A New Segmentation Algorithm for Online Handwritten Word Recognition in Persian Script

Sara Izadi, Mehdi Haji, Ching Y. Suen
Centre for Pattern Recognition and Machine Intelligence, 1455 de Maisonneuve Blvd. West, Montral, Qubec, Canada, H3G 1M8
{s_izadin, m_haji, suen}@cs.concordia.ca

Abstract

The cursive nature of Persian alphabet, and the complex and convoluted rules regarding this script cause major challenges to segmentation as well as recognition of Persian words. We propose a new segmentation algorithm for the main stroke of online Persian handwritten words. Using this segmentation, we present a perturbation method which is used to generate artificial samples from handwritten words. Our recognition system is composed of three modules. The first module deals with the preprocessing of the data. We propose a wavelet-based smoothing technique which enhances the recognition performance compared to the conventional widely used technique. The second module is word segmentation into convex portions of the global shape which we call Convex Curve Sectors (CCSs). The third module is to analyze those CCSs and use the information for recognition performed by Dynamic Time Warping (DTW) technique. Using CCSs provides the DTW-based classifier with a compact word representation which makes comparison much faster.

Keywords: Online, Handwriting Recognition, Segmentation, Cursive, Dynamic Time Warping (DTW)

1 Introduction

Pen-based handwriting input provides a convenient means of interaction with devices such as Personal Digital Assistants (PDAs), smart phones, and hand-held computers. The international growth in the usage of these devices encourages research in online handwriting recognition for different scripts. The techniques to understand online shapes, sketches and handwritings however, are far behind the advances in hardware. Developing online handwriting recognition can provide the modern technological devices with ease of using pen and paper. Most pen-based interactions are still done using either traditional mouse motions or with artificial gesture languages like Palms Graffiti [17].

There is much less research for recognizing Persian text than other scripts like Latin and Chinese. However, handwritten recognition in this script has recently started to receive a lot of attention from a number of researchers [8],[22]. Insufficiency of data is the first obstacle faced in developing a recognizer for this script. In addition to lack of database, Persian script has very specific characteristics which make the recognition task even more difficult compared to Latin. Persian alphabet consists of thirty two basic letters, several of which share the same basic form and differ only by a small complementary part. The complementary part could be a dot, a group of dots or a slanted bar. A Persian letter can appear in up to four different forms: *isolated*, *initial*, *medial*, and *final*. The effective size of the alphabet increases due to this characteristic. Also, there are many different styles of writing in Persian which introduce different allographs for letters or some letter combinations.

The difficulty of automatic handwriting recognition, regardless of the script, lies in the huge variety of writing styles among different individuals and also in different samples written by the same person. Persian script is cursive in both printed and written forms, meaning that most letters join the following letter along a baseline in order to make a word. The cursiveness of the script makes segmentation task extra difficult.

Most cursive recognizers [5] use a combination of Hidden Markov Models (HMM) and a sliding window to extract features, in order to avoid the segmentation problem. Another approach to word recognition is to recognize a word by its individual letters. In this type of approach it is difficult to find the correct segmentation and the character recognition processing is not error free. Our idea is to segment the word into elementary convex curve sectors which we call *CCSs*.

In our approach, a word is first represented by features extracted from CCSs, and then classified by Dynamic Time Warping (DTW) method. DTW has been used for several handwriting recognition tasks such as writer identification [7] and adaptive online character recognizer [19]. Aligning two cursive strokes with complex shapes by DTW can be costly and slow. However, our proposed segmentation algorithm significantly reduces the computational cost, and makes the comparison very fast. In order to provide our system with enough data to experiment, we introduced a perturbation procedure to generate some synthetic samples. The introduced recognition system is then tested on the mixture of the real and synthetic data and shows promising results for cursive handwriting recognition.

The paper is organized as follows. In Section 2 we provide details on preprocessing operations on online data. The segmentation algorithm is presented in Section 3 followed by the procedure we used to generate synthetic data from the samples written by humans. We explain the features we extracted from each CCS in Section 4. In Section 5 we present the DTW technique employed by our recognition system. The dataset we used in our experiments and the experimental results are provided in Section 7. Finally,in Section 8 we conclude this paper with a discussion of different applications and suggestions for further improvements.

2 Preprocessing

The purpose of preprocessing prior to segmentation is to remove irrelevant information and recover from artifacts in order to ultimately improve recognition accuracy. Despite the low-pass hardware filters embedded in most of the digital tablets, recorded online trajectories often appear in rough and uneven forms. Typically, preprocessing for online data includes elimination of high frequency noise, hooks, and duplicate points as well as point re-sampling [19]. Smoothing aims for reducing the high frequency from the digitizer or erratic pen motions. Smoothing is usually done by applying a moving average of a fixed size window. Geometric variations in handwritings are usually due to varieties in writing styles. Normalization is therefore important, especially for a writer-independent system, as it tries to eliminate geometric variations in handwriting data. The preprocessing we applied in this research includes smoothing, de-hooking, and size normalization.

Wavelet Transform provides a way of analyzing local behavior of functions. We used a smoothing method based on the orthogonal decomposition of the online data into the *Haar* basis [9],[6] instead of the conventional moving average. We perform a single-level one-dimensional wavelet decomposition for each of the x and y streams and take the approximation coefficients vectors as the smoothed trajectory. A higher order decomposition was not applied since it tended to remove useful details about the shape of the word in our experimental studies. As we discuss in Section 7, the recognition rate is improved by using wavelet filtering instead of the simple averaging.

3 Convex Curve Sector Segmentation

Segmenting a complex curve can be done by piecewise approximation using a number of functions. Detecting zero-crossing of curvature (inflection points) is another approach for curve segmentation. Curvature maxima found by applying the mathematical definition of the curvature, may result in cut-points which are not perceptually relevant [18].

There have been few attempts for segmenting online cursive handwriting in the literature. In [4], segmentation for English script is done by using two modules: a cutter that generates candidate cuts using the pen velocity, and a combiner that gets those candidates and delivers candidate segments. In each module, several heuristic filters have been employed for removing incorrect outputs.

The rapid human movements that generate handwriting can be seen as a composition of elementary movements which correspond to elementary shapes [12]. The main idea of the segmentation algorithm 1 we devise is to decompose a digital curve into convex/concave segments which we consider as elementary shapes. The algorithm is based on a simple concept yet provides robust functionalities. We use the notions of concavity and convexity in Euclidian space: a curve in 2D is concave if every line segment joining two points on its graph does not lie above the graph. Symmetrically, a curve is convex if every line segment joining two points on its graph does not lie below the graph at any point. In order to avoid finding segments of very short lengths, we apply a threshold on the segment curve length. By curve length we mean the sum of the lengths of the piecewise linear segments which construct the curve.

If the trajectory is given by a point sequence $A = \{P_0, P_1, ..., P_N\}$, the algorithm takes two points at a time as the start and the end of the segment. While convexity criteria and the length constraint are satisfied, the algorithm proceeds the end point along the curve keeping the first point unchanged. Once convexity criteria is violated, a segmentation point is found. The same procedure is then applied to the rest of trajectory points. It should be noted that the considered definition of the convexity does not guarantee strictly convex/concave functions. Therefore a CCS can include a combination of a strictly convex curve and a straight line segment attached to it. However, this does not affect our segmentation algorithm as straight line is includes in the set of selected primitives.

Figure 1 shows some examples of segmentation of two-letter sub-words. As can be seen in this Figure, our segmentation algorithm finds the points showing abrupt changes.

Algorithm 1 CCS Curve partitioning

INPUT: A set of successive preprocessed trajectory points and a threshold L:
$A = \{P_1, P_2, ..., P_N\}$
OUTPUT: A set of segmentation points $S \subset A$
$Initialization : P_s = P_1; P_e = P_3; S = \emptyset$
while $e < N$ **do**
  for all $i = s$ to e **do**
    Compute d_i= signed distance$(P_i, \overrightarrow{P_s P_e})$
  end for
  if All d_i's $i = s, .., e$ are not of the same sign **then**
    if Curve-Length $(P_s P_e) > L$ **then**
      $S = S \bigcup p_{i-1}$
      s=i; e=s+3;
    end if
  end if
end while
Return S

3.1 Generating Artificial Data

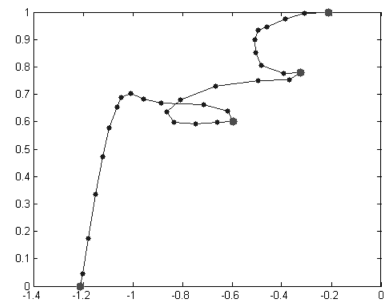
Collecting handwriting databases is a time consuming task, especially for online samples. This has motivated researchers to design methods for generating offline and online new samples based on the collected ones [3]. However, due to differences in the nature of offline and online data, not every method applied to offline images can be applied to online data stream.

In this paper we present a method for generating synthetic words from existing ones written by human writers. This method is based on our segmentation algorithm. The idea is to add degradation to real samples by randomly selecting some segments of the main stroke. The curve width of each selected segment to be distorted is changed by an arbitrary amount of perturbation α. The size of α is small potion of the length, and the sign of α is also determined randomly. Figure 2 shows a real sample (a) and artificial samples (b,c) generated based on that. It is noticed that the generated synthetic samples are very similar to natural variation of the original sample.

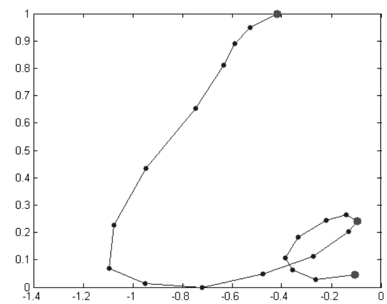
4 Feature Extraction

Feature extraction for cursive word recognition can be done in two major ways: extracting features by using a sliding window which is a segmentation-free approach, or by segmenting the curve into meaningful pieces and extracting features from each segment.

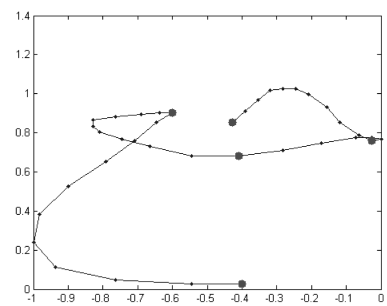
Some parts of the trajectory may carry more information than the others, and we design our features to reflect this fact. Here we describe each curve segment by five features as follows. Let $S_i = \{P_1(x_1, y_1), ..., P_M(x_M, y_M)\}$ be a CCS representation. The width of a closed convex/ curve is defined as the distance between parallel lines bounding it. We calculate an approximation of the distance between the lower and the upper supporting lines as a feature. Let d_i be the distance of the point P_i from the line segment $P_1 P_M$. Now, we express the first feature by:



(a)



(b)



(c)

Figure 1. Persian 2-letter sub-word segmentation.

$$F = \max(d_i), \quad (1)$$

A diameter of a convex curve is a line joining two points of the curve at which the support lines are parallel and distinct. Two other features we used to describe a CCS are the slope and the length (i.e. a measure of the flatness of the curve) of the convex diameter $(\overrightarrow{P_1 P_M})$. In order to count for the relative position of the segment in the whole curve, we included two more features which are the normalized vertical coordinates of points P_1 and P_M.

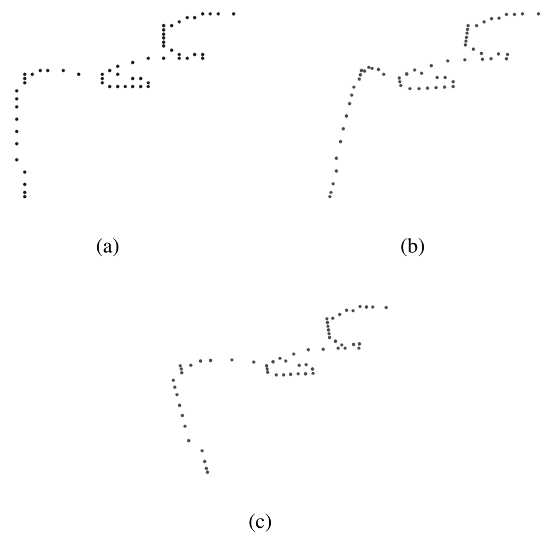


Figure 2. Persian 2-letter sub-words: (a)original, (b)and (c)synthetic.

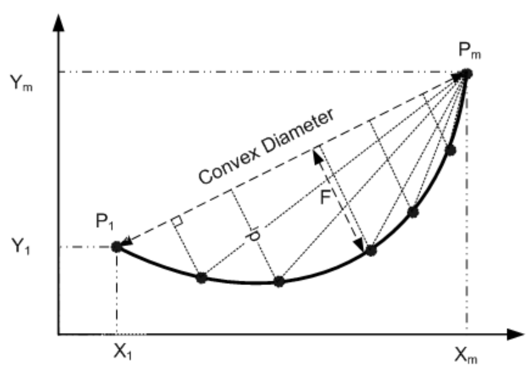


Figure 3. Feature extraction for a curve segment.

5 Recognition

Dynamic time warping (DTW) is a method for measuring overall similarity between two time varying sequences of different sizes based on the measure of local minimum distances. DTW has been frequently used after it was introduced by Sakoe and Chiba [16]. DTW is a template matching mechanism that aligns test pattern T, and reference patterns, $R = \{R_1, ..., R_n\}$, and computes the best possible alignment warp, Θ_v , between T and R_v , and the associated distortion $D(T, R_v)$.

The strength of this technique lies in using mapping for computing the overall distance. Comparing a reference pattern and a test pattern at equal times may yield a large distance despite of the overall similarity. Therefore, to achieve a better comparison, DTW tries to make similar events happen at the same time for the two sequence by aligning them. After aligning the two streams by non-linearly warping them in the time domain via a point-to-point mapping, the overall distance is computed by adding the local distances between each pair of aligned points. Among different possible alignment warps, one that minimizes the distance is the optimal alignment, and the associated distance is the overall distance. In DTW, a set of local continuity constraints are usually imposed on the warping function to ensure proper time alignment while keeping any potential loss of information to a minimum [13]. These constraints include continuity, and monotonicity of the mapping in time dimensions, and bound the search space for the optimal mapping.

In order to achieve a satisfactory accuracy for a user-independent recognition system without constraining the allowed writing styles, the recognizer has to be adaptive. A DTW-based recognition system facilitate adapting to the new styles by adding new prototypes. In DTW word classification, there is a set of template streams associated with the words in the vocabulary. Given a test stream, first the minimum distance between each template and the instance is computed. The template with the best score (closest distance) determines the class of the instance. DTW has been applied for automatic extraction of handwriting styles and verification of a large human-labeled data set of online handwriting [11] and [21] respectively. We use DTW for online Persian word recognition through matching because of its flexibility for pattern alignment compensates for the variation of handwritten words.

DTW has been proved to be an efficient method to calculate distance in online handwriting recognition [10], [1]. The major weaknesses of DTW is its time complexity of $O(N^2V)$, where N is the length of the sequence and V is the number of templates to be considered. Our solution to this problem is avoiding DTW point-to-point matching between samples on a trajectory in our recognition system. By segmenting the trajectory we reduced the length N of the series significantly. This consequently speeds up the recognition task as we empirically illustrate in our results.

6 Data

Design of online handwritten word recognizers for Persian script is an open and active area of research. Currently, the lack of comprehensive data sets is a limiting issue for online handwritten recognition methods. There is no publicly available online Persian database. The Data we used for this research is Persian sub-words [15][1]. In the database we used, there are 12 samples of each sub-word on average. As the number of samples are too few

[1]This database is the courtesy of Tarbiat Modarres University.

in this data set, we created some more artificial data from the existing samples by the method we described in Section 3.1. The total number of samples increased up to 40 for each class, after applying our degradation procedure.

Persian is a cursive script, and its alphabet consists of 32 letters. In order to make a word, all letters except for seven of them, connect directly to the letter which immediately follows along a writing line or baseline. If one of those seven letters happen to be in the middle of the word, they create *sub-words*. Therefore, each part of the word, we refer to as sub-word, has a main stroke that includes its basic shape, and complementary stokes which include dots or complementary parts. In this research we focus on the recognition of shape of the main strokes .

7 Experiments

To examine the utility of our approach, we have carried out recognition experiments on the sub-word data set explained in the previous section. We considered 20 classes of each 2-letter and 3-letter sub-words in our experiments. It is known that DTW is not a fast algorithm. The speed of a DTW-based classifier can be improved by some computational techniques to minimize the number of possible matches as well as using faster approximations of DTW that has lower complexity. However, our segmentation strategy already reduces the length of the series to be aligned by DTW, significantly. Therefore, it quadratically reduces the time required for recognition of samples. As a result, the average measured time in our experiment for classification of a test sample is about $3ms$. While it is $600s$ for the case we use normalized coordinates as feature vectors. These results presented in Table 1 and 2 which are sufficient for an online application. Table shows the recognition improvement gained with the wavelet-based smoothing technique we introduced in Section 2.

Table 1. K-NN classification results for different values of k using 20 templates for each class.

	Accuracy(%)		
	k=1	k=3	k=5
2-letter word	89.4	70.1	63.0
3-letter word	85.0	73.4	67.7

It should be noted that we do could not compare our results with any previous work for Persian online words, since no benchmark currently exists. To the best of authors knowledge, also there has not been any reports in the literature on the recognition rate for the word section of the database we used in this paper. However, to have an intuition about the recognition rate of DTW technique on other scripts one can see the results in [14] and [20].

Table 2. Recognition performance for 2 and 3-letter sub-words using 1-NN.

	# templates	Accuracy(%)
2-letter word	10	76.9
	15	83.0
	20	89.4
3-letter word	10	74.1
	15	76.1
	20	85.0

Table 3. Comparison of recognition performance using wavelet-based versus conventional smoothing with 1-NN.

	# templates	Accuracy(%)	
		wavelet	moving avg
2-letter word	10	76.9	71.4
	15	83.0	82.8
	20	89.4	88.2

8 Conclusion

In this paper, we have proposed a new segmentation approach for cursive handwriting to enhance the performance of DTW-classifier. We also introduced a degradation scheme to generate synthetic samples from the real handwriting ones. DTW has a quadratic time and space complexity that limits its usage to only small time series and small dictionary sizes. However, using our new segmentation algorithm, we achieved a significant (about 200) speed-up which makes it suited for real-time applications. The performance of our DTW-based recognition system shows promising results for Persian sub-words. We expect even a higher recognition rate by taking the extra information of the complementary stokes of the subwords which we did not take into account in this paper.

In our future work, we focus on improving the accuracy using data clustering, ensemble of classifiers, and pruning techniques. We are currently working to design a clustering method for template selection directly from time series based on the DTW distance. Combination of likelihood measures of some stochastic approaches, and DTW scores should also improve the results. In the case of support vector machines and DTW, such combinations has been already proved useful in [2].

References

[1] C. Bahlmann and H. Burkhardt, ”The Writer Independent Online Handwriting Recognition System frog on hand and Cluster Generative Statistical Dynamic Time Warping”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):299–310, 2004.

[2] C. Bahlmann, B. Haasdonk and H. Burkhardt, ”On-Line Handwriting Recognition with Support Vector Machines

- A Kernel Approach", *IWFHR '02: Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, 2002, pp 49, Washington, DC, USA. IEEE Computer Society.

[3] H. Baird, "The State of the Art of Document Image Degradation Modeling", *4th IAPR International Workshop on Document Analysis Systems*, 2000, pp 1–16, Rio de Janeiro, Brazil.

[4] Y. Bengio, Y. LeCun, C. Nohl and C. Burges, "LeRec: a NN/HMM hybrid for on-line handwriting recognition", *Neural Comput.*, 7(6):1289–1303, 1995.

[5] F. Biadsy, J. El-Sana and N. Habash, "Online Arabic Handwriting Recognition Using Hidden Markov Models", 2006. the 10th International Workshop on Frontiers of Handwriting Recognition (IWFHR).

[6] I. Daubechies, *Ten lectures on wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[7] M. Faundez-Zanuy, "On-line signature recognition based on VQ-DTW", *Pattern Recognition*, 40(3):981–992, 2007.

[8] R. Halavati, S. Shouraki and S. Hassanpour, "Evolution of Multiple States Machines for Recognition of Online Cursive Handwriting", *World Automation Congress, 2006. WAC '06*, July 2006, pp 1–6, Budapest.

[9] S. Jaffard, Y. Meyer and R. D. Ryan, *Wavelets*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, revised edition, 2001Tools for science and technology.

[10] H. Mitoma, S. Uchida and H. Sakoe, "Online Character Recognition Using Eigen-Deformations", *IWFHR '04: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR'04)*, 2004, pp 3–8, Washington, DC, USA. IEEE Computer Society.

[11] R. Niels and L. Vuurpijl, "Generating Copybooks from Consistent Handwriting Styles", *ICDAR*, 2007, pp 1009–1013.

[12] R. Plamondon and F. Maarse, "An evaluation of motor models of handwriting", *IEEE Transactions on Systems, Man and Cybernetics*, 19, 1989.

[13] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[14] B. S. Raghavendra, C. K. Narayanan, G. Sita, A. G. Ramakrishnan and M. Sriganesh, "Prototype Learning Methods for Online Handwriting Recognition", *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, 2005, pp 287–291, Washington, DC, USA. IEEE Computer Society.

[15] S. M. Razavi and E. Kabir, "A Data base for Online Persian Handwritten recognition", *6th Conference on Intelligent Systems*, 2004, pp 859–863.

[16] S. Sakoe, H. Chiba, "Dynamic programming algorithm optimization for spoken word recognition", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, Feb 1978.

[17] T. F. Stahovich, R. Davis, R. C. Miller, J. A. Landay and E. Saund, "Pen-based computing", *Computers & Graphics*, 29(4):477–479, 2005.

[18] C. D. Stefano, M. Garruto and A. Marcelli, "A Multiresolution Approach to On-line Handwriting Segmentation and Feature Extraction", *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, 2004, pp 614–617, Washington, DC, USA. IEEE Computer Society.

[19] V. Vuori, M. Aksela, J. Laaksonen and E. Oja, "Adaptive character recognizer for a hand-held device: Implementation and evaluation setup", 2000, pp 13–22.

[20] V. Vuori, J. Laaksonen, E. Oja and J. Kangas, "Speeding Up On-line Recognition of Handwritten Characters by Pruning the Prototype Set", *ICDAR '01: Proceedings of the Sixth International Conference on Document Analysis and Recognition*, 00:501–505, 2001.

[21] L. Vuurpijl, R. Niels, M. van Erp, L. Schomaker and E. Ratzlaff, "Verifying the UNIPEN Devset", *IWFHR '04: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition (IWFHR'04)*, 2004, pp 586–591, Washington, DC, USA. IEEE Computer Society.

[22] M. Ziaratban, K. Faez and F. Faradji, "Language-Based Feature Extraction Using Template-Matching In Farsi/Arabic Handwritten Numeral Recognition", *ICDAR '07: Proceedings of the 9th International Conference on Document Analysis and Recognition*, 1:297–301, 2007.