Segmentation of Handwritten Numeral Strings in Farsi and English Languages

Javad Sadri, Ching Y. Suen, Tien D. Bui
Center for Pattern Recognition and Machine Intelligence (CENPARMI)
Concordia University, 1455 de Maisonneuve Blvd. West, Montreal, Quebec, Canada H3G 1M8
Tel.: (514)-848-2424-Ext: 7954, Fax.: (514)-848-4522
Email: {j_sadri, suen, bui}@cs.concordia.ca

Abstract: A new method for segmentation of unconstrained handwritten numeral strings is proposed, and its performance in segmentation of handwritten Farsi and English numeral strings is compared. Our segmentation method uses a combination of features from the foreground and background of the image. In order to find foreground features, an algorithm based on skeleton tracing is introduced for the first time. The skeletons of the connected digits are traversed in clockwise and anti-clockwise directions, and intersection points are mapped onto the outer contour to form our foreground features. In order to find background features, another new algorithm is proposed. Considering vertical projections of top and bottom profiles, two background skeletons are found, and feature points are extracted from them. Background and foreground feature points are connected together to construct segmentation paths. Finally each segmentation path is evaluated based on the properties of its left and right connected components. Experiments show our method can provide a list of good segmentation hypotheses for segmentation-based recognition systems. The results of our system are shown on both Farsi and English numeral strings.

Keywords: Digit Recognition, Numeral String Segmentation, Numeral String Recognition, Foreground Features, Background Features, Skeleton Tracing

1 Introduction

Segmentation and recognition of unconstrained handwritten numeral strings is one of the most difficult problems in the area of Optical Character Recognition (OCR). It has been a popular topic of research for many years, and it has many potential applications such as postal code reading, bank check processing, tax form reading, and recognition of many other business forms [1]. Generally numeral strings in both Farsi and English languages include isolated digits (with a lot of variations), touched, overlapped, and noisy or broken digits [2]. Some examples of these cases in both languages are shown in Figure 1.

![Figure 1: (a) Samples of numeral strings in Farsi, (b) Samples of numeral strings in English.](image)
In most languages, segmentation of the touching numerals is one of the main challenges in handwritten numeral string recognition systems. Although many methods have been presented for recognition of isolated Farsi digits [such as 3, 4, and 5], to the best of our knowledge nothing has been reported on separating of touching digits, and constructing segmentation paths in handwritten Farsi numeral strings. However for segmentation and recognition of handwritten numeral strings in English many algorithms have been proposed, and generally they can be classified into three categories. The first uses foreground features extracted from black pixels in the image [6], [7]. The second uses background features, extracted from white pixels in the image [8], [9], and the third category uses a combination of both these features [10]. As shown in the literature, approaches based on a combination of both foreground and background features have better results [10]. Also for the recognition of numeral strings, two approaches generally exist: segmentation-then-recognition, and segmentation-based-recognition [11]. In the former approach, the segmentation module provides a single sequence of hypotheses, where each subsequence should contain an isolated character, which is submitted to the recognizer. Since this technique does not use any contextual or recognition information for segmentation, it shows its limits rapidly when the correct segmentation does not fit the predefined rules of the segmentation module. In the latter strategy, first the segmentation module provides a list of candidate segments, then each candidate segment is evaluated by the recognition module. Finally the list is post-processed by using contextual information [11]. Although many methods have been proposed for segmentation and recognition of connected handwritten numerals, much more improvement is still required to build robust systems for practical applications. Especially we need new methods to reduce the number of candidate cutting paths to be tested, and decrease the rejection rate, while maintaining a good recognition performance. In this paper, we present a new method of generating candidate segmentation paths for touching numerals in Farsi and English Languages, with unknown lengths, and we compare its performance in both languages. In this method segmentation is based on combining features from the foreground and background of the image. The novelty of the method is that extracting features in the foreground is based on a new method called skeleton tracing, and in the background it is based on combining the vertical top and bottom projection profiles and their skeletons. Based on a combination of this information, and some global information from the string image, our algorithm tries to construct the best segmentation paths for separating the touched digits. Our proposed approach can be used for generating good hypotheses in segmentation-based recognition systems. This paper is organized as follows: Section 2 introduces isolated Farsi numerals, Section 3 presents the pre-processing steps, Section 4 presents our proposed method for feature extraction, Section 5 describes our algorithm for constructing segmentation paths, Section 6 presents the method for evaluating segmentation paths, Section 7 presents the experimental results, and finally we draw the conclusion.

2 Farsi Numerals

Some examples of Farsi numeral strings were presented in Figure 1-a. Here isolated Farsi digits are shown in Figure 2. As shown in this figure, there are more than one style of writing for some digits, such as 0, 4, and 6.

3 Pre-Processing

As seen in Figure 1, some images need to be smoothed by pre-processing steps. The method in [6] is used to remove the small spots of noise and to smooth the edges of the connected components. In the next step, to make the segmentation task easier and the segmentation paths as straight as possible, a method similar to [12] is used to correct the slant of each connected component. See Figure 3 for some results of our pre-processing. The details of our method for finding feature points in the foreground and background are explained in [13], here we briefly explain its application on segmentation of Farsi handwritten numeral strings.
4 Extracting Feature Points

4.1 Foreground Features

The procedure for finding foreground features of two different examples taken from our handwritten Farsi and English numeral strings are shown in Figure 4, and 5 respectively. To find feature points on the foreground (black pixels) of each connected component, we introduce a new algorithm which is called skeleton tracing. First by Zhang-Suen thinning algorithm [14], the skeleton of each connected component is extracted, (see Figures 4-b, 5-b). Then on these skeletons, two points called starting and ending points (denoted by S, and E) are found. Actually the starting point and ending points are points with lowest and highest value of the x-coordinates, respectively. Then from the starting point (S), the skeletons are traversed in two different directions: first clockwise, and then anti-clockwise until both traverses reach the same ending points (E), and stop. In Figures 4-c, and 5-c we define the traverse in the clockwise as top-skeleton, and the traverse in the anti-clockwise as bottom-skeleton. In the next paragraph, using these skeletons we explain how we can obtain important features for segmentation.

When the algorithm traverses the top/bottom skeletons, it looks for the intersection points (IPs) which are visited on the skeleton. IPs are points which have more than two connected branches (in Figures 4-b, and 5-b they are denoted by ○). Some intersection points on top/bottom skeleton may be visited more than once. Corresponding to each visit of any intersection point in the skeletons, there is an angle where its bisector can be found (Figures 4-c, 5-c). The intersections of these bisectors with the outer contour of the connected component are obtained, they are denoted by ☐ in Figures 4-d, and 5-d. These points form our foreground feature points. Therefore during the tracing of the top or bottom skeleton, bisectors map intersection points on the outer contour to form feature points. These feature points can be divided into two different subsets, one corresponding to top-skeleton and one corresponding to bottom-skeleton, and they are called top-foreground-features and bottom-foreground-features respectively. As shown in Figures 4-d, and 5-d each
feature point is very close to an intersection point on the skeleton. Therefore, these feature points can carry important information about the location of touching strokes, and segmentation regions.

4.2 Background Features

Lu et al. in [8], and Chen, and Wang in [10] used the skeleton of the background to extract features, and they used all white pixels of the background, including those inside the holes and inner parts of the digits. However, as seen in Figure 6-c, we do not consider all the white pixels as background. Here, a new method of finding background features is introduced. First, vertical top and bottom projection profiles of the connected component are found, as depicted in Figures 6-d and 6-e. Then, skeletons of these images (black regions) are extracted, which are shown in Figures 6-f, 6-g, and they are called top-background-skeleton and bottom-background-skeleton respectively.

Figure 6: (a) Original image, (b) Pre-processed and slant corrected image, (c) Background region (White pixels outside of the black object), (d) Top projection profile, (e) Bottom projection profile, (f) Top-background-skeleton, (g) Bottom-background-skeleton.

These skeletons normally have long or extra branches or tails, which sometimes do not lead us to reach to the cut regions, an example shown in Figure 7-a. We use a simple algorithm to remove these long or extra parts. As shown in Figure 7-b all parts of top-background-skeleton which are lower than the middle line, and all parts of bottom-background-skeleton which are higher than the middle line of the image are removed. Then on each background skeleton (top / bottom), end points (points which have just one black neighbor and they are denoted by □ in Figure 7-b) are found, and they are inserted into two different lists: one for end-points on top-background-skeleton, and another list for end-points on bottom-background-skeleton. The first and last end points in each list will not be considered. Compared to [8, and 10], our background feature points are more informative, and we also have a smaller number of feature points in the background, which makes decision making easier in the next step. The same procedure for finding background features in Farsi is used, and an example is shown in Figure 8.

Figure 7: (a) Original top / bottom-background-skeletons, (b) Top / Bottom-background-skeletons after removing parts which are lower/higher than the middle line (Horizontal dashed line is the middle line, which divides the image into two parts with equal heights).

Figure 8: (a) Pre-processed and slant corrected image, (b) Background region (White pixels outside of the black object), (c) Top/Bottom background skeletons, (d) Top / Bottom-background-skeletons after removing parts which are lower/higher than the middle line.

5 Constructing Segmentation Paths

In the previous steps (4.1, and 4.2), four different lists of features are obtained: two lists of foreground features (top/bottom-foreground-features), two lists
of background features (top/bottom-background-features). Now two ways of searching are conducted: upward searching, and downward searching. If feature points in these lists from top to bottom, or from bottom to top, match with each other, they are assigned together to construct segmentation paths. Two feature points \( A \) and \( B \) from two different lists match with each other if condition (5-1) is met.

\[
|x_A-x_B| \leq \alpha \cdot H , \quad \alpha \in [0.25, 0.5], \quad (5-1)
\]

Here \( x_A \), and \( x_B \) are the horizontal coordinates of \( A \), and \( B \) respectively, \( \alpha \) is a constant, which in our experiments we select it equal to 0.4, and \( H \) is the vertical height of the input connected component. The flowchart in Figure 9 shows the details, and an example is depicted in Figure 10.

\[ \text{Figure 9: Flowchart of downward/ upward searching for constructing segmentation paths.} \]

6 Evaluation of Segmentation Paths

Each segmentation path divides a connected component into two new connected components: one is on the left side and the other one is on the right side. Here each segmentation path is evaluated individually (without considering other candidate paths). Two constraints, on the left and right side connected components of each path, are applied as below. Paths which satisfy these two constraints are saved, and other paths are removed. (Figure 11 shows all the parameters which are used in these constraints)

\[ \text{Figure 10: Feature points on the background and foreground (from the top and bottom) are matched and assigned together to construct segmentation paths.} \]

\[ \text{Figure 11: Illustration of all parameters which are used in the evaluation of the segmentation paths.} \]

- Overlap constraint: if horizontal overlap between the left and right side connected components of a segmentation path is above a threshold (see 6-1), the segmentation path is rejected. Here overlap is the horizontal overlap between the left and right side connected components, and \( W_l, W_r \) are their widths respectively (see Figure 11), and \( T_{\text{ovl}} \) is a threshold. We select \( T_{\text{ovl}} \) equal to 0.6 in our experiments.

\[
\frac{\text{Overlap}}{\min(W_l, W_r)} \leq T_{\text{ovl}} \quad (6-1)
\]

- Height constraint: if ratio of the heights of the left and right side components of a segmentation path, is below a threshold (see 6-2), then the segmentation path is rejected. Here \( H_l \), and \( H_r \) are the heights of the left and right side connected components respectively (see Figure 11), and \( T_{\text{Height}} \)
is a threshold. We select $T_{\text{Height}}$ equal to 0.4 in our experiments.

$$\frac{\min(H_t, H_r)}{\max(H_t, H_r)} \geq T_{\text{Height}} \quad (6 - 2)$$

7 Experimental Results

Our Farsi touching digits come from a new data-base which is currently under construction in CENPARMI (Center for Pattern Recognition and Machine Intelligence), for handwritten Farsi alphanumeric Character Recognition. Currently we have about 88 samples of touching digits in this database. Our English touching digits come from a standard database (NSTRING SD19) which has 2099 touching digits for testing. To evaluate our method, we selected randomly 900 test samples from this database, and also we used all the Farsi touching digits that we have (88 samples). We have carried out some experiments on these two sets of samples. In These experiments our system does not know the length of the string as prior knowledge. After segmentation of the numerals by the system, we did a visual analysis and verified that in 76% of the cases for Farsi touching digits, the best segmentation path is among the paths produced by the system, and for English touching digits 95.1% of the cases, the best segmentation path is among the paths produced by the system. Because the system does not know the length of the input numeral string, for some images it produces more than one cutting path (see Figure 12). Most cases of error in Farsi touching digits are due to low quality of our images after binarization, or special shapes of some numerals such as 4, 7, and 8. However for English touching digits, errors are due to high overlaps of connected digits. Figure 12 illustrates some results of our system.

8 Conclusions and Future Works

A new method of generating segmentation candidates for touching digits in numeral strings is presented, and its performance in Farsi, and English is compared. Segmentation candidates are found by combining foreground and background features, and new algorithms for finding feature points are presented. Due to the variability of handwriting, we found that foreground and background features complement each other. Although we did not use any prior knowledge about the length of the string in our system, nor we included any recognition feedback for decision making, our experiments on touching digits show very encouraging results. Our experiments also show further investigation and experiments are required in order to develop more sophisticated approaches for segmentation of handwritten Farsi touching digits. In our future studies, we are going to add more samples to our Farsi touching digit database and also to add a recognition module (recognition feedback) to our system, to rank the multiple segmentation hypotheses by the recognition scores.

Acknowledgment

Authors would like to express their sincere thanks to Mr. Farshid Soleymani for providing sample images of Farsi numeral strings.

References


