Handwritten Numeral String Recognition with Stroke Grouping

Chan E. Cheong, Ho-Yon Kim, Jang-Won Suh, Jin H. Kim
Department of Computer Science, KAIST
373-1 Kusong-dong Yusong-Gu, Taejon, 305-701 Korea
{chan, hykim, suh, jkim}@ai.kaist.ac.kr

Abstract

In this paper, a framework for off-line handwritten numeral string recognition based on stroke grouping is proposed. In our approach, strokes are aligned into a sequence of strokes and then segmentation process is performed to partition strokes in the sequence into possible-digits, that is, groups of strokes which may be possibly a digit. As the result of stroke grouping, grouping-hypotheses, which imply possible segmentation, are generated. An input numeral string is recognized by dynamic programming scheme, in which the best grouping-hypothesis with maximum matching score is chosen. The framework also provides systematic way of reducing computational complexity by embedding external knowledge into the framework. The experimental results to evaluate the proposed framework are shown.

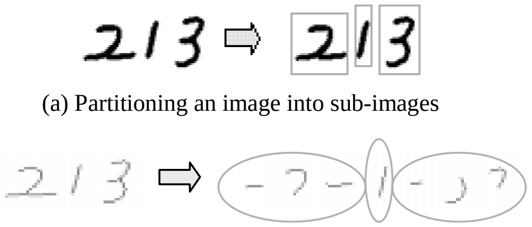
1. Introduction

In order to automatically process handwritten documents by computer, handwriting recognition is essential. Research on off-line handwritten numeral string recognition has been intensively performed for the last two decades due to diverse applications such as reading zip codes on envelopes, courtesy amount on checks, data filled in forms, and so on.

The major problems of handwritten numeral recognition are caused by digits that are overlapped, that touch each other, and that are broken into several fragments. Moreover symbols such as hyphen, comma, ligature between adjacent digits must be either removed or recognized. In contrast with word recognition, the difficulties in recognizing handwritten numeral strings mainly come from the fact that there is no available context in numeral strings. This means that handwritten numeral string recognition must be done digit by digit. Accordingly, the segmentation process is emphasized.

There are two approaches for handwritten numeral string recognition. One is that a numeral string is considered as a sequence of images of individual digits as shown in Fig.1(a) [1][2][3]. From this point of view, recognizing a numeral string is similar to partitioning the image into those of individual digits. In the other approach, a digit is treated as a set of strokes and then basic operations are conducted at stroke level [4][5][6]. Our approach can be categorized into the latter.

In this paper, a framework for off-line handwritten numeral string recognition based on stroke grouping is proposed. In our approach, strokes are aligned into a sequence of strokes as shown in Fig.1(b) and then segmentation process is performed to partition strokes in the sequence into *possible-digits*, that is, groups of strokes which may be possibly a digit. See Fig. 2. As the result of stroke grouping, *grouping-hypotheses*, which imply possible segmentation, are generated. An input numeral string is recognized by dynamic programming scheme, in which the best grouping-hypothesis with maximum matching score is chosen. Fig. 2 illustrates the overview of the proposed recognition process.



(a) Partitioning an image into sub-images

(b) Partitioning a stroke sequence into groups of strokes

Fig. 1. Viewpoints of recognition

In comparison with conventional stroke-based approaches, the characteristics of the proposed framework are summarized as follows: Firstly, possible-digits are produced by sequential stroke grouping instead of connected components that are commonly used in most conventional methods. Therefore, the importance of heuristic decision functions in segmentation process is considerably alleviated. Secondly, neither pre-segmentation process for handling broken fragments nor connected component-based segmentation scheme is applied to this framework. Lastly, knowledge utilized for stroke grouping can be systematically represented in the framework.

The organization of this paper is as follows: section 2 describes the framework of stroke grouping and section 3 presents knowledge embedding for complexity reduction. Experimental results and discussions are included in section 4, followed by concluding remarks.

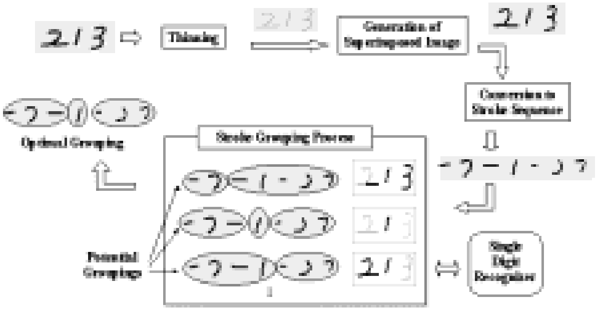
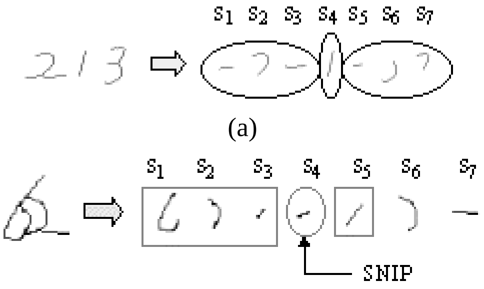


Fig. 2. Recognition process

2. Recognition with stroke grouping

2.1. Generation of stroke sequence

The proposed approach is based on the fact that a numeral string can be considered as a sequence of strokes. A stroke is defined as a thin line that has either an end point or a branch point at each end. In representing a numeral string with a sequence of strokes, many different sequences can be obtained from the same numeral string depending on an ordering function. In this paper, strokes in a numeral string are aligned with a heuristic measure, which is the center of mass of each stroke in terms of its x-coordinate. After the strokes are aligned with respect to their x-coordinates, each stroke is labeled by increasing integers from left to right.



(a)

(b)

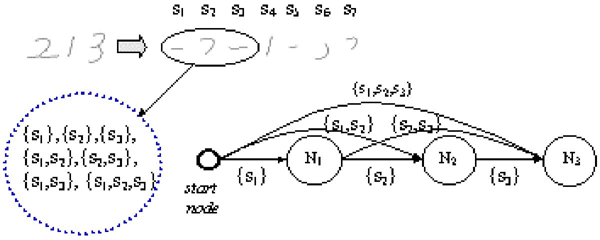
Fig. 3. Stroke sequences

In general, in a stroke sequence as shown in Fig. 3(a), the strokes composing each digit are in near vicinity and the strokes of the former digit appear in front of the strokes of the latter one. However, some strokes involved in the latter digit may be placed in between the strokes involved in the former digit as shown in Fig. 3(b). Such a stroke is called a *stroke-not-in-place* (SNIP) in this paper. SNIPs appear when consecutive digits are severely overlapped.

2.2. Stroke grouping

A stroke sequence is a set of strokes and then can be written as follows: $S = \{s_1, \ldots, s_T\}$, where s_i is stroke and T is the length of the stroke sequence. At each position of S, denoted as t and $1 \leq t \leq T$, a sub-sequence is produced. Let R_t be the sub-sequence at position t of S, and defined as $R_t = \{s_i \mid s_i \in S, 1 \leq i \leq t\}$. Stroke grouping at position t is conducted for R_t. The possible-digits to be produced from R_t are all the subsets of R_t. For example, in Fig. 4(a), sub-sequence R_3 at $t = 3$ is $\{s_1, s_2, s_3\}$ and 7 possible-digits are produced from R_3.

Stroke grouping can be represented with a directed graph as shown in Fig. 4(b). In the graph, an arc indicates a possible-digit and a path from the start node to the end node corresponds to a grouping-hypothesis of the given sub-sequence. A node corresponds to a unique sub-sequence for stroke grouping and the strokes in the sub-sequence must be consumed in each path coming into the node from the start node. Fig. 4 shows an intermediate result of stroke grouping for the sub-sequence $R_3 = \{s_1,s_2,s_3\}$. All paths from the start node to node N_3 correspond to grouping-hypotheses and the strokes in R_3 identified by node N_3 are all consumed in each path.

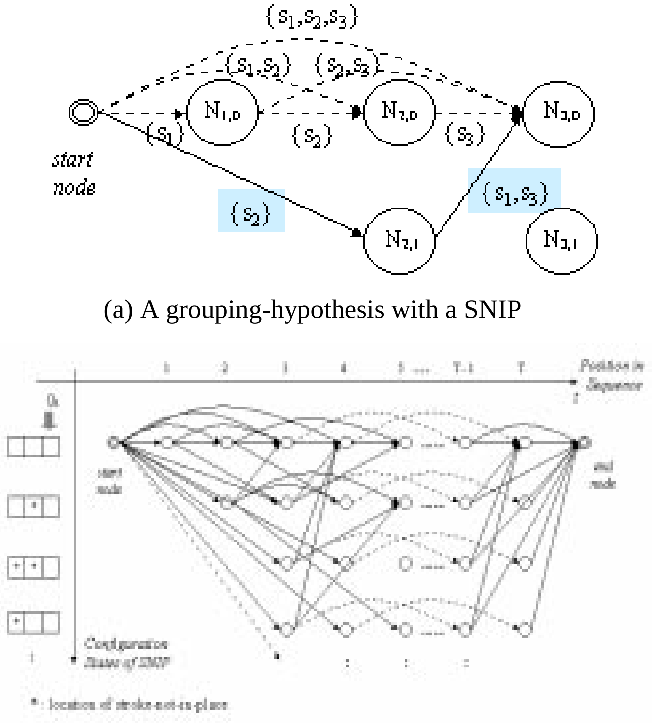


(a) Possible-digits (b) Representation in directed graph

Fig. 4. An example of stroke grouping

By the way, a grouping-hypothesis composed of $\{s_2\}\{s_1,s_3\}$ was not represented in Fig.4(b). This grouping-hypothesis means that possible-digit $\{s_2\}$ was produced and then possible-digit $\{s_1,s_3\}$ was produced. In terms of strokes, stroke s_2 had been consumed before stroke s_1 was consumed in the next possible-digit $\{s_1,s_3\}$. According to our definition described in section 2.1, stroke s_1 is a SNIP. In Fig. 4(b), however, there is no node representing the possible-digit $\{s_2\}$ with allowing stroke s_1 to be consumed in the next possible-digit. It is obvious that a typical one-dimensional graph is not enough to represent all grouping-hypotheses with respect to SNIP. More generalized representation is required.

In our approach, a new node in each position is added into the graph that results in a lattice as shown in Fig. 5(a). A node in the second layer identifies a sub-sequence that all strokes are consumed except for a preceding stroke, which is a SNIP. A sub-sequence corresponding to a node $N_{t,i}$ is now denoted as $R_{t,i}$, where t is a position in S and i indicates a layer in the graph. Adding a node in each position that implies the specific location of a SNIP enables to represent the grouping-hypothesis $\{s_2\}\{s_1,s_3\}$ which is represented as the path of solid arrows in Fig. 5(a).



(a) A grouping-hypothesis with a SNIP

(b) Overall representation

Fig. 5. Representation of stroke grouping

In general, it is impossible to find out which stroke is a SNIP and how many SNIPs there are in a stroke sequence before recognition is done. Therefore, the representation shown in Fig. 5(a) must be expanded to permit that every stroke could be a SNIP and that there may be more than one SNIP. All possibilities of occurrence of SNIPs, called configuration states, can be represented in the lattice structure depicted in Fig. 5(b). The number of possible states can be expanded to 2^{T-1} theoretically, where T is the number of strokes. In the structure, the end node is added in order to consume SNIPs left in every state at T.

2.3. Matching in lattice

The framework explained so far can be formulated based on dynamic programming. Let $\delta_{t,\ j}$ be the accumulated matching score at node $N_{t,\ j}$, which is in configuration state j of position t. To choose the best grouping-hypothesis with maximum matching score at the node is

$$\delta_{t,\,j} = \max_{d,\,I} \left[\delta_{t\,-\,d,\,i} \cdot C(\Delta R)\right], \qquad 1 \le t \le T,\ 0 < d < t,$$
$$0 \le i < j < 2^{t-1}$$
$$\zeta_{t,\,j} = \arg\max_{d,\,i} \left[\delta_{t\,-\,d,\,i} \cdot C(\Delta R)\right],$$

where $\Delta R \equiv R_{t,\,j} - R_{t\,-\,d,\,i}$. Consequently, ΔR is a newly generated possible-digit between node $N_{t,\,j}$ and $N_{t\,-\,d,\,i}$. $C(\Delta R)$ is the confidence resulting from the recognition of ΔR. If t - d indicates the start node, $R_{t\,-\,d,\,i}$ is treated as a null set.

Like other methods operated at stroke level, the proposed framework also has a high computational complexity. Let N_s be the number of configuration states and T be the number of strokes. Then the complexity of the proposed framework is up to N_s^2 x T. For practical implementation, N_s can be considerably reduced by applying heuristic measures for the generation of grouping-hypotheses.

3. Knowledge embedding

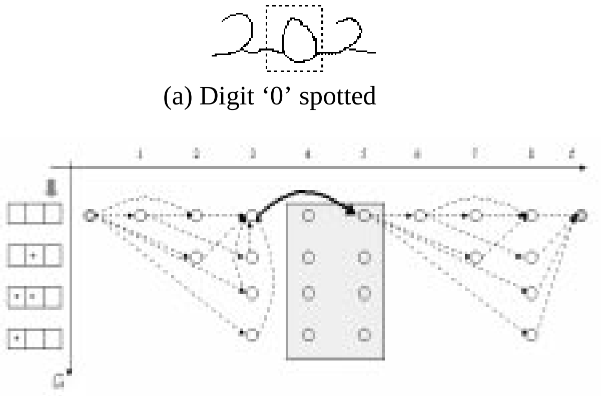
In the handwritten numeral string recognition field, it is natural to utilize knowledge about digits themselves and the spatial relations between them in an input image. Complexity is reduced by the spatial relations between digits in most conventional methods which are based on connected components: After each connected component is recognized, segmentation process is applied to only the connected components that are not accepted as a digit.

For the same purpose, knowledge of relative location between digits is used for building possible-digits in the proposed framework. The knowledge is based on the fact that when a group of strokes is known to form a single digit, strokes *before* that group cannot be combined with strokes *after* it to form a digit. In the numeral string of Fig. 6(a), if digit ‘0’ can be definitely spotted out, possible-digits composed of strokes across the digit ‘0’ cannot be generated. Therefore, strokes involved in digit ‘0’ are consumed for only one possible-digit as in the shaded area of Fig. 6(b). In terms of stroke sequence, this causes that an input stroke sequence is divided into two shorter stroke sequences and, consequently, stroke grouping is performed in each sequence. Embedding knowledge in the generation of possible-digits results in considerable reduction of computational complexity.

A method of spotting digits by identifying key shapes, called *target digit spotting*, is proposed in this paper. A key shape is a feature that can be detected only in some digit(s) with high probability. A hole is a good example of key shape. A target digit is defined as the digit containing a stroke with some specific key shape. For instance, if a hole is extracted as a key shape, then digit ‘0’, ‘6’, ‘8’, ‘9’ can be the target digits. Key shapes and their target digits are predefined by the statistical data obtained from a numeral database for single digit recognition.

When a stroke with a key shape is found, its neighboring strokes are gathered according to the models of target digits related to the key shape. Then strokes gathered are fed into a single digit recognizer in order to verify whether it is the expected target digit. The acceptance of a target digit is determined both by the confidence of recognition and by the comparison between the expected digit and the recognized result given by a single digit recognizer.

Unlike connected component-based methods, target digit spotting can cut out digits that are even touching adjacent ones as well as isolated. Furthermore, the result of target digit spotting can be systematically represented in the proposed framework.



(a) Digit ‘0’ spotted

(b) Representation in lattice structure

Fig. 6. Target digit spotting

4. Experimental results

To evaluate the performance of the proposed approach, experiments were conducted on unconstrained numeral strings from the KAIST databases, in which data were collected from student registration cards. Table 1 shows the experimental results. In the table, the number of the tested images is 100 numeral strings per length and the complexity indicates how many times a single digit recognizer was called to recognize all images in each length. The overall recognition rate obtained is 85% at zero-rejection level. The knowledge about the length of an input numeral string was not used and the recognition result was counted by numeral string. The single digit recognizer is a neural network-based one using multi-resolution statistical features. The more details are described in [7].

The majority of errors occurred when a single digit was recognized as two separate digits. This phenomenon happens when the confidence for a ‘whole’ digit is less than the product of the confidences for two ‘partial’ digits. This fact suggests that additional structural information should be used in building possible-digits.

Length	2	3	4	5	6	7
Rec.Rate(%)	92	84	82	89	83	80
Complexity	2469	4287	6877	3609	6201	5696

Table 1. Experimental results

5. Conclusion

A framework for off-line handwritten numeral string recognition with stroke grouping is proposed. Stroke grouping, which is based on dynamic programming, with the concept of SNIP and the target digit spotting for complexity reduction are newly introduced in this paper. The implemented system has been tested on 600 images of unconstrained numeral strings from the KAIST database and the recognition rate of 85% is obtained. Even though further improvements are needed, experimental results are encouraging and our approach is on the right track.

References

[1] Z. Shi, S. N. Srihari, Y-C. Shin, and V. Ramanaprasad, “A system for segmentation and recognition of totally unconstrained handwritten numeral strings,” in Proc. of the 4nd International Conference on Document Analysis and Recognition, (Ulm, Germany), pp. 455-458, 1997.

[2] B. Zhao, H. Su, and S. Xia, “A new method for segmenting unconstrained handwritten numeral string,” in Proc. of the 4nd International Conference on Document Analysis and Recognition, (Ulm, Germany), pp. 524-527, 1997.

[3] N. W. Strathy and C. Y. Suen, “A new system for reading handwritten zip codes,” in Proc. of 3rd International Conference on Document Analysis and Recognition, (Montreal, Canada), pp. 74-77, 1995.

[4] T. M. Ha, M. Zimmermann, and H. Bunke, “Off-line Handwritten Numeral String Recognition By Combining Segmentation-based and Segmentation-free Methods,” in Pattern Recognition, vol. 31, no. 3, pp. 257-272, 1998.

[5] H. Nishida and S. Mori, “A model-based split-and-merge method for character string recognition,” in International Journal of Pattern Recognition and Artificial Intelligence, vol. 8, no. 5, pp. 1205-1222, 1995.

[6] S. Seshadri and D. Divakumar, “A technique for segmenting handwritten digits,” in Proc. of the International Workshop on Frontiers in Handwriting Recognition III, (Buffalo, NY), pp. 443-448, 1993.

[7] C-L Liu, Y-J Liu, and R-W Dai, “Preprocessing and Statistical/Structural Feature Extraction for Handwritten Numeral Recognition,” in Progress in Handwritting Recognition, pp. 161-168, A.C.Downton and S. Impedovo, (Eds.), World Scientific Publishing, 1997.