on Frontiers of Handwriting and Recognition*, 2002.

[18] N. Mezghani, A. Mitiche, and M. Cheriet. Bayes classification of online arabic characters by gibbs modeling of class conditional densities. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1121–1131, 2008.

[19] E. L. R.G. Casey. Strategies in character segmentation: a survey. In *Third International Conference on Document Analysis and Recognition*, volume 2, page 1028, 1995.

[20] R. Saabni and J. El-Sana. Justifying holistic approach for arabic script recognition. Technical report, Ben Gurion University of the negev, Israel, 2008.

[21] S. T. Souici and L. M. Sellami. Off-line handwritten arabic character segmentation algorithm: Acsa. In *Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 452–457, 2002.