MVA2002 IAPR Workshop on Machine Vision Applications, Dec. 11 - 13, 2002, Nara - ken New Public Hall, Nara, Japan

13—20

A Study on Correction Functions for On-Line Handwriting Entry of Mathematical Formulas

Naoya YAMADA
Graduate School of Eng., Nagoya Univ.,
Nagoya 464-8603, Japan
nyamada@suenaga.cse.nagoya-u.ac.jp

Kenichi TOYOZUMI
Graduate School of Eng., Nagoya Univ.,
Nagoya 464-8603, Japan
ktoyo@suenaga.cse.nagoya-u.ac.jp

Kensaku MORI
Faculty of Eng., Nagoya Univ.,
Nagoya 464-8603, Japan
mori@cse.nagoya-u.ac.jp

Yasuhito SUENAGA
Faculty of Eng., Nagoya Univ.,
Nagoya 464-8603, Japan
suenaga@cse.nagoya-u.ac.jp

Abstract

This paper describes functions for correcting mathematical formulas inputted using the handwritten mathematical formula entry system that we have proposed [8, 9]. This system encourages intuitive entries of mathematical formulas using a user-friendly handwriting interface. We have implemented two basic operations specialized for correcting formula structures. One of the operations is "move and resize" of selected characters. This is because the position and the size of mathematical symbols are quite important to recognizing the meanings of formulas. The other operation makes it possible to select a block consisting of several characters and symbols. These two operations enable more flexible and efficient correction of the structure of the mathematical formulas. The experimantal results showed that a user can easily and efficiently correct wrong formulas by using the proposed functions.

1. Introduction

Mathematical formulas are essential in technical writings to express concepts and theories concretely. Although most technical writings are prepared using word processing applications on computers, there are difficulties that arise when entering mathematical formulas into a computer, unlike the ease of entering sentences. This is because mathematical formulas have a special structure, such as characters and symbols of various sizes located on a two-dimensional plane.

Many applications to help the user enter mathematical formulas have been developed. Typesetting languages such as LaTeX uses tag commands to express the mathematical symbols and structures in a formula. By using a typesetting language, a mathematical formula is expressed as text, thus the user can enter formulas using a keyboard. But the user must learn many commands and they have difficulty recognizing the original formula directly from the entered text. Equation editor enables the user to enter a formula interactively, and they can easily understand an entered formula by using the WYSIWYG environment. But GUI menus must be used to enter mathematical symbols and templates of mathematical structures, thus the operation is quite complicated.

An easier way to enter formulas is the handwritten entry of mathematical formulas. The user can directly write mathematical formula with handwriting. Handwritten entries are more useful than equation editors because the user can express a formula intuitively using handwriting. To realize a handwritten entry, handwritten mathematical formula recognition has been studied by many researchers [1-9]. However, there is the potential for a user to input formulas incorrectly and for the system to incorrectly recognize inputs. These cases are inherently inevitable, thus the system should accommodate functions that enable the user to correct the recognized result easily.

The fundamental function of correcting an entered formula is the deletion of characters. Using this method, the user can delete incorrect entries, and corrections are performed by rewriting characters. This is similar to the natural method where the user erases a formula written on the paper and writes it again. However, this method is regarded as a time-consuming and burdensome task. Moreover, the formula-recognition errors are generally divided into two categories. One is errors of character recognition and the other is errors of structural recognition. Because the delete method corrects both the characters and structure of a for-

mula, and it must correct not only the incorrect parts but the correct parts as well, the method should be redundunt. Thus, the character errors and structure errors must be corrected independently to increase the efficiency of the correction process.

The method for correcting character errors in the current system is to select the correct character from the N list [7, 11], as seen in the general character recognition system. Meanwhile, the methods for correcting the formula structure are moving characters [7], changing the conjunctions of the characters [11], extending a fraction line [8], and inserting characters[10]. However, these methods are inadequate for altering an incorrect formula structure flexibly.

This paper describes functions for correcting incorrect entries on the mathematical formula recognition system [8, 9]. In the next section, we describe the system and the recognition methods. Section 3 explains the proposed method for correcting incorrect entries of a mathematical formula. Section 4 reports the results of experiments we conducted that verified the efficiency of the proposed method.

2. Recognition Method

This section presents the recognition scheme for the mathematical formula entry system. A mathematical formula is a set of characters located on a two-dimensional plane. The characters are aligned not only horizontally, there are vertical expansions around particular mathematical symbols, called parent symbols in this paper (Fig. 1). For example, a fraction line has a set of characters above or below itself. We call the subordinate characters a sub-expression of the parent symbol. Such a structure in a mathematical formula is called a two-dimentional structure in this paper. The two-dimensional structure appears hierarchically, for example, a fraction could contain another fraction in its denominator or numerator. Thus, the two-dimensional structure is expressed as a tree structure (Fig. 2).

The process flow of the system is shown in Fig. 3. The inputs to this system are strokes written by the user. In the Stroke Analysis, the system calculates the features of the input stroke. This features is used for Character Recognition, and a character of the best matching prototype is assigned to the input stroke.

The Structure Recognition function makes a tree structure as the recognition result of the formula written by the

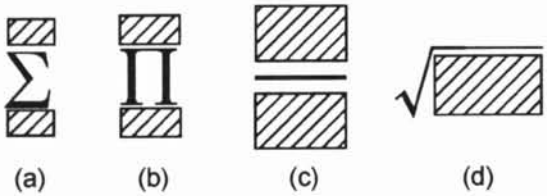

Fig. 1 Parent symbols and its input areas.
(a) summation (b) production
(c) fraction , under- or over- line (d) square root

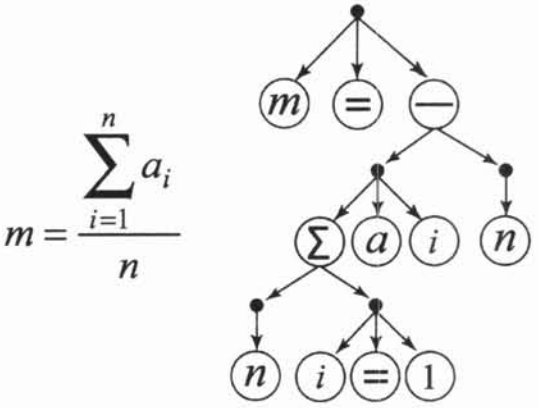

Fig. 2 An example of mathematical formula and its tree structure.

user. In the Structure Recognition step, the location of the input stroke in the tree structure is determined from the relationship of the positions of the input stroke and the characters existing in the tree structure. This process runs recursively because of the hierarchy of the mathematical formulas. When a stroke is written in the input area of a certain parent symbol, shown as a rectangle in Fig. 1, the recognition continues recursively at the sub-expression of the parent symbol. The recognition proceeds from the root, and descends to its child node. Resultingly, the inputted character is added to an appropriate location on the tree structure.

The Superscript and Subscript Recognition functions recognize the relation of two adjacent characters in the same sub-expression. The relation is classified into one of three categories, i.e., subscript, superscript, and same line, by

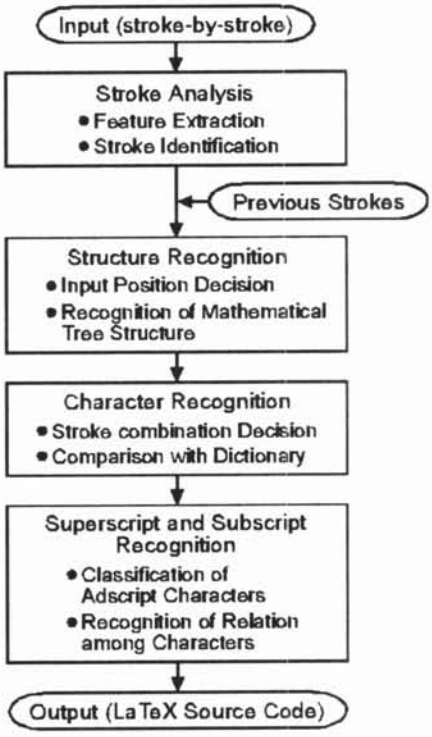

Fig. 3 Process flow.

thresholding the vertical positions of the characters.

The Structure, Superscript, and Subscript Recognition functions contribute to the recognition of the structure of a mathematical formula. This process runs in real-time, because the recognition consists of simple tasks based on the relations among the characters in a formula.

3. Correction Functions

The correction methods we propose move and resize the characters to alter their position and size in a formula. The position and size of the characters is quite important for recognizing formula strucutures. Recognition errors often arise when a characters is misaligned. In that case, the recognition errors are adequately corrected by altering the position and size of the characters responsible for the errors.

We also propose a novel method for selecting a block that consists of several characters or symbols. We call this selection mechanism the block selection method. We usually regard mathematical formulas as a set of hierarchical boxes in which multiple mathematical characters and symbols exist simultaneously. These boxes are called "blocks" in this paper. For example, consider the case of a user correcting the formula shown in Fig. 5. The user may want to correct the whole fraction all at once. Or, the user may correct just its numerator or denominator. Furthemore, there are cases where a user may have to correct a whole formula in one step. The block selection method can handle such tasks by preparing possible units of characters and symbols. By using the block selection method, the user can select an arbitrary block and make the characters contained in the block move, or they can resize or delete them.

The blocks are composed from the recognition result of the entered formula. A block consists of a set of characters in a corresponding subtree of the tree structure. Therefore, by covering all of the subtree in the tree, the system can thus obtain all blocks in the formula.

Figure 6 shows an example of the correction operation using the "move and resize" method and the block selection method. When the user begins correcting, bounding boxes for all blocks are displayed (a). The user can select an arbitrary block by indicating the block with the cursor. If the cursor enters a block, the block becomes highlightened (b). Thus, the user can intuitively understand which block is to be selected. With a block selected, the user can interactively move (c, d) or resize (e, f) the characters and symbols contained in the block by a drag and drop operation using the pen or mouse. Such an operation is analogous to the one in the GUI.

When the characters are moved or resized by the proposed methods, the strucuture of the altered formula is recognized instantly. The recognition is achieved in real-time in the system, thus the operation of the correction is quite interactive and the user can execute the correction at any moment.

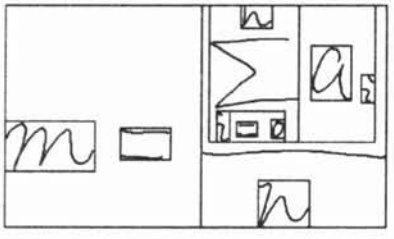

Fig. 5 Bounding boxes for all blocks.

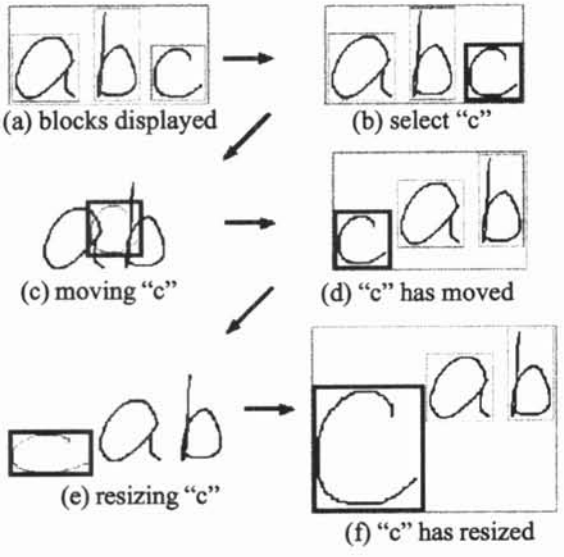

Fig. 6 An example of correction operation.

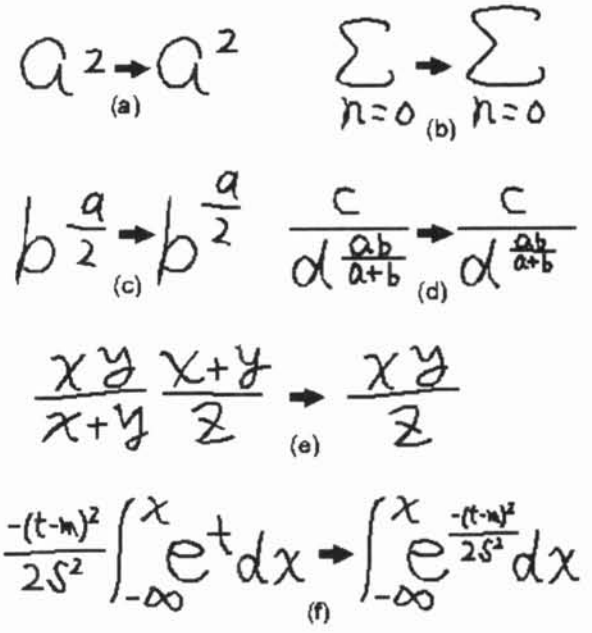

Fig. 7 Evaluated mathematical formulas.
(a) move "2" so that it is the superscript of "a" (b) resize "Σ" so that it includes "$n=0$" as a sub-expression (c) move "$a/2$" so that it is the superscript of "b" (d) resize "$ab/a+b$" so that it is the superscript of "d" (e) delete "$x+y$" and move "z" (f) delete "t" and move and resize "$-(t-m)^2/2s^2$"

Table 1. Experimental results.

Formula	Proposed	Previous	Proposed/Previous
(a)	0.59 sec.	1.62 sec.	36.6%
(b)	0.87 sec.	2.29 sec.	38.1%
(c)	0.77 sec.	4.72 sec.	16.2%
(d)	1.22 sec.	9.80 sec.	12.4%
(e)	3.44 sec.	4.21 sec.	81.7%
(f)	6.50 sec.	14.63 sec.	44.4%
Average	2.23 sec.	6.21 sec.	35.9%

4. Experimental Results

We evaluated the efficiency of the proposed method by comparing the operation time spent altering inputted formulas with our methods and the time required in the previous system. In the experiment, eleven subjects tried to correct the six formulas described in Fig. 7 with a tablet (WACOM PL-500) by using the "delete and rewrite" method and the proposed method, respectively. This system is embedded in a Java applet and runs in the Microsoft Internet Explorer environment. Table 1 shows the experimental results.

The experimental results show that the proposed method required less operation time than the previous method. Particularlly, when the user corrects multiple characters, such as in cases (c) and (d) in Fig. 7, the proposed methods produced significant results. This is because the "move and resize" method can reuse the previously entered characters, while the previous method cannot.

When multiple recognition errors occur simultaneously, the user may not be able to select the proper characters using the block selection method, because the the units of the characters or blocks are composed from the recognition result of the entered formula. In such a case, however, corrections are accomplished by selecting characters one at a time.

5. Conclusion

We proposed a correction function for mathematical formula structures and a novel selection method for a handwritten mathematical formula entry system. The "move and resize" method reuses the symbols entered previously, thus the burden of correcting formulas is reduced. The block selection method utilized the recognition result to compose units of symbols to be selected. This method is especially designed to correct mathematical formulas. The experimental results indicated that the operation time was reduced to 35.9% on the average and demonstrated that a user can easily correct wrong formulas by using these functions.

Acknowledgements

The authers thank the members of the Suenaga and Toriwaki laboratories for their useful suggestions and discussions.

References

[1] Kam Fai Chan and Dit Yan Yeung, "Mathematical Expression Recognition: A Survey," International Journal on Document Analysis and Recognition, 3(1):3-15 (August, 2000).

[2] Dorothea Blostein and Ann Grbavec, "Recognition of Mathematical Notation," In H. Bunke and P.S.P Wang, editors, Handbook of Character Recognition and Document Image Analysis, World Scientific Publishing, chapter 21, pp. 557-582, 1997.

[3] Andreas Kosmala and Gerhard Rigoll, "On-Line Handwritten Formula Recognition Using Statistical Methods," In Proc. Int. Conference on Pattern Recognition (ICPR), pp. 1306-1308 (August, 1998).

[4] Kam Fai Chan and Dit Yan Yeung, "PenCalc: A Novel Application of On-Line Mathematical Expression Recognition Technology," In Proc. Sixth Int. Conference on Document Analysis and Recognition (ICDAR), pp. 774-778 (September, 2001).

[5] Atsushi Murase, Takashi Satou, and Masaki Nakagawa, "Prototyping of METAH, A Recognition System for On-Line Handwritten Mathematical Expressions," Information Processing Society of Japan SIG Notes, **93-HI-48**, pp. 25-32 (May, 1993) (in Japanese).

[6] Ryoji Fukuda, Sou I, Fumikazu Tamari, Xie Ming, and Masakazu Suzuki, "A Technique of mathematical Expression Structure Analysis for the Handwritten Input System," In Proc. Fifth Int. Conference on Document Analysis and Recognition (ICDAR), pp. 131-134 (September, 1999).

[7] Steve Smithies, Kevin Novins, and James Arvo, "A Handwriting-Based Equation Editor," In Proc. Graphics Interface, pp. 84-91 (June, 1999).

[8] Kenichi Toyozumi, Takahiro Suzuki, Kensaku Mori, and Yasuhito Suenaga, "A System for Real-Time Recognition of Handwritten Mathematical Formulas," In Proc. Sixth Int. Conference on Document Analysis and Recognition (ICDAR), pp. 1059-1063 (Sepetember, 2001).

[9] Takahiro Suzuki, Shiro Aoshima, Kensaku Mori, and Yasuhito Suenaga, "A New System for the Real-time Recognition of Handwritten Mathematical Formulas," In Proc. 15th Int. Conference on Pattern Recognition (ICPR), 4, pp. 515-518 (September, 2000).

[10] Andreas Kosmala, Gerhard Rigoll and Anja Brakensiek, "On-Line Handwritten Formula Recognition with Integrated Correction Recognition and Execution," In Proc. Int. Conference on Pattern Recognition (ICPR), 2, pp. 590-593 (September, 2000).

[11] Takashi Satou, Katsuhiko Akiyama, and Masaki Nakagawa, "User Interface for Correction and Editing in METAH, A Recognition System for On-Line Handwritten Mathematical Formulas," Information Processing Society of Japan SIG Notes, **94-HI-54**, pp. 7-15 (May, 1993) (in Japanese) .