A Study of Symbol Segmentation Method for Handwritten Mathematical Formula Recognition using Mathematical Structure Information

Kenichi TOYOZUMI, Naoya YAMADA
Graduate School of Engineering, Nagoya Univ.
Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan
{ktoyo,nyamada}@suenaga.cse.nagoya-u.ac.jp

Takayuki KITASAKA, Kensaku MORI, Yasuhito SUENAGA
Graduate School of Information Science, Nagoya Univ.
Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan
{kitasaka,kensaku,suenaga}@is.nagoya-u.ac.jp

Kenji MASE
Information Technology Center, Nagoya Univ.
Furo-cho, Chikusa-ku, Nagoya, 464-8603 Japan
mase@itc.nagoya-u.ac.jp

Tomoichi TAKAHASHI
Faculty of Science and Technology, Meijo Univ.
1-501 Shiogamaguchi, Tempaku-ku, Nagoya, 468-8502 Japan
ttaka@ccmfs.meijo-u.ac.jp

Abstract

Symbol segmentation is very important in handwritten mathematical formula recognition, since it is the very first portion of the recognition process. This paper proposes a new symbol segmentation method using mathematical structure information. The base technique of symbol segmentation employed in the existing methods is dynamic programming which optimizes the overall results of individual symbol recognition. The new method we propose here improves symbol recognition performance by using correction values together with evaluation values of symbol recognition. These correction values are calculated from the relations among handwritten stroke positions and mathematical structure. There is no report which takes account of mathematical structure information for symbol segmentation in the handwritten mathematical formula recognition. Our experiments have proven that the recognition rate of symbol segmentation by existing methods is between 90.2% and 93.3%, while our proposed method gives correct recognition rate of 97.1%.

1. Introduction

Since on-line handwriting recognition combined with handwriting devices such as tablets is considered as the intuitive and efficient computer interface, a lot of topics have been studied in this domain. Handwriting input has not only user-friendly nature but also high expression potential. It is expected to provide efficient input methods for complex expressions, such as mathematical formulas and diagrams which include vertically placed symbols derived from particular structures in contrast with plain text where symbols are aligned just horizontally. For example, in the case of mathematical formula input to the computer, we often use markup languages or equation editors equipped with menu pallets for mathematical symbols and structures. However, they take much time compared with mathematical expression writing on papers with pens or pencils. Accordingly, many researches are conducted on on-line recognition of handwritten mathematical formulas [1].

The recognition process of handwritten mathematical formulas mainly consists of three phases: (a) symbol segmentation, (b) symbol recognition, and (c) mathematical structure recognition. In particular, the following natures have to be considered in the symbol segmentation phase; a symbol in mathematical formulas (1) consists of several strokes (usually up to four), (2) is displayed in various size (i.e. superscript and subscript) (3) is placed horizontally and occasionally vertically (i.e. fraction). Because of these natures, there are many burdensome stroke combinations which can be misinterpreted as a symbol in handwritten patterns of mathematical formulas.

Conventional methods of symbol segmentation for handwritten mathematical formulas often use the following clues: input order of strokes, positional stroke relation, and symbol recognition results [2, 3, 4, 5]. In particular, Winkler et al. [2] indicated that detailed stroke features could bring out accurate segmentation results. However, these methods barely have ability to deal with tricky situations where an incorrect stroke combination results in a symbol with strong confidence at an inferior-quality pattern affected by handwriting fluctuations.

0-7695-2128-2/04 $20.00 (C) 2004 IEEE

In this paper, we propose a novel symbol segmentation method for handwriting mathematical formulas using positional relation of strokes and mathematical structure information. Particularly, there exist no symbol segmentation approaches that take account of mathematical structure information. In Section **2**, symbol segmentation problems in mathematical formula recognition are summarized. They include a pilot study on an existing method. Section **3** presents the proposed processing procedure. Experimental results are shown in Section **4**. We add brief discussion in Section **5**.

2. Problems on Current Symbol Segmentation Technologies — Preliminary Experiments using Existing Methods —

2.1. Candidate Character Lattice Method

We start with the Candidate Character Lattice Method (CCLM) which segments each symbol so as to optimize the overall results of individual symbol recognition [6]. Candidate symbols in mathematical formulas are created by combining up to four sequential strokes. Then, a symbol recognition procedure is carried out for each candidate by consulting a symbol pattern dictionary. The CCLM evaluates each stroke combination, namely candidate symbol, with a distance between strokes. This distance is calculated based on a dynamic programming method using positional and directional features extracted from handwritten strokes. The normalized size (width or height of bounding boxes) of strokes is 128, which relates with magnitude of correction values mentioned below.

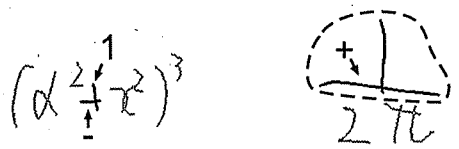
2.2. Baseline performance of symbol segmentation

To demonstrate the nature of segmentation problem in the recognition of mathematical formulas, a pilot experiment is conducted to measure the performance of the CCLM. Because of lack of handwritten mathematical formula database, we originally prepared a handwritten pattern set $\mathcal{S}_1$. $\mathcal{S}_1$ consists of 191 handwritten mathematical formulas including 3381 strokes and 2561 symbols, and as mathematical structures, fraction, root, right subscript and superscript, summation, production and matrix are appeared. The dictionary for symbol recognition is created by storing handwritten patterns collected beforehand in another phase of creating $\mathcal{S}_1$. This dictionary has 377 symbol patterns of 133 categories including alphanumerals, Greek letters, and a part of mathematical symbols. The symbol recognition rate of the CCLM we implemented is about 92% (2353/2561) if symbols are segmented correctly in advance.

Table 1(a) shows a result of symbol segmentation with the CCLM. Correct segmentation indicates the rate of correctly segmented symbols. The number of two types of segmentation errors are counted: *over segmentation* and *under segmentation* (Fig.1). In the first type of the segmentation error, one symbol is segmented as multiple symbols. In the latter case, multiple symbols are segmented as one symbol.

Table 1. Performance of CCLM.

	correct seg.	over seg.	under seg.
(a) $\alpha = 0$	90.2%	220	26
(b) $\alpha = 14$	93.3%	88	51
(c) $\alpha = 28$	90.8%	35	116



over segmentation (+) under segmentation ($\frac{1}{2\pi}$)

Figure 1. Examples of segmentation errors.

It is known that segmentation performance can be improved by adding a correction value α to the evaluation value of each symbol in segmentation methods that optimize the overall results of symbol recognition [7]. Let v_k as a symbol evaluation value in a sequence of segmented K symbols, an overall evaluation value V is written as

$$V = \sum_{k=1}^{K} (v_k + \alpha). \quad (1)$$

The segmentation results with correction value α are shown at Table 1(b) and (c). The results indicate that many *over segmentations* occur because a subset of stroke combinations of a symbol often have high likelihood as other symbols. The correction value indeed improves segmentation performance by reducing the number of *over segmentation*, but it also influences with the increase of *under segmentation*.

2.3. Desired Characteristics

The original CCLM hardly takes account of positional stroke relation and geometrical features of strokes such as sizes and shapes and so forth. Though these features seem to be very useful for symbol segmentation, they require the following considerations.

[Stroke combination rule] Some previous methods control combinations and separations among strokes by stroke combination rules [3, 4]. However the stroke identification process is needed before applying the rules, these approaches probably do not work for errors at the identification process of inferior-quality handwritings.

[Touching and intersection detection] The occurrence of touched strokes and intersection of bounding boxes of strokes does not necessarily result in stroke combinations under handwriting fluctuations. Especially, a symbol may be completely included in a bounding box of another symbol because of some structure such as subscript and root.

The clues mentioned above possibly bring out bad recognition results in the deterministic (hard-decision) approaches. The soft-decision frameworks are required if these stroke features are used [2, 7].

3. Proposed Method

To improve the performance of a symbol segmentation process in recognition of handwritten mathematical formulas, the following clues are available: positional relation of strokes, mathematical structure, and contextual information. This paper focuses on the positional relation of strokes and mathematical structure. The integration of contextual information will be presented in future work.

In the proposed method, the symbol segmentation process optimizes the evaluation values of symbol recognition corrected by β and γ instead of α. The term β is a correction value calculated by stroke neighbor relation, and γ is by mathematical structure information. The following sections explain these correction values in detail. The proposed method is one of the soft-decision approaches.

3.1. Stroke Neighbor Relation

The results of the pilot experiment (Table 1) shows a number of *over segmentation* arise only with the evaluation values of symbol recognition. First, we tackle the problem of *over segmentation* using positional relation of strokes. Neighbor distance d_n is adopted since it is thought of as one of the robust features against variation of size, shape and identification caused by handwriting fluctuations. Neighbor distance is defined by $d_n = \min_{i,j}\{e(p_i, p_j)\}$, where $p_i(i = 1, 2, .., I), p_j(1, 2, ..., J)$ are the coordinate point sequences of strokes, and a function $e(\cdot, \cdot)$ indicates Euclidean distance between two points.

The correction value β for symbol segmentation is calculated from the confidence value of stroke combinations based on the frequency distribution of d_n. For this purpose, the frequency distributions of two kind of neighbor distance are acquired for the pattern set $\mathcal{S}_1$. The first distance $d_n^{(in)}$ represents the distance to the most neighbor stroke belonging to the same symbol. On the other hand, the distance $d_n^{(out)}$ means the distance to the most neighbor stroke which does not combine together as a symbol. The rates of frequency about the two kind of distance are shown in Fig. 2.

In order to reduce the number of *over segmentation*, in other words, to increase the probability of combination of

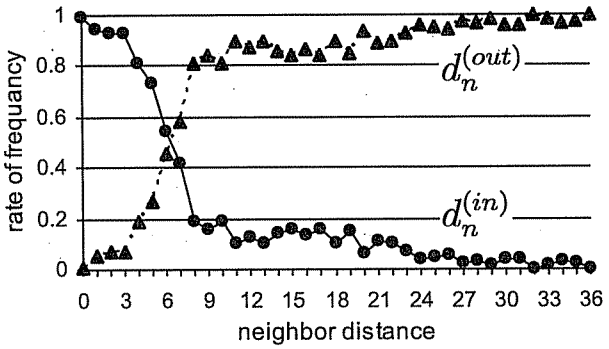


Figure 2. Rate of frequency.

proximal strokes, a correction value β is applied to a candidate symbol which has a proximal stroke. We suppose that a neighbor distance is converted into a confidence value of stroke combination from the frequency distributions. The distribution curve is approximated by the regression line of points within a certain range. Since the rate of frequency on $d_n^{(out)}$ is considered as the separability between some two strokes, the correction value β is calculated by multiplying this rate by a constant C_β.

3.2. Mathematical Structure

In inferior-quality patterns affected by handwriting fluctuations, incorrect stroke combinations sometimes have strong confidence of a symbol. This usually appears as *under segmentation*. It is impossible to handle such a case only with the clues derived from positional relation of strokes and symbol recognition. To cope with this problem, candidate symbols are corrected by γ which expresses mathematical structure information.

We first define the spatial grammars on structure symbols such as fraction line, $\sqrt{}$, Σ, Π. For instance, a fraction consists of a horizontal line and a pair of subexpressions (i.e. a numerator and a denominator) placed at the appropriate position above/below the line (Fig. 3). In the mathematical structure recognition phase, these grammars interpret handwritten patterns as mathematical structure when the particular spatial relation of symbols is detected. This also means corresponding structure symbols likely exist in handwritten mathematical formulas, which is fairly useful information for the symbol segmentation phase.

In order to calculate possibility of existence of structure symbols, we count up stroke patterns satisfying the each grammar in $\mathcal{S}_1$. The grammars regard a stroke as a part of subexpression if the gray boxes (Fig.3) include the center point of the bounding box of the stroke. The probability is defined as the rate of the case that the structure symbol re-

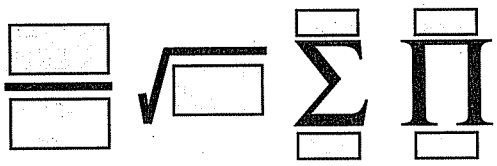


Figure 3. Spatial grammars of structure symbols. Gray boxes represent subexpressions.

Table 2. Results of symbol segmentation.

	C_β	C_γ	correct seg.	over seg.	under seg.
$\mathcal{S}_1$	0	0	90.2%	220	26
	31	0	96.2%	36	36
	31	51	97.1%	36	23
$\mathcal{S}_2$	0	0	88.7%	341	55
	31	0	94.3%	80	69
	31	51	95.3%	86	47

ally exists in $\mathcal{S}_1$. The grammar about fractions draws 0.948 as a probability of existence of horizontal line, for example. The correction value γ is calculated by multiplying this probability by a constant C_γ, and is added to candidate symbols which preclude existence of the structure symbols.

4. Experiment

The symbol segmentation performance of the proposed method is evaluated with $\mathcal{S}_1$ and another set of handwritten mathematical formulas $\mathcal{S}_2$, respectively. $\mathcal{S}_2$ consists of 200 handwritten mathematical formulas including 4803 strokes and 3579 symbols. The other conditions are same as the pilot experiment described in Section **2**.

Table 2 shows the results of symbol segmentation. The constants C_β and C_γ are determined so as to achieve the optimal performance for $\mathcal{S}_1$. We now learn each constant as an integer by the exhaustive search. The results show the correction values β and γ work reasonably. Particularly, it is important that the number of *under segmentation* is decreased by γ, namely mathematical structure information, with few increases of *over segmentations*.

5. Discussion

The results prove that mathematical structure information is useful for symbol segmentation, which is not verified until now. The remaining errors are mostly about vertically separated symbols such as 'i', '$=$', and '$\pm$'. Handling these symbols requires another clue like neighbor distance weighted by directions.

There may be symbol recognition procedures which can overcome some segmentation errors in this paper, but they also suffer from the patterns with strong confidence of a symbol incorrectly. The proposed method can work well for such situations. In particular, it essentially requires no restriction such as stroke input order, geometrical assumptions of strokes, and so on. The statistical nature of this method seems to be a good match with learning techniques.

6. Conclusion

We presented a symbol segmentation method for on-line recognition of handwritten mathematical formulas using positional relation of strokes and mathematical structure information. It is confirmed that mathematical structure information is useful for the symbol segmentation process. In future work, we plan to integrate mathematical structure information in a probabilistic method. An automatic learning of parameters will be also introduced.

Acknowledgments

This work was partly supported by the Grants-in-Aid for Scientific Research and the 21st Century COE Program "Intelligent Media (Speech and Images) Integration for Social Information Infrastructure" from the Ministry of Education, Culture, Sports, Science and Technology.

References

[1] K.F.Chan and D.Y.Yeung, "Mathematical expression recognition: A survey," International Journal on Document Analysis and Recognition, vol.3, no.1, pp.3-15, Aug. 2000.

[2] S.Lehmberg, H.J.Winkler, and M.Lang, "A Soft-Decision Approach for Symbol Segmentation within Handwritten Mathematical Expressions," Proc. of ICASSP, pp.3434-2437, May 1996.

[3] T.Kanahori, K.Tabata, W.Cong, F.Tamari and M.Suzuki, "On-Line Recognition of Mathematical Expressions Using Automatic Rewriting Method," ICMI2000, LNCS1948, Springer, pp.394-401, 2000.

[4] K.Toyozumi, T.Suzuki, K.Mori and Y.Suenaga, "A System for Real-Time Recognition of Handwritten Mathematical Formulas," Proc. of 6th ICDAR, pp.1059-1063, Sep. 2001.

[5] R.Zanibbi, D.Blostein and J.R.Cordy, "Recognizing mathematical expressions using tree transformation," IEEE Trans. PAMI, vol.24, no.11, pp.1455-1467, Nov. 2002.

[6] H.Murase, T.Wakahara, and M.Umeda, "Online Writing-Box Free Character String Recognition by Candidate Character Lattice Method," (in Japanese) Trans. of IEICE, Vol.J68-D, No.4, April 1985.

[7] S.Senda, M.Hamanaka and K.Yamada, "Box-free Online character Recognition Integrating Confidence Values of Segmentation, Recognition and Language Processing," (in Japanese) Tech. Rep. of IEICE, PRMU98-138, Dec. 1998.