An approach for locating segmentation points of handwritten digit strings using a neural network

Daekeun You and Gyeonghwan Kim
Dept. of Electronic Engineering
Sogang University
CPO Box 1142, Seoul 100-611 Korea
gkim@ccs.sogang.ac.kr

Abstract

An approach for segmentation of handwritten touching numeral strings is presented in this paper. A neural network has been designed to deal with various types of touching observed frequently in numeral strings. A numeral string image is split into a number of line segments while stroke extraction is being performed and the segments are represented with straight lines. Four types of primitive are defined based on the lines and used for representing the numeral string in more abstractive way and extracting clues on touching information from the string. Potential segmentation points are located using the neural network by active interpretation of the features collected from the primitives. Also, the run-length coding scheme is employed for efficient representation and manipulation of images. On a test set collected from real mail pieces, the segmentation accuracy of 89.1% was achieved, in image level, in a preliminary experiment.

1. Introduction

In areas of document image analysis and recognition, the correct interpretation of digit strings, including street numbers and amounts, is of crucial importance. However, touching between digits in handwritten scripts is common place. Issues that related to the segmentation and recognition of handwritten strings have been dealt with various ways for last decades[1]. It is well known that the difficulties stem from not only variations existing in shape but also various ways of touching. And, in many cases, the segmentation and the recognition could not be considered independently when touching between characters/digits is assumed. There have been three strategies for dealing with the segmentation problem: *segmentation-based (dissection)*[2, 4, 6, 8, 13], *recognition-based*[3, 5, 7, 12], and *holistic*[9] approaches.

Correct segmentation is important since errors made in the segmentation stage get compounded by the time the recognition of the segmented object is performed. Algorithms dealing with digit segmentation that have been reported in the literature are primarily rule-based. Types of touching are analyzed and rules for locating potential segmentation points are developed[11].

In this paper, we propose a method for locating possible segmentation points in a digit string image using a neural network. To provide inputs for the neural network, we define primitives that represent strokes in an abstractive manner. Based on many observations, we assume that segmentation points are located mainly in horizontal primitives. Therefore, for each horizontal primitive, features surrounding the primitive are examined and provided to the neural network as inputs, and the neural network determines whether a segmentation point can be located in the primitive under examine. Performance of the proposed scheme is compared with a rule-based method and some of particular cases are illustrated. Also, we introduce the run-length coding scheme[10] for image manipulation to make the overall process more efficient. Figure 1 illustrates the overview of the approach.

Organization of this paper is as follows. Section 2 describes the image representation scheme briefly. Section 3

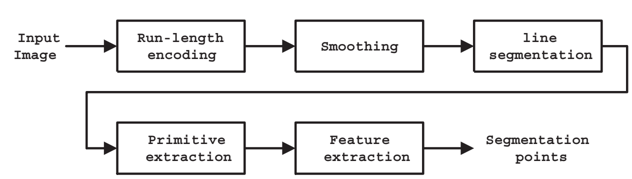


Figure 1. Overview of the proposed segmentation method

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 **IEEE**

explains the definition of the primitives and steps for extracting the primitives in an input image. Section 4 is about locating segmentation points based on the primitives extracted. Experimental results including failure analysis is in Section 5 and some concluding remarks are in Section 6.

2. Image manipulation

Instead of conventional pixel-based image representations, we adopt a run-length coding scheme and all procedures for image manipulation are performed based on the representation. By employing the image manipulation scheme, block-based image processing is possible and we can take full advantage of the scheme in subsequent procedures, especially in primitive extraction. To make effective representation of handwritten strokes, both of horizontal runs(Hrun) and vertical runs(Vruns) are utilized.

3. Extraction of primitives

Primitives are defined to represent a digit string in an abstractive way. Four steps for extracting primitives are described in this section. In the description, we assume that the input image shown in Figure 2(a) is used in common.

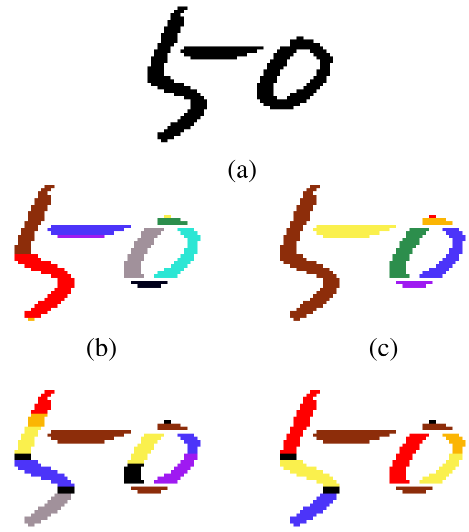
step 1: coarse line segmentation

Coarse line segmentation is performed on the input string image with Hrun representation. Branch runs, in which adjacent runs form a branch, are primarily used for the segmentation. As we can see in Figure 2(b), the first stage segmentation is performed by simply omitting the branch runs, and segments are identified. Also, significant change observed in the width of a segment, while a connected component is being traced in run by run, is a good indication of segment boundary and can be used as a clue for the step.

step 2: merging line segments based on the segment width

The process in the step 1 is very sensitive to sudden changes could happen in width of runs due to noise, for example. Therefore, the segments produced in the step 1 are examined and merged if the following conditions are met. (In the description, we define a horizontal segment as one with avg_width $\geq \mu_t + 2\sigma_t$, where μ_t is the average stroke with obtained from all runs in the image, excluding branch runs, ${\sigma_t}^2$ is the variance, and avg_width is the average width of the segment.)

1. two consecutive ones are horizontal segments,



(a)

(b) (c)

(d) (e)

Figure 2. Line segmentation: (a) input image, (b) coarse segmentation, (c) merging based on the width, (d) segmenting based on the directions, and (e) merging based on the slopes

2. neither of two consecutive ones are horizontal segments, and the differences between the starting points and the ending points of the end_run of the upper segment and the start_run of the lower segment are less than μ_t, or
3. both of the consecutive segments are consisted of single runs.

Figure 2(c) shows a result of the step 2.

step 3: segmentation based on directions

Runs in a segment are examined and new segmentation boundaries are introduced based on directions. Types of directions are defined as shown in Figure 3, and assigned by comparing the center points of two consecutive runs. First row of Figure 4 illustrates the directions of the first connected component of "5" in Figure 2(c). By applying small size 1-D masks, such as 1×3 and 1×5, spurious changes in directions are eliminated. Remaining rows of Figure 4 show how the masks work.

Figure 2(d) shows the segmentation result after considering directional changes of strokes.

step 4: merging line segments based on slopes

In this step, slope information is utilized for refining the segmentation result. The directional information, used in

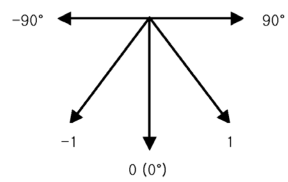


Figure 3. Convention of the directions

```
-1-1 1-1 0-1 0-1 0-1-1-1-1-1 0-1-1 0 0 1 1 1 1 1
-1-1-1-1-1 0-1 0 0 0-1-1-1-1-1-1-1-1 0 0 1 1 1 1
-1-1-1-1-1-1 0 0 0 0-1-1-1-1-1-1-1-1 0 0 1 1 1 1
-1-1-1-1-1-1 0 0 0 0-1-1-1-1-1-1-1-1 0 0 1 1 1 1
```

Figure 4. Filtering spurious direction changes

step 3, is obtained from consecutive runs and has rough resolution as we can see in Figure 3. Therefore, the notion of slope is introduced and obtained by connecting centers of start_run and end_run of a segment. The following rules are used to merge two consecutive segments.

1. for a vertical segment, if a segment above and a segment below have the same direction both can be merged.

2. if neither of two consecutive segments are vertical but both have same direction, and if the slope difference is less than the threshold (30 degree) the segments can be merged.

3. new slope of the merged segments is calculated, and depending on the sign of the slope, new direction is assigned based on the convention shown in Figure 3.

4. above rules are applied for the consecutive segments until the last segment is reached.

Figure 2(e) shows the final result of the line segmentation after applying the procedures described in this section. Horizontal segments are represented using Vrun structure to take full advantage of the Vrun structure in segmenting horizontal segments.

Four types of primitive are defined as shown in Figure 5. The line segments obtained are approximated and one of the primitive_id is assigned according to the slope. Figure 6 shows that how strokes can be approximated by the primitives.

4. Locating segmentation points

Potential segmentation points are located by analyzing the primitives, instead of the raw image or the segments.

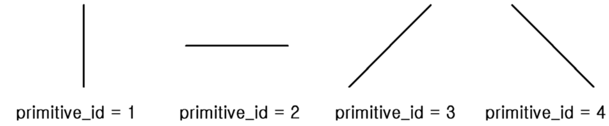


Figure 5. Four types of primitive

There could be numerous ways to locating the points by interpreting relations between primitives. In this section, we introduce the NN-based approach as an effective way for the purpose. In addition, rule-based approach is described briefly because the approach is used for the comparison purpose in Section 5. In the implementations, we assume that the segmentation points are located in horizontal primitives according to many observations.

Also, over-segmentation is allowed and the segmentation points located are interpreted and consumed by digit recognizers with various schemes, in later stages.

4.1. NN-based approach

As a means to interpreting various ways of touching between digits, a neural network-based approach has been considered. In contrast to rule-based approaches, generalization of ways of touching is expected and numerous thresholds used in decision making are determined while training is performed.

Input parameters to the neural network are properties of primitives and listed as followings:

- *the relative position of the horizontal primitive under examination*: the feature is represented by the equation below.

$$f_0 = (C_y - S_y)/CC_h$$

  where C_y and S_y are y-coordinates of the baseline and the horizontal primitive under examination, respectively, and CC_h is the height of the connected components in the image.

- *the existence of nearby segmentation points*: for each horizontal primitive, two features (left and right) indicate whether the horizontal primitive is segmented into more than one due to vertical crossings.

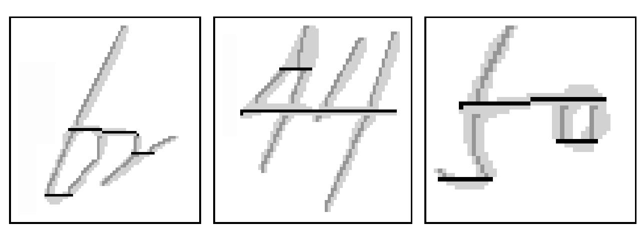


Figure 6. Strings represented by primitives

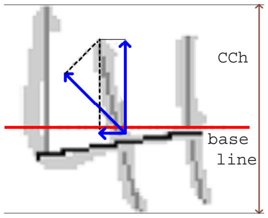
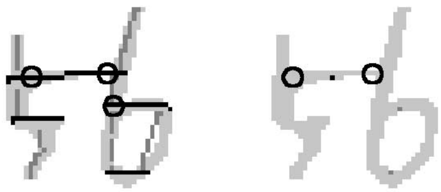


Figure 7. Features extracted from the primitives



(a) (b)

Figure 8. Segmentation points: (a) rule-based approach and (b) NN-based approach

- *the normalized length of adjacent primitives*: normalized lengths of four adjacent primitives.
- *directional components of adjacent primitives*: directional components obtained by projection of three adjacent primitives in the four directions which are defined in Figure 3.

Therefore, the number of features for the neural network becomes 23. Figure 7 shows the features used as inputs to the neural network. In the figure, a baseline (a dotted line in the middle) is determined using y-coordinates of major primitives which have lengths longer than the average stroke width.

Figure 8 shows segmentation points located by applying the rule-base approach(a) and the NN-based approach(b).

4.2. Rule-based approach

A number of rules are developed to locate possible segmentation points. In the first stage, horizontal primitives are clustered so to virtually combine overlapped horizontal primitives. The motivation of the clustering is from the observation that major horizontal primitives are overlapped in a digit as long as the digit is written without significant slant.

Once the clustering process is completed, primitives which belong to each cluster is examined to locate possible segmentation points.

5. Experimental results

1,587 street number images, collected from live Korean mail pieces, were used for the performance evaluation. The images have resolution of 200dpi. Among the images, 78.6% of them contain no touching digits and 19.2% of them contain two touching digits, which take 97.8% in total. Therefore, in the experiments we do not consider images with more than two touching digits.

Contour smoothing and elimination of simple noises are only preprocessing steps applied to the images. In this section, we describe the performance of both segmentation algorithms[1].

5.1. Performance of the Rule-based approach

To evaluate the performance of the rule-based approach, a set of 180 touching string images, non of them were involved while the rules were developed, were used. Segmentation points determined by the algorithm were examined in a subjective manner, and as long as the points included the correct segmentation point we regarded the image as a success. As the result, the algorithm located correct segmentation point for 82.8% of the images. Major cause of failure turns out to be from wrong clustering of the primitives.

5.2. Performance of the NN-based approach

80 images were used for training the neural network. From the images, 255 training patterns were collected by visual examination with desirable segmentation points marked manually. The network was tested on a set of 220 images comprising of ones not used in the training stage. Over-segmentation is allowed in the testing phase since the recognition procedure will combine the segments anyway. This gives an effective segmentation accuracy of 89.1%.

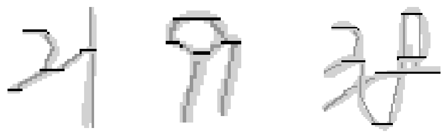
The approach located 1∼2 segmentation points for 65% of touching strings, which have relatively simple types of touching. On the other hand, more number of segmentation points was produced for strings comprising of digits with complicated shapes, such as “6” and “8”.

Figure 9 shows the cases in which the NN-based approach located more accurate segmentation points against the rule-based approach. The NN-based approach perfectly has dealt with failure cases observed in the rule-based approach due to primitive clustering errors. The observation proves that the trained neural network can decode the types of touching as we have expected.

We examined images classified as failures. Main sources of error have been identified:

[1] At the time when the paper is preparing, the NN-based approach is under development. Only preliminary results, but good enough to show the effectiveness of the approach, are provided.

Figure 9. Segmentation points located by the NN-based approach



(a) (b) (c)

Figure 10. Main sources of error: (a) the touching type is missed, (b) narrow horizontal primitive, and (c) touching in vertical direction

- *touching types*: some particular touching types are not provided to the neural network in the training stage (Figure 10(a)). It is expected that the cases can be resolved easily by providing the patterns to the network.
- *segmentation points*: some of primitives are ignored due to peculiarity in shape. (Figure 10(b) is a case where the length of the horizontal primitive is less than the average stroke width.)
- *touching in vertical direction*: a touching occurs in vertical direction cannot be handled because the network assumes that segmentation points are located in horizontal primitives. (Figure 10(c))

It is expected that most of the errors can be dealt with by further analysis of touching types and providing the corresponding features to the neural network.

6. Conclusion

A NN-based approach for locating segmentation points in a string with touching digits has been introduced. The notion of primitives is introduced to represent the strokes in an abstractive manner, and to simplify the types of touching eventually. Once the primitives are extracted from the input string, features for the neural network are collected. The trained neural network decodes types of touching and determines segmentation points. A rule-based approach has been implemented for the comparison purpose. The NN-based approach resulted in better performance, even it is obtained in the early stage of the development. In addition, according to the failure analysis it is expected that further significant improvement is expected. All of the image manipulation procedures were implemented using the run-length coding scheme.

References

[1] R. G. Casey and E. Lecolinet. Strategies in character segmentation : A survey. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1028–1033, 1995.

[2] Y. K. Chen and J. F. Wang. Segmentation of single- or multiple-touching handwritten numeral string using background and foreground analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1304–1317, November 2000.

[3] G. Congedo, G. Dimauro, S. Impedovo, and G. Pirlo. Segmentation of numeric strings. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1038–1041, 1995.

[4] H. Fujisawa, Y. Nakano, and K. Kurino. Segmentation Methods for Character Recognition: From Segmentation to Document Structure Analysis. *Proceedings of the IEEE*, 80(7):1079–1092, 1992.

[5] T. M. Ha, M. Zimmermann, and H. Bunke. Off-line handwritten numeral string recognition by combining segmentation-based and segmentation-free methods. *Pattern Recognition*, 31(3):257–272, 1998.

[6] K. Kim, J. Kim, and C. Suen. Segmentation-based recognition of handwritten touching pairs of digits using structural features. *Pattern Recognition Letters*, 23:13–24, 2002.

[7] S.-W. Lee and S.-Y. Kim. Integrated segmentation and recognition of handwritten numerals with cascade neural networks. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 29(2):285–290, February 1999.

[8] Z. Lu, Z. Chi, W. Siu, and P. Shi. A background-thinning-based approach for separating and recognizing connected handwritten digit strings. *Pattern Recognition*, 32:921–933, 1999.

[9] S. Madhvanath. *The Holistic Paradigm in Handwritten Word Recognition and Its Application to Large and Dynamic Lexicon Scenarios*. PhD thesis, Dept. of Computer Science, State University of New York at Buffalo, 1996.

[10] T. Pavlidis. *Algorithms for Graphics and Image Processing*. Computer Science Press, Rockville, MD, 1982.

[11] Z. Shi and V. Govindaraju. Segmentation and Recognition of Connected Handwritten Numeral Strings. In *The Fifth International Workshop on Frontiers in Handwriting Recognition, Essex, England*, pages 305–308, 1996.

[12] N. W. Strathy and C. Y. Suen. A new system for reading handwritten zip codes. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 74–77, 1995.

[13] D. Yu and H. Yan. Separation of touching handwritten multi-numeral strings based on morphological structural features. *Pattern Recognition*, 34:587–599, 2001.