IJDAR (2016) 19:305–319
DOI 10.1007/s10032-016-0272-4

ORIGINAL PAPER

A system for recognizing online handwritten mathematical expressions by using improved structural analysis

Anh Duc Le[1] · Masaki Nakagawa[1]

Received: 28 May 2015 / Revised: 2 August 2016 / Accepted: 16 August 2016 / Published online: 26 August 2016

Abstract A system for recognizing online handwritten mathematical expressions (MEs), by applying improved structural analysis, is proposed and experimentally evaluated on two databases. With this system, MEs are represented in the form of stochastic context-free grammar (SCFG), and the Cocke–Younger–Kasami (CYK) algorithm is used to parse two-dimensional (2D) structures of online handwritten MEs and select the best interpretation in terms of the results of symbol segmentation and recognition as well as structural analysis. A concept of "body box" is proposed, and two SVM models are applied for learning and analyzing structural relations from training patterns without the need for any heuristic decisions. Stroke order is used to reduce the complexity of the parsing algorithm. Even though SCFG does not resolve ambiguities in some cases, the proposed system still gives users a list of candidates that contains the expected result. The results of experimental evaluations of the proposed system on the CROHME 2013 and CROHME 2014 databases and on an in-house ("Hand-Math") database show that the recognition rate of the proposed system is improved, while the processing time on a common CPU is kept to a practical level.

Keywords Online recognition · Handwritten mathematical expressions · Two-dimensional stochastic context-free grammar · CYK parsing algorithm

✉ Masaki Nakagawa
nakagawa@cc.tuat.ac.jp

Anh Duc Le
leducanh841988@gmail.com

[1] Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology, 2-24-16, Naka-cho, Koganei-shi, Tokyo 184-8588, Japan

1 Introduction

In recent years, devices such as pen-based or touch-based tablets, smartphones, and electronic white boards are becoming more and more popular. They allow users to annotate documents, draw figures, and write mathematical expressions, etc., more naturally and easily than traditional PCs with keyboard and mouse. New types of electronic pen and paper devices, such as the Anoto pen, for inputting handwriting are also available. Since 1960s and very actively in the last decade, researchers proposed many approaches for recognizing handwritten mathematical expressions (MEs).

Recognition of MEs can be divided into three major processes: segmentation of symbols, recognition of symbols, and analysis of structures. Firstly, a sequence of input strokes is segmented into hypothetical symbols (symbol segmentation), where each stroke is a sequence of coordinates from pen/touchdown to pen/touch-up. Secondly, each hypothetical symbol is then recognized by a symbol classifier (symbol recognition). Finally, structural relations among the recognized symbols are determined and the structure of ME is analyzed by parsing the recognized symbols to determine the most likely interpretation as a ME (structural analysis). Recognition of handwritten MEs is one of the current challenges concerning handwriting recognition. It requires several complex tasks, namely segmentation of symbols, recognition of symbols, analysis of the structure of ME, consideration of context, and evaluation of total likelihood. Some local ambiguities can be solved by using contexts (geometrical context, syntax context, or sublanguage), while others cannot be solved, even by using the contexts. In this work, the geometrical context is presented as structural relations among symbols and sub-expressions, while the syntax context and sublanguage are presented as SCFG. When ambiguities are not solved by the above-mentioned contexts,

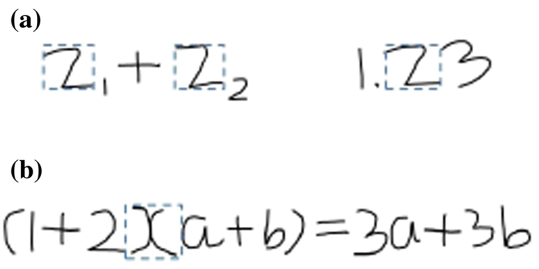


Fig. 1 Examples of local ambiguities

the semantics of MEs must be verified. This verification process is so complicated and needs much more computational time than recognition of ME process. Instead of using the semantics of MEs, giving users a list of candidates is proposed in this study as a simpler way. An example of local ambiguities that can be solved by the contexts is given in Fig. 1a. The first expression is "$z_1 + z_2$", while the second expression is "1.23". An example in which semantics is required to settle local ambiguities is shown in Fig. 1b. If the semantics is not used, this expression can be recognized as "$(1+2)(a+b) = 3a+3b$" or "$(1+2xa+b) = 3a+3b$". The context-based method proposed here ensures they both appear in the list of candidates.

The complexity of parsing context-free grammars was pointed out by Ruzzo [1]. A text recognition system considers only a linear structural relation from left to right, whereas a ME recognition system should consider many possible structural relations between symbols and sub-expressions in 2D. Determining structural relations between symbols and sub-expressions is not easy because mathematical symbols are written in various sizes, shapes, and positions, which all affect the structural relations.

As for the methods for inputting MEs, markup languages (such as LATEX) or editors (such as MathType) are commonly available and provide alternatives to handwriting recognition. For children to input MEs during e-learning and so on, however, these common methods are not straightforward to use. If they could write MEs in the same way as writing on paper, they would be able to use math e-learning or math learning software more easily and freely without being limited in terms of what they can input by mouse selection or key typing.

The rest of this paper is organized as follows. Related works on online recognition of handwritten mathematical expressions are reviewed in Sect. 2. The proposed system for recognizing online handwritten MEs is described in Sect. 3, where processes for segmentation and recognition of symbols, extraction of structural relations, and analysis of 2D structures are described in detail. The results of an experimental evaluation of the system are presented and discussed in Sect. 4. Conclusions and future works are discussed in Sect. 5.

2 Related works

Since the early works for recognizing printed MEs by Anderson (employing a top-down approach [2]) and Chang (employing a bottom-up approach [3]), many other approaches have been published. They are surveyed in two survey papers [4,5] in three processes of symbol segmentation, symbol recognition, and structural analysis. In the following, the three processes proposed during the last one or two decades are reviewed.

2.1 Symbol segmentation

For symbol segmentation, there are several approaches published. Symbol hypotheses net (SHN) was proposed by Lehmberg et al. [6]. To reduce nodes in SHN, stroke-specific features from a stroke and geometrical features from consecutive two strokes were employed. Prerecognition is used to avoid over-segmentation errors caused in symbols such as i, j, and $=$. A method based on the candidate character lattice was presented by Toyozumi et al. [7]. Over-segmentation and under-segmentation were reduced by stroke neighbor relations and mathematical structures, respectively. A symbol graph generation algorithm was proposed by Shi et al. [8], where symbol segmentation and recognition were considered at the same time. Recently, a method based on combination of geometrical features and multi-scale shape context features to classify successive strokes in time series was proposed by Hu et al. [9]. Three kinds of the shape context features (stroke pair, local neighborhood, and global shape contexts) were employed to get reliable information for segmentation.

2.2 Symbol recognition

For symbol recognition, there are three main approaches: online, offline, and combined approaches. For the online approach, several online methods have been used such as elastic matching [10], HMM [11], and RNN [12]. For the offline approach, AdaBoost, SVM, and random forest were employed by Davila et al. [13]. Offline features such as global features (angular change, line length, aspect ratio, and so on), crossing features, 2D fuzzy histogram of points, and fuzzy histograms of orientations were used. Synthesized patterns were generated to train the recognizer. For the combined approach, a combination of a feature matching and a hidden Markov model (HMM) was presented by Garain et al. [14]. Recently, a combination of online and offline methods with both using RNN was presented by Álvaro et al. [15].

2.3 Bounding box

Bounding boxes of symbols have been often employed to determine structural relations, but their problem mainly due to ascending and descending symbols has been recognized. To solve the problem for printed math expressions, the modified center of the bounding box was proposed by Okamoto et al. [16] for ascending and descending symbols. The normalized size and center of the bounding box were employed by Eto et al. [17] with ascending and descending parts of a symbol being extended to reduce the size variation. Then, the relative size and position of 2 symbols were used to determine the structural relations. The similar method was also employed by Aly et al. [18] for distinguishing the structural relation between adjacent pair. Each symbol was added virtual ascender and descender into its bounding box depending on its type such as ascending and descending normal symbols. The type of a symbol was determined by its size and position. For handwritten symbols, however, these methods may have difficulty with determining the type because of the large variation of the size and the position. A method to determine the structural relation between handwritten symbols/sub-MEs was presented by Álvaro and et al. [19]. Geometric features (including the normalized center) and shape context features were used for classifying the structural relations. The proposed body box in this paper is the trainable box for each type of a symbol.

2.4 Structural analysis

For structural analysis, there are many approaches. Several kinds of structures, like superscripts, subscripts, limits, and square roots, were detected using the baseline information of each symbol, and a bottom-up approach was used to merge two sub-expressions into a larger expression until the final expression was constructed by Garain et al. [14]. A method based on a tree transformation was proposed by Zanibbi et al. [20]. Input symbols are processed in three passes. A baseline structure tree describing the 2D arrangement of input symbols was constructed in the first pass. Tokens comprised of multiple input symbols such as decimal numbers and function names were grouped in the second pass. Expression syntax was analyzed, and an operator tree suitable for computation was produced in the last pass.

A layered search framework for recognizing online handwritten MEs was proposed by Rhee et al. [21]. A set of heuristic predictions was defined to estimate the cost of symbol recognition and structural relation for input strokes. Although nodes of undesirable interpretations are pruned, the computation time is still more than 15 s per expression (although information on CPU employed was not available).

The ME recognition problem was formulated as a search problem of the most likely ME candidate in a framework of stochastic context-free grammar (SCFG) by Yamamoto et al. [22]. In this approach, stroke order was employed to reduce the search space and the CYK algorithm was used to parse a sequence of input strokes. It was experimentally shown that the recognition result is improved in terms of different grammatical constraint levels. A similar approach was presented by Simistira et al. to parse segmented symbols with probabilistic SVMs and SCFG [23]. The CYK table and the CYK algorithm were visualized in detail.

A method for recognizing MEs by a top-down parsing algorithm was proposed by MacLean et al. [24]. A shared parse forest presenting all recognizable parses of an input was constructed by an incremental parsing method. Then, the top to *n*th most highly ranked trees were extracted from the forest. By using offline information (horizontal and vertical orders), the method was independent from stroke order with infeasible partitions reduced. This system incorporates a correction mechanism to help users to edit recognition errors. An improved version of this system based on the Bayesian model was presented [25].

A formal model for online handwritten ME recognition based on 2D-SCFGs and HMM was proposed by Álvaro et al. [26]. The CYK table was modified into one index to parse an input ME in 2D. For combining 2 sub-partitions, the structural relation between 2 sub-partitions was used instead of stroke order, since their system was independent from stroke order. However, the complexity of parsing was increased. For optimizing the complexity, the range search was employed. To determine structural relations among symbols and sub-expressions, a support vector machine (SVM) learns geometric features among bounding boxes. A global approach allowing mathematical symbols and structural relations to be learned directly from expressions was proposed by Awal et al. [27]. During the training phase, symbol hypotheses are generated without using a language model. A dynamic programming algorithm found the best segmentation and recognition of the input. A classifier learns both the correct and incorrect segmentations. The training process is repeated to update the classifier until the classifier recognizes the training set of MEs correctly. Gaussian models are used for modeling structural relations. The differences between baseline positions and heights of two bounding boxes are used for training the Gaussian models.

It is difficult to directly compare these methods, however, because they are often evaluated on in-house databases. Therefore, the competition on recognition of online handwritten mathematical expressions (CROHME) for evaluation of recognition systems based on common databases has been organized by Mouchère et al. Taking part in this competition, the last three works [24,26,27] gave the most promising results. The system by Álvaro et al. won the best system at CROHME 2011 and got the best system trained on the CROHME dataset at CROHME 2013 and 2014. The system by MacLean et al. and that by Awal et al. received the second and third prizes at CROHME 2012, respectively. The system developed by MyScript won the best system at CROHME

2012, 2013, and 2014. It was trained on their private database [28].

2.5 Application

We are now working on self-learning systems to help elementary, junior high, and high school students learn math. The advantages of pen-based handwriting input such as being faster, more efficient, and more preferable for entering mathematics on computers are shown in Anthony et al. [30,31]. Students write MEs for answering questions. For this purpose, an automatic method for recognizing handwritten MEs must be developed. The systems, which have joined in CROHME, show high recognition rates; however, employing handwritten ME recognition in a real environment, namely education, still faces significant problems. For instance, recognition speed and memory size are practically important. Moreover, if the correct answer is present within several candidates, even if the top candidate is not correct, a user can select it without rewriting MEs. Thus, the cumulative recognition rate is an important parameter for evaluating a recognition algorithm.

2.6 Complexity

As for the self-learning system being developed, the number of partitions from n input strokes in 2D is defined by Bell number as $B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$ [27]. It is unnecessary to parse all the partitions because there are many infeasible partitions. A method for reducing infeasible partitions (by using horizontal and vertical order) was proposed by Maclean et al. With this method, however, the worst-case number of subpartitions that must be considered during parsing is still quite large as n^4 [24]. Since the order to write symbols (symbol order) is not completely free and people write MEs in common orders, variations in symbol order can be handled by extending the grammar to deal with multiple common symbol orders. Like the complexity of the original CYK parsing algorithm, that of the modified CYK parsing algorithm is $O(n^3|P|)$, where n is the length of an input string and $|P|$ is the number of production rules in the grammar. In addition to the complexity, the computation time of the proposed system on a common CPU is feasibly short (and large waiting times are not incurred).

As for the proposed recognition system, local ambiguities are solved through applying an SCFG. The recognition system simultaneously optimizes symbol segmentation, recognition, and structural analysis. The contributions of the present study are twofold. First, stroke order is employed to reduce the number of symbol hypotheses to $nk - k(k-1)/2$ (k is the maximum number of strokes per symbol), limit the possible sub-expressions to subsequences (described later in Sect. 3.5), and achieve less time complexity in the parsing process than that involved in the two methods described above [24,26]. The grammar is extended to cope with common variations in stroke order. Second, a body box (rather than a bounding box) is proposed, and two SVM models are used to determine structural relations and improve the recognition rate. The recognition system is experimentally evaluated on the CROHME 2013 and 2014 databases [28,29] and on our Hand-Math database [32]. The evaluation results show that the proposed system improves recognition rate while keeping processing time on a common CPU to a practical level.

3 System organization

The handwritten ME recognition problem is formulated as a search problem of the most likely interpretation of handwritten strokes. The search problem is modeled as the following formula:

$$\begin{aligned} C = &\sum ln(P_{\text{seg}}(S_j, S_{j+1})) + \sum ln(P_{\text{rec}}(S_i|G_i)) \\ &+ \sum ln(P_{\text{rel}}(R_k|SE_k)) + \sum ln(P_{\text{Gram}}) \end{aligned} \quad (1)$$

where $P_{\text{seg}}(S_j, S_{j+1})$ is the probability of separation between stroke j and $j+1$, $P_{\text{rec}}(S_i|G_i)$ stands for the probability that a group of strokes (G_i) is recognized as symbol S_i, $P_{\text{rel}}(R_k|SE_k)$ is the probability that two sub-expressions are combined into a larger expression in relation R_k, and P_{Gram} is the probability of production in the grammar. The detail of these probabilities is described as follows.

3.1 Symbol segmentation

As for the proposed system, the separation probability of each pair of adjacent strokes is computed, and all symbol hypotheses of a ME are generated in the segmentation process. The separation probability is calculated from twelve geometric features (shown in [6,7]) and nine additional features. Consequently, we extracted 21 geometric features, such as the minimum distance of adjacent strokes over the average height of strokes in the ME (Avg_h), length of off-stroke over Avg_h, overlap area projected onto the x- and y-axis over Avg_h, overlap between two bounding boxes over the area of the bounding box for the first stroke, and distance between centers over Avg_h. An SVM classifier is used for segmentation. From the distribution of the output score of the SVM classifier, threshold T could be set so that a pair of strokes is segmented if its score (t) is higher than T and not segmented if the score is lower. However, this hard decision does not allow the later processes to recover from error decisions. The distribution of the SVM output scores for training patterns is shown in Fig. 2. Segmentation th$_s$ and

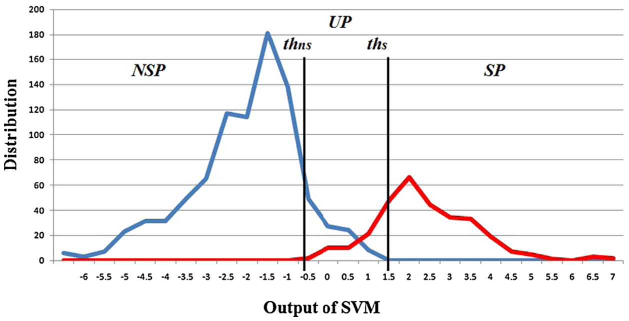


Fig. 2 Distribution of SVM output scores

non-segmentation threshold th$_{ns}$ are set so as to divide the output score to three areas: segmentation points (SP), non-segmentation points (NSP), and undecided points (UP). The separation probabilities in *SP* and *NSP* are 1 and 0, respectively. In the UP area, a Sigmoid function is used to transform the SVM score to probability.

3.2 Symbol recognition

After symbol hypotheses are generated, each symbol hypothesis is recognized by a character recognizer that combines offline and online recognition methods [33]. This combination is robust against stroke connections and cursive strokes due to elastic matching by the online method and robust against stroke disorders or duplicated strokes due to image matching by the offline method.

The original recognizer for recognizing Japanese characters is constructed from online and offline recognizers [33]. The online recognizer employs Markov random field models, while the offline recognizer employs modified quadratic discriminant functions. Both of the recognizers employ a coarse classifier to reduce computation time on over 8000 categories of Japanese characters. For ME recognition, however, the original recognizer for Japanese characters is modified. First, since compression of its dictionaries and its coarse classifier, which are employed to reduce memory size and computation time, are not necessary for a small set of symbols used in MEs (i.e., 101 classes), they are removed. The resultant memory size for handwritten ME recognition is 1.38 MB for 101 categories. Second, since the true candidate often appears within the top five candidates (although more candidates have been kept for the later stages of Japanese text recognition), only five candidates for each symbol pattern are kept. Third, an auxiliary process for recognizing period, comma, and prime is introduced. These symbols are treated uniformly here and discriminated in the later processes by an SCFG that will be defined in the next subsection.

For each symbol recognized as a period, comma, or prime, a size penalty based on its height and width is added. The size penalty P is calculated as follows:

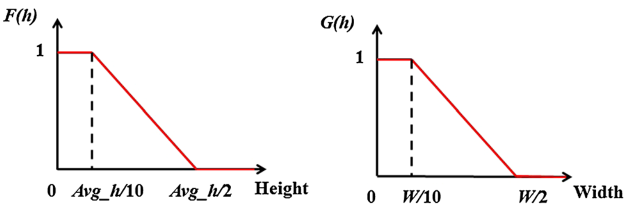


Fig. 3 Membership function for period/comma/prime

$$P_{\text{period/comma/prime}} = F(\text{height}) * G(\text{width}) \tag{2}$$

where F and G are fuzzy functions shown in Fig. 3. If the height of the bounding box is less than Avg_$h/10$, $F = 1$; otherwise, F decreases linearly to zero at Avg_$h/2$. Function G is defined in a similar manner in regard to the width of the bounding box.

On the other hand, the likelihoods of other candidates are derived from symbol recognition scores. The reasons of adding the size penalty are as follows:

- Distinction among period, comma, and prime is difficult and somehow meaningless since they can be distinguished by using the grammar.
- These symbols are often misrecognized as other symbols when they are normalized as a standard size. The size penalty reorders the list of candidates to decease misrecognition.

3.3 Structural analysis

All local ambiguities in the previous phases are solved in the structural analysis. An SCFG is defined formally by a five-tuple $G = (N, \Sigma, R, P, S)$, where

- N is a finite set of non-terminal symbols.
- Σ is a finite set of terminal symbols.
- R is a finite set of relations between symbols and subexpressions, i.e., *horizontal*, *over*, *under*, *superscript*, *subscript*, and *inside*.
- P is a finite set of productions of one of the following forms:
  - $(X \to \alpha, p)$
  - $(X \to A, p)$
  - $(X \overset{r}{\to} AB, p)$, with $X, A, B \in N, \alpha \in \Sigma, r \in R$. Each production rule is associated with a probability p and these probabilities satisfy the condition: $\forall A \in N : \sum_{i=1}^{n_A} p(A \to \alpha_i) = 1$, where n_A is the number of rules associated with non-terminal symbol A.
- $S \in N$ is a distinguished start symbol.

(a)

$$SupExp \overset{Super}{\rightarrow} Symbol\ Symbol$$
$$SupExp \overset{Super}{\rightarrow} Symbol\ Number$$
$$SupExp \overset{Super}{\rightarrow} Number\ Symbol$$
$$SupExp \overset{Super}{\rightarrow} Number\ Number$$

(b)

$$Term \rightarrow Symbol$$
$$Term \rightarrow Number$$
$$SupExp \overset{Super}{\rightarrow} Term\ Term$$

Fig. 4 Grammars for superscript expressions written in CNF and with added unary rules. **a** Grammar in CNF. **b** Grammar in CNF with adding unary rules

The stochastic estimation of the rule probabilities could be performed with the inside–outside algorithm [34]. Álvaro et al. presented a modified version of the inside–outside algorithm for estimating 2D SCFGs [26]. In this work, we simplify the estimation of the rule probabilities by assuming that each rule probability associated with the non-terminal symbol A is equal. As the result, it is defined as : $p(A \rightarrow \alpha_i) = 1/n_A$.

Although handwritten strokes may be ambiguous, the modified SCFG used in the proposed system is defined to be unambiguous. This definition implies that every valid expression generated from the grammar has a unique leftmost derivation. It is hard to define an unambiguous grammar in a Chomsky normal form (CNF), so one more type of production rules ($X \rightarrow A$) (unary rule) is added to make it easier to define the grammar. The production rules ($X \rightarrow A$) provide an effective way to define an unambiguous grammar. Figure 4 shows grammars for superscript expressions that are written in CNF and that are with the added unary rules. Owing to them, we do not need to repeat rules for combination of non-terminal rules. As the result, the number of rules is reduced.

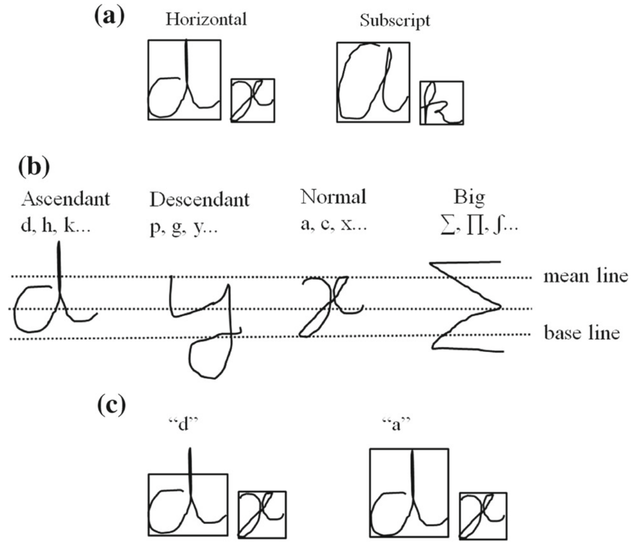


Fig. 5 Bounding boxes and body boxes for mathematical symbols. **a** Bounding boxes. **b** Four groups of symbols (ascendant, descendant, normal, and big)

Table 1 Body box for a mathematical symbol depending on its type

Group	Derivation of body box from bounding box
Ascendant	Top Body = Top − r_1* Height
	Bottom Body = Bottom
Descendant	Top Body = Top
	Bottom Body = Bottom + r_1 * Height
Normal	Top Body = Top
	Bottom Body = Bottom
Big	Top Body = Top − r_2 * Height
	Bottom Body = Bottom + r_2 * Height

3.4 Structural relations

The structural relations expressed in MEs are sometimes ambiguous even for humans. For example, Fig. 5a shows the case that bounding boxes are similar, but relations are different. Moreover, handwritten symbols are written in various shapes and sizes. To estimate structural relations, bounding boxes have often been employed, but problems with them have been pointed out [22]. We follow the method of Hidden Written Area, which was proposed to represent logical relations more stably [22]. However, it must be obtained statistically and it is often blurred by unstable ascendant and descendant strokes. Therefore, we propose the concept of body box for each symbol and expression. First, symbols are classified into four groups: ascendant, descendant, normal, and big. Ascendant or descendant symbols extend above a mean line or below a base line. Normal symbols fall between the mean and base lines. Big symbols extend beyond both the mean and base lines, as shown in Fig. 5b. For each group, it is assumed that a body box just covers the main body of each symbol. It is defined by the predefined proportion of the bounding box. The left and right of the body box are same as the bounding box, whereas the top (Top Body) and bottom (Bottom Body) of the body box are calculated by the function of r_1 and r_2 as shown in Table 1. Here, Top, Bottom, and Height are the top, bottom, and height of the bounding box, respectively. The parameters r_1 and r_2 are trainable. Therefore, the body boxes of different symbol recognition candidates differ. Those of recognition candidates "d" and "a" are shown in Fig. 5c.

The body box is calculated not only for terminal symbols but also for non-terminal symbols. The body box of a non-terminal symbol is calculated as shown in Fig. 6. For each non-terminal symbol X in a production rule ($X \overset{r}{\rightarrow} AB$), the body box of X is calculated from the body boxes of A, B, and the structural relation between A and B. The rules presented in Fig. 6 are applied to calculate the body box. For the relations horizontal, over, under, or inside, the body box of X is the bounding box of the body boxes of A and

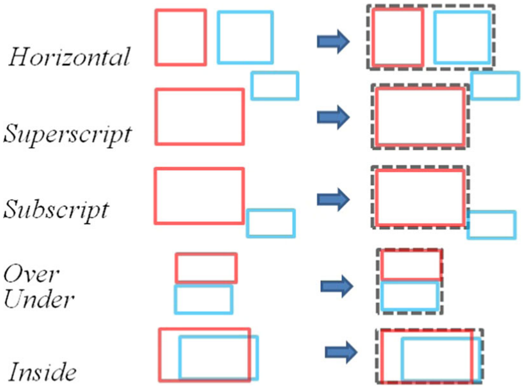


Fig. 6 Body boxes for mathematical expressions

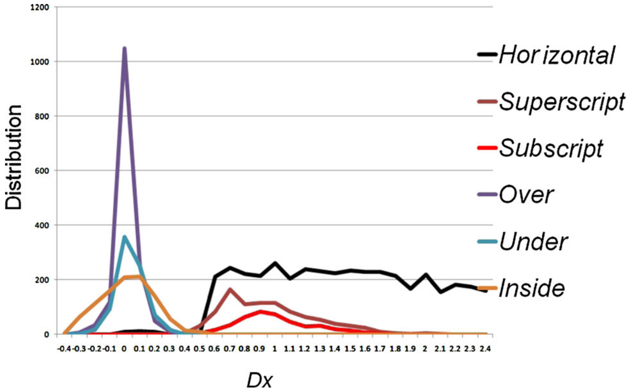
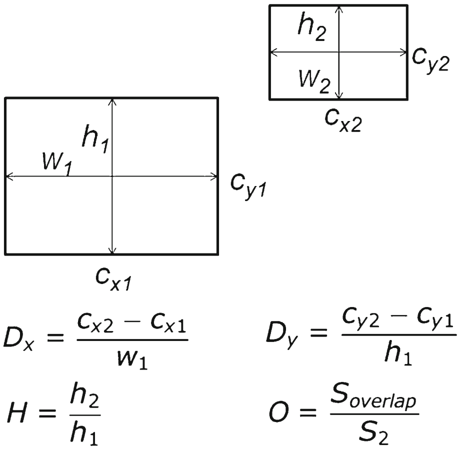


Fig. 7 Distribution of feature Dx

B. For the relations superscript or subscript, the body box is the body box of A. For each non-terminal symbol X in a production rule ($X \rightarrow A$), the body box of X is the same as the body box of A.

To determine the structural relations between symbols and sub-expressions, several approaches have been proposed [24,26,27]. A method based on fuzzy logic was proposed by MacLean et al. The membership function is defined in terms of the distance, angle, and amount of overlap between bounding boxes. The approach taken by Awal et al. uses a Gaussian model for calculating the probability of structural relations. This model is based on the differences between baseline positions and heights of two bounding boxes. However, both approaches must define several thresholds to identify structural relations. Another approach proposed by Álvaro et al. learns structural relations from training patterns without heuristic decision. This approach uses nine geometric features to classify relations. Distinguishing some of the relations does not need all nine features; however, the approach taken here prunes redundant features and adds the new essential feature. We reuse the features Dx, Dy, and H in Álvaro et al.'s paper and introduce the feature O (the overlap between 2 body boxes). The distribution of feature Dx (relation between horizontal centers of two body boxes for symbols and sub-expressions) is shown in Fig. 7. It is clear from the figure that the over, under, and inside relations are distinguished clearly with the horizontal, superscript, and subscript relations. Therefore, two steps with SVMs are introduced here. First, a threshold for the feature Dx (th_{Dx}) is used to divide relations into two groups: group 1 (over, under, and inside) and group 2 (horizontal, superscript, and subscript). Then, two SVM models to get the probability of certain structural relations within each group are built. The feature H represents the relation between the sizes of two body boxes. The feature Dy represents the difference between vertical centers of two body boxes. The feature O represents the overlap between 2 body boxes (S_{overlap}) which is normalized by area of the second body box (S_2). Features H and Dy are used for training the SVM of group 1, and features H, Dy, and O are used for training the SVM of group 2. All features are defined in Fig. 8.



$$D_x = \frac{c_{x2} - c_{x1}}{w_1} \qquad D_y = \frac{c_{y2} - c_{y1}}{h_1}$$

$$H = \frac{h_2}{h_1} \qquad O = \frac{S_{overlap}}{S_2}$$

Fig. 8 Features used for determining structural relations

3.5 CYK algorithm and its implementation

By employing stroke order, it is only necessary to consider subsequences of the time-ordered sequence of strokes. Thus, $n(n+1)/2$ subsequences exist in total. The number of allowable subsequences is equivalent to the number of cells used in the CYK table (Fig. 9). First, the proposed system is configured by a definition of SCFG. Then, the configured system invokes the CYK algorithm to produce a list of candidates for each subsequence in the input sequence of strokes: $s = s_1, s_2, \ldots s_n$. The algorithm has the following two stages:

Initial stage The maximum number of strokes per symbol is set as five (the largest number 4 for math symbols plus 1), in consideration of an additional or separated stroke, so that the cells in the CYK table may be initialized up to the fifth level, as shown in Algorithm 1. From the SVM output score, subsequences of strokes called symbol hypotheses are generated. Invalid hypotheses are discarded according to the following two constraints: (1) delete hypotheses containing SP or hypotheses having NSP between them, (2) reject hypotheses giving excessively low recognition probability.

Algorithm 1 Initializing CYK table.

```
for $j = 1, 5$ do
  for $i = 1, n - j$ do
    $t$ = group of strokes $s_i, ...s_{i+j}$
    if $t$ does not satisfy constraints of rejecting invalid hypotheses then
      for each production $(X \to \alpha)$ do
        if $P_{reco}(\alpha|t) > 0$ then
          Add node $((X \to \alpha), ln(P_{reco}(\alpha|t)) + ln(P_{seg}) + ln(P_{gram}(x \to \alpha)), t)$ into cell $(i, j)$
```

Parsing stage The CYK algorithm operates only on SCFG given in CNF. To the SCFG modified grammar, however, extra production rules $(X \to A)$ have been added, so the original CYK algorithm has to be modified as shown in Algorithm 2. Accordingly, $(X \xrightarrow{r} AB)$ production rules are used to reduce two sub-expressions to a non-terminal. Then, from the reduced non-terminal, $(X \to A)$ production rules are used to reduce it further to another non-terminal. In each cell in the CYK table, we store an array of nodes. Its size equals to the set size of non-terminal symbols. Each node is a representative of a non-terminal symbol, so we can access a node from the cell indexes (i_1, i_2) and the non-terminal symbol in $O(1)$. In each node, five best candidates for its non-terminal symbol are stored. LP(C, NT) is a variable to store the log-probability of non-terminal NT in Cell C. The candidates for the final result are extracted from cell $(1, N)$.

Algorithm 2 Parsing an input sequence of strokes.

```
for $i = 2, n$ do
  for $j = 1, n - i + 1$ do
    for $k = 1, i$ do
      for each production $(X \xrightarrow{r} AB)$ do
        $C_1$ = get node of non-terminal symbol A in Cell$(k, j)$
        $C_2$ = get node of non-terminal symbol B in Cell$(j - k - 1, j + k + 1)$
        $LP = LP(C_1, A) + LP(C_2, B) + ln(P_{rel}(C_1, C_2|r)) + ln(P_{seg}(s_{j+k}, s_{j+k+1})) + ln(P_{gram}(X \xrightarrow{r} AB))$
        if $LP > 0$ then
          Add node $((X \xrightarrow{r} AB), LP, C_1, C_2)$ into cell $(i, j)$
      for each production $(X \to A)$ do
        if cell$(i, j)$ has node $A$ then
          Add node $((X \to A), LP(Cell(i, j), A)$ into cell $(i, j)$
```

The complexity of the modified CYK algorithm is still $O(n^3|P|)$, like that of the original CYK algorithm, whereas the complexity of the algorithm proposed by Álvaro et al. [26] and that of the algorithm proposed by MacLean et al. [24] are $O(n^3 log(n)|P|)$ and $O(n^4|P|)$, respectively, where n is the number of input strokes for a ME and $|P|$ is the number of production rules in the grammar. Figure 9 shows an example of parsing for the ME $\frac{x^2}{2x}$. The CYK table is a triangular table where cell(i, j) contains the parsed candidates for the subsequence of length i starting from j. In each cell, we show only the best candidate which contains the non-terminal, the result, and the probability.

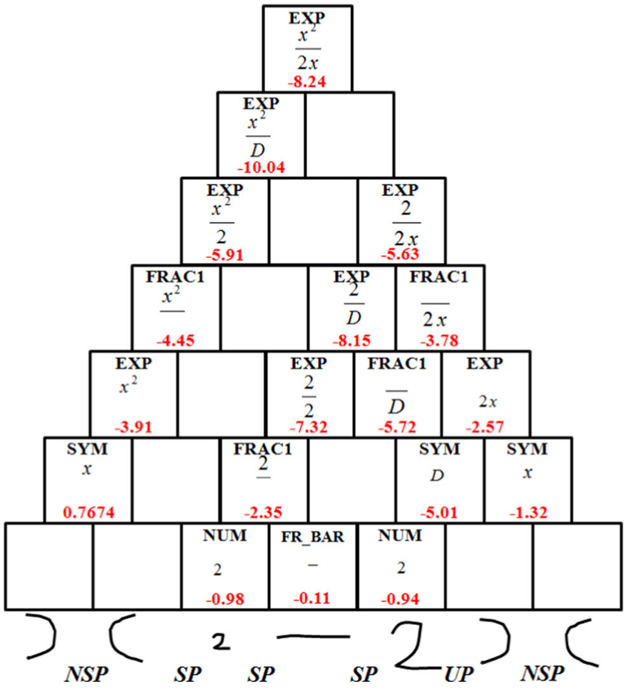


Fig. 9 Result of CYK table for the expression $\frac{x^2}{2x}$

3.6 Coping with common variations in symbol order

The method for handling common variations in symbol order is described as follows.

As for this method, first, mathematical structures that are often input with multiple symbol orders are identified. Then, for each structure, symbol order variations that users may use are enumerated. The proposed recognition system can handle structures such as fractional expressions, expressions containing periods or commas, bracket expressions, root expressions, integral expressions, summation expressions, and prod expressions. Finally, new production rules for handling each symbol order are added.

Six common symbol orders for fractional expressions, namely (1) fraction bar, numerator, and denominator; (2) fraction bar, denominator, and numerator; (3) numerator, fraction bar, and denominator; (4) denominator, fraction bar, and numerator; (5) numerator, denominator, and fraction bar; and (6) denominator, numerator, and fraction bar, are listed in Table 2. In the "production rules" column, the rule $(X \xrightarrow{relation} AB)$ denotes B relation A.

Table 2 Productions for handling multiple symbol orders for fractional MEs

Symbol order	Production rules
(1) Fraction bar, numerator, denominator	(1) Frac$_1 \overset{\text{over}}{\rightarrow}$ fraction bar, numerator (2) Frac $\overset{\text{under}}{\rightarrow}$ Frac1, denominator
(2) Fraction bar, denominator, numerator	(3) Frac$_1 \overset{\text{under}}{\rightarrow}$ fraction bar, denominator (4) Frac $\overset{\text{over}}{\rightarrow}$ Frac1, numerator
(3) Numerator, fraction bar, denominator	(5) Frac$_1 \overset{\text{under}}{\rightarrow}$ numerator, fraction bar (6) Frac $\overset{\text{under}}{\rightarrow}$ Frac1, denominator
(4) Denominator, fraction bar, numerator	(7) Frac$_1 \overset{\text{over}}{\rightarrow}$ denominator, fraction bar (8) Frac $\overset{\text{over}}{\rightarrow}$ Frac1, numerator
(5) numerator, denominator, fraction bar	(9) Frac $_1 \overset{\text{under}}{\rightarrow}$ numerator, denominator (10) Frac $\overset{\text{middle}}{\rightarrow}$ Frac1, fraction bar
(6) Denominator, numerator, fraction bar	(11) Frac $_1 \overset{\text{over}}{\rightarrow}$ denominator, numerator (12) Frac $\overset{\text{middle}}{\rightarrow}$ Frac1, fraction bar

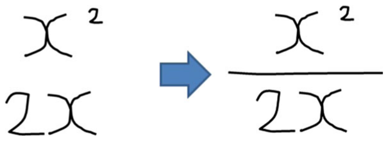


Fig. 10 An example of a stroke order for middle relation

As for symbol orders 5 and 6, a fraction bar is written after the numerator and denominator as shown in Fig. 10. In their productions, the following middle relation, which is a joint occurrence of over and under relations, as shown in Eq. (3), is used:

$$P_{\text{middle}}(\text{Frac}_1, \text{frac bar}) = P_{\text{over}}(\text{frac bar, numerator}) \\ * P_{\text{under}}(\text{frac bar, denominator}) \qquad (3)$$

For unexpected stroke orders such as delayed strokes, however, a more complicated method (such as sorting strokes or modifying the parsing algorithm) may be needed. As a result, the complexity might become too large for an ordinary CPU to handle.

4 Evaluations

The proposed ME recognition system was experimentally evaluated as follows. First, each component of the system (i.e., symbol segmentation, symbol recognition, and structural analysis) was tested individually. Then, the whole system was evaluated.

4.1 Databases

The experimental evaluations used the CROHME 2013 and 2014 databases [28,29] and our Hands-Math database [32].

Organized at ICDAR 2013, CROHME 2013 is a contest in which algorithms for online handwritten ME recognition compete [28]. It allows the performance of the proposed system to be compared with others under the same conditions. Selected from five different MEs databases, the CROHME 2013 database contains 8,836 MEs for training and 671 MEs for testing. The number of symbol classes is 101, including many similar symbols such as ($\{1, |, l\}$, $\{P, p\}$, $\{S, s\}$, $\{C, c\}$, $\{X, x, \times\}$, $\{V, v\}$, and $\{o, 0\}$. The proposed system participated in CROHME 2013 with eight other systems.

CROHME 2014 was organized at ICHFR 2014 [29]. The training set was kept as the same of CROHME 2013, while the testing set was created newly. The number of MEs in the testing set is 986 MEs. The organizer provided 2 more tasks of isolated symbol recognition and matrix recognition.

Hands-Math is a database of MEs collected from 62 primary school children, 27 junior high school students, and 26 students in the authors laboratory for the purpose of making a recognizer for MEs used in elementary and secondary schools [32]. The total number of MEs is 11,069. The number of symbol classes is 94, including math symbols used in Japanese elementary and secondary schools such as numerals, operators, uppercase and lowercase letters, and unit symbols. The database is divided into training and testing sets for training and evaluating the proposed system. The numbers of MEs in the training and testing sets are 8266 and 2803, respectively.

For training systems, we extract a validation set by taking 10 % of the training set and use the remaining 90 % for training. The validation set is used for estimating parameters for classification methods in segmentation of symbols, recognition of symbols, and classification of structural relations.

The grammars for recognizing MEs were designed based on the published grammar of CROHME and the grammar of Hands-Math. The common symbol order variations are detected from validation samples of the databases. The additional rules are defined based on these common symbol orders.

All the experiments were performed on an Intel(R) Core$^{\text{TM}}$ 2 Duo E8500 3.16-GHz CPU with 2.0 GB memory.

Table 3 Comparison of hard decision and soft decision methods selected after parsing on CROHME 2013 (%)

Method	Performance		
	Rec. (%)	Pre. (%)	Avg. time/ME (s)
Hard decision	45.49	44.82	0.359
Soft decision	88.49	91.70	0.651

Table 4 Isolated symbol recognition rates on CROHME 2013 (%)

Method	Top N rate	
	Result (top 1 rate)	Result (top 5 rate)
Online	76.53	93.29
Offline	83.68	98.29
Combination	85.84	98.34

4.2 Experiments on CROHME 2013 and 2014

The first experiment compared the symbol segmentation performance of each method using a single threshold (hard decision) or two thresholds (soft decision) on CROHME 2013. For the hard decision, the default threshold of SVM was chosen as $T = 0.0$. For the soft decision, $T_{ns} = -1.0$ and $T_s = 1.4$ were set by finding the best segmentation rate on the validation set. The results for the hard decision and soft decision methods selected after parsing are listed in Table 3. It is clear from the table that the soft decision achieves significantly improved segmentation. On the other hand, the soft decision is two times slower than the hard decision because the system has to generate and recognize more hypotheses when segmentation is undecided.

As described in Sect. 3.2, online and offline methods were combined for mathematical symbol recognition. The second experiment was carried out without the additional size penalty in Eq. (2). The symbol recognition rates on CROHME 2013 (top one and top five) are listed in Table 4. The best combination has a recognition rate of 85.84 %, which is higher than the rates (76.53 and 83.68 %) achieved by online or offline classifier alone, respectively. The performance of the online classifier is lower than that of the offline classifier by seven points due to large variations in stroke direction and stroke order.

Table 5 shows the comparison of our combined classifier and others having joined the isolated symbol recognition task in CROHME 2014 (there are 9 systems of 7 groups). Our two offline classifiers [29] participated in this task with recognition rates as 84.31 and 82.08 %. Our combined classifier in this paper takes the fifth position among 10 classifiers. The better classifiers developed by Álvaro et al. (I), MyScript (III), and Davila et al. (IV) use online and offline features and advanced classification methods such as SVM (Davila et al.) and neural network (Álvaro et al., MyScript).

Table 5 Isolated symbol recognition on CROHME 2014 (%)

Method	(Top 1 rate)
I-A	91.24
I-B	89.79
II	82.72
III	91.04
IV	88.66
V	85.00
VI-A	84.31
VI-B	82.08
VIII	77.25
Our combined method	85.33

The third experiment evaluated the performance of the structural analysis. From the CROHME 2013 database, 84,514 relations were extracted from the 8836 training MEs and 5885 relations were extracted from the 761 testing MEs. The coefficients r_1 and r_2 for body box were adjusted by simply enumerating cases by changing values of r_1 and r_2 from 0 to 1 every 0.05 step and finding the best classification rate on the validation set with the best result obtained by r_1 and r_2 as 0.3 and 0.15, respectively. Body box in combination with two SVMs was compared with bounding box, normalized center, normalized size and center with two SVMs. It was also compared with four features (Dx, Dy, H, O) extracted from body box in combination with a single SVM as well as the method by Álvaro et al. (7 geometric features without the parameter Dx with a single SVM) [26]. For normalized center, we followed the method in Okamoto et al.'s paper [16]. For normalized size and center, we searched for the best value for the height of the ascending and descending parts by the enumeration method with the result of 0.2 of the height of a bounding box. The classification rate achieved by the proposed method was evaluated with changing th_{Dx} from 0.2 to 0.6. Table 6 lists the classification results. We employed paired t test to compare the performance. The classification of n samples can be assumed as Bernoulli trials, and the probability of misclassification is p. Therefore, the mean and variance are calculated as np and $np(1 - np)$, respectively. Body box with two SVMs recorded the best performance. It is significantly better than bounding box, normalized size and center, four features in combination with a single SVM, and the Álvaro et al.'s method with $P < 0.05$ by paired t tests although the significance is not verified against normalized center. The distributions of horizontal, superscript, and subscript relations determined by using the body box and the bounding box are visualized in Fig. 11. It is clear that the body box discriminates the features better than the bounding box.

Table 6 Performance of structural analysis on CROHME 2013 test set(%)

Dx	Method					
	Body box + 2 SVMs	Bounding box + 2 SVMs	Normalized center + 2 SVMs	Normalized center and center + 2 SVMs	Four features + a single SVM	Álvaro et al.'s method
0.2	96.04	94.09	95.63	94.72		
0.3	96.18	94.43	95.82	94.87		
0.4	95.94	94.22	95.58	94.61	91.59	89.89
0.5	95.65	93.98	95.29	94.29		
0.6	94.73	93.15	94.32	93.25		

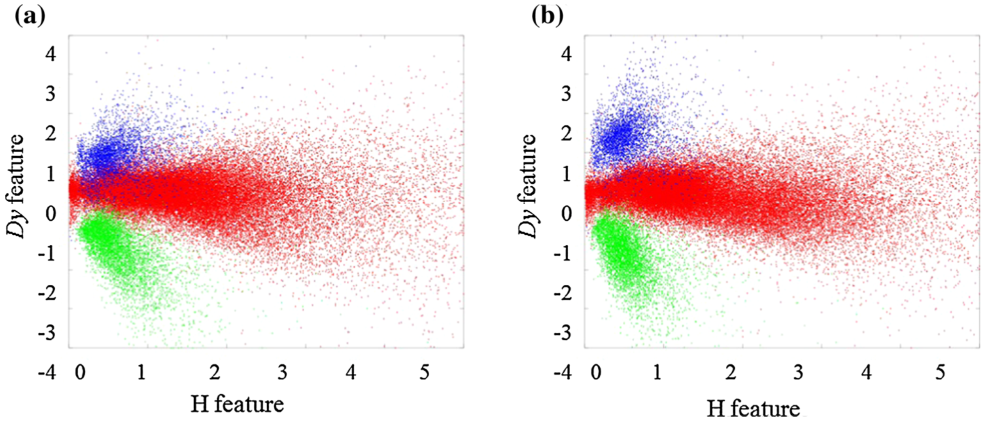


Fig. 11 Distribution of *horizontal*, *superscript*, and *subscript* relations determined with using a body box and a bounding box. **a** Bounding box. **b** Body box

Examples in the CROHME 2013 database in which the body box produces correct relations while the normalized center of bounding box fails are shown in Fig. 12.

In the final evaluation, the performance of the whole system was evaluated. The four measurements performed in the evaluation, namely Sym Seg as symbol segmentation, Sym Seg + Rec as symbol segmentation and recognition, Rel Tree as structural analysis (termed "relation tree"), and Exp Rec as expression recognition in CROHME 2013 and 2014, are listed in Tables 7 and 8. By achieving expression recognition accuracy of 19.97 % in CROHME 2013 (System II) and 25.66 in CROHME 2014 (System VI), the proposed system was ranked third and fourth in the two contests among the participants [28,29,35]. The last column of Table 7 shows the updated results of systems that participated in CROHME 2014. MyScript's system (system VII in Table 7 and system III in Table 8, using in-house training patterns), were ranked top for Exp Rec. Except MyScript's system, Ávaro's system (system IV in Table 7 and system I in Table 8) achieved the best result using the CROHME database alone. The proposed system achieved very low recognition rate for Rec Tree on CROHME 2013 because heuristic rules were used to identify relations.

In light of the competition's results, symbol segmentation and structure analysis were improved as described in Sect. 3. In particular, the number of production rules was increased from 175 to 194 to cope with common symbol orders and the parameters were estimated in the validation set. Then, the proposed algorithm was re-evaluated using the same databases. For CROHME 2013, the recognition accuracies achieved for Rel Tree and Exp Rec were improved to 71.17 and 32.34 %, respectively. The accuracy for the top five candidates reaches 43.37 % for Exp Rec. Our result is the best result among the others, except that for System VII. For CROHME 2014, the recognition accuracy achieved for Exp Rec was improved to 32.86 % with additional rules. Although the accuracy of our system in CROHME 2014 is higher than in CROHME 2013, our system was ranked third in CROHME 2014.

Although the performance of the proposed system was improved, the gap with MyScript's system is still large. This gap is probably due to the difference between the training samples and the lack of a statistical language model. MyScript's system employs 30,000 MEs for training and a very large corpus (hundreds of thousands of equations) for estimating the statistical language model. Using more training samples improves symbol segmentation and recognition as well as structural analysis, whereas the statistical language model revises ambiguities that the grammar cannot solve.

The average processing times for a single ME are 0.651 seconds with additional production rules and 0.649 seconds without additional production rules on CROHME 2013 (average stroke number is 12.74 and symbol number is 9.06), 0.813 seconds with additional production rules, and 0.769 seconds without additional production rules on CROHME 2014 (average stroke number is 13.99 and symbol number is 10.14). This result implies that the time cost of additional production rules is almost negligible. Although there is no available information about the processing times of the other systems to compare with, we believe that the processing time of the proposed system is small enough to allow the system to be implemented in common tablets.

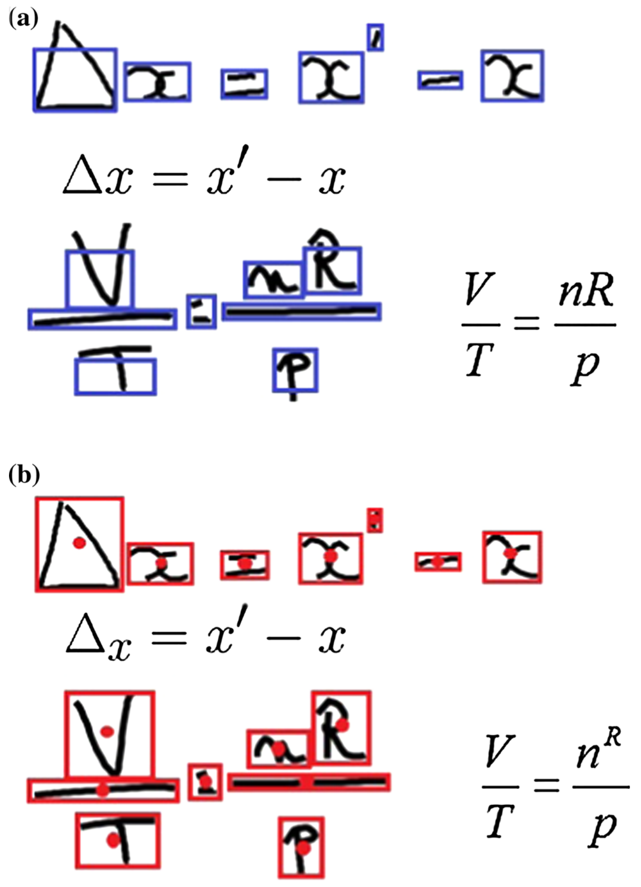


Fig. 12 Examples that body box produces correct relations, while normalized center of bounding box fails (recognition results are shown under/on the right of samples). **a** Result by body box (body boxes are shown in *blue rectangles*). **b** Results by normalized center of bounding box (bounding boxes are shown in *red rectangles* and normalized centers are shown in *red dots*)

4.3 Experiments on Hands-Math

The whole proposed system (with and without additional production rules) was re-evaluated by using the Hands-Math database. The re-evaluation results for the testing set used are listed in Table 9. With increasing the production rules from 148 to 181, the performance is improved in all measures. This result implies that Hands-Math has more symbol order variations than CROHME 2013, 2014 and children may write MEs in more ways than adults.

4.4 Analysis of recognition results and future works

Some correctly and incorrectly recognized samples in the CROHME 2013, 2014, and Hands-Math databases are shown in Figs. 13 and 14. The experimental evaluation described above demonstrates that the performance of the ME recognition system is improved by using soft decision in segmentation, a body box in structural analysis, and additional production rules. Stroke order is employed to reduce the unfeasible partitions and reduce processing time for parsing. The computation time of the proposed system is feasible enough to avoid imposing a waiting time on users.

Table 7 Recognition performance on CROHME 2013 test set in CROHME 2013 and 2014 (%)

System	Measure				
	Sym Seg	Seg + Class	Rel Tree	Exp Rec	Updated Exp Rec
VII	97.86	93.03	88.65	60.36	62.15
IV	84.97	73.94	49.73	23.40	30.70
II	80.70	66.41	22.44	19.97	25.04
III	85.24	62.63	53.24	9.39	–
VI	57.86	47.68	33.63	8.35	–
I	46.93	25.19	24.85	2.68	–
VIII	90.32	73.84	50.19	18.33	17.59
V	84.45	66.66	41.34	14.31	19.52
Updated II w/o additional productions	87.76	78.78	69.63	–	31.74
Updated II with additional productions	88.49	79.63	71.17	–	32.34

Table 8 Recognition performance on CROHME 2014 test set in CROHME 2014 (%)

System	Measure			
	Sym Seg	Seg + Class	Rel Tree	Exp Rec
I	93.31	86.59	84.23	37.22
II	76.33	66.97	60.31	15.01
III	98.42	93.91	94.26	62.68
IV	85.52	76.64	70.78	18.97
V	88.23	78.45	61.38	18.97
VI	83.05	69.72	66.83	25.66
VII	89.43	76.53	71.77	26.06
Updated VI w/o additional productions	82.92	74.96	61.95	31.03
Updated VI with additional productions	84.67	77.21	68.62	32.86

Table 9 Recognition performance on Hands-Math (%)

System	Measure			
	Sym Seg	Seg + Class	Rel Tree	Exp Rec
W/o additional productions	88.63	84.04	78.02	56.51
With additional productions	92.46	88.24	89.79	68.07

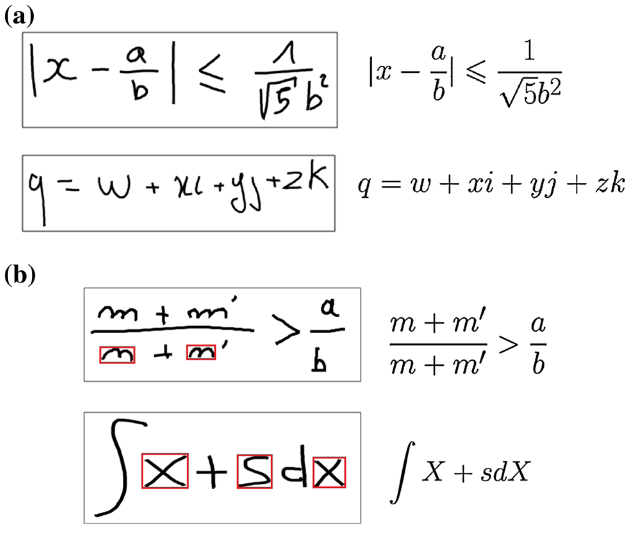

(a)
$|x - \frac{a}{b}| \leqslant \frac{1}{\sqrt{5}b^2}$
$q = w + xi + yj + zk$
(b)
$\frac{m+m'}{m+m'} > \frac{a}{b}$
$\int X + sdX$

Fig. 13 Examples in CROHME 2013, 2014 (recognition results are shown on the right of samples). **a** Correctly recognized MEs. **b** Misrecognized MEs (errors are located in *red rectangles*)

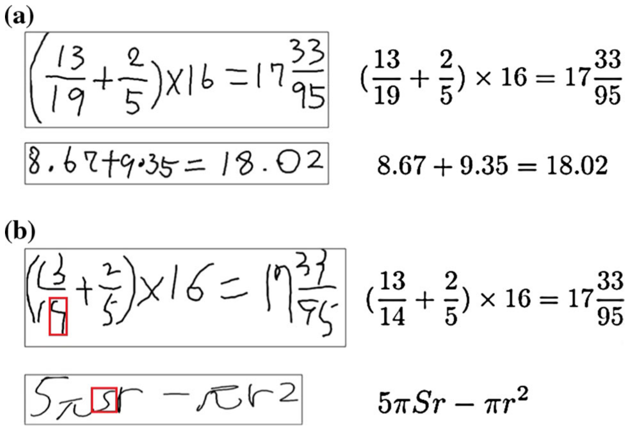

(a)
$(\frac{13}{19} + \frac{2}{5}) \times 16 = 17\frac{33}{95}$
$8.67 + 9.35 = 18.02$
(b)
$(\frac{13}{14} + \frac{2}{5}) \times 16 = 17\frac{33}{95}$
$5\pi Sr - \pi r^2$

Fig. 14 Examples in Hands-Math (recognition results are shown on the right of samples). **a** Correctly recognized MEs. **b** Misrecognized MEs (errors are located in *red rectangles*)

On the other hand, we analyzed incorrectly recognized MEs after experiments had be done. It shows that misrecognitions occur for MEs containing delay strokes, ambiguities in segmentation and symbol recognition, and structural relations. Moreover, SCFG is biased to consider non-terminals with fewer children. The accuracy of the proposed ME recognition system could be improved by achieving the following tasks:

- Adding weights (w_1, w_2, w_3, w_4) to evaluation function as the following formula:

$$C = w_1 \sum ln(P_{\text{seg}}(S_j, S_{j+1})) + w_2 \sum ln(P_{\text{rec}}(S_i|G_i)) + w_3 \sum ln(P_{\text{rel}}(R_k|SE_k)) + w_4 \sum ln(P_{\text{Gram}}) \quad (4)$$

- Devising a method for partitioning input strokes and parsing independent from stroke order.
- Adding more context information for the ME recognition system through training SCFG by the inside–outside algorithm and using another language model such as N-gram. Moreover, it is necessary to collect a large ME corpus from the internet or books.
- Collecting more ME patterns from students and availing them for research purposes.

5 Conclusion

A system for recognizing online handwritten MEs was proposed and evaluated. It employs SCFG and the CYK algorithm in combination to represent MEs and parse their 2D structures. When SCFG cannot solve local ambiguities in symbol segmentation, symbol recognition, and structural analysis, the proposed system presents a list of candidates that may contain the correct result. For symbol segmentation, soft decision is employed so that errors can be recovered in later processes. To resolve local ambiguities in the results of structural analysis, a concept of body box representing the main body of a symbol and two SVM models is applied. Stroke order is used to reduce the number of symbol hypotheses, limit the number of sub-expressions, and achieve less time complexity of the parsing algorithm. To handle common symbol orders that people often write, productions are added for SCFG. Experimental evaluations using the CROHME 2013, 2014 databases and an in-house (Hands-Math) database show that the recognition rate of the proposed system is improved and its processing time is kept to a practical level.

Acknowledgments This research is being supported partially by Grant-in-Aid for Scientific Research under contract No. (B) 24300095, NEDO under the No. 27J1103, and JSPS fellowship under the No. 15J08654. The authors thank Dr. Bilan Zhu, Dr. Truyen Van Phan, Mr. Cuong Tuan Nguyen, Mr. Hai Dai Nguyen, Mr. Khanh Minh Phan, and other members of Nakagawa Laboratory for availing the symbol recognizer, giving their helpful comments and suggestions. We employ the CROHME 2013, 2014 databases and we are indebted to the people for availing the databases.

References

1. Ruzzo, W.L.: On the complexity of general context-free language parsing and recognition, In: Automata, Languages, and Programming, pp. 489–497. Lecture Notes in Computer Science. Springer Verlag (1979)
2. Anderson, R.H.: Syntax-directed recognition of handprinted two-dimensional mathematics. Ph.D. thesis, Harvard University (1968)
3. Chang, S.K.: A method for the structural analysis of two-dimensional mathematical expressions. Int. J. Inf. Sci. **2**, 253–272 (1970)
4. Chan, K.F., Yeung, D.Y.: Mathematical expression recognition: a survey. Int. J. Doc. Anal. Recogn. **3**, 3–15 (2000)
5. Zanibbi, R., Blostein, D.: Recognition and retrieval of mathematical expressions. Int. J. Doc. Anal. Recogn. **15**, 331–357 (2012)
6. Lehmberg, S., Winkler, H.J., Lang, M.: A soft-decision approach for symbol segmentation within handwritten mathematical expressions. In: Proceeding of international conference on acoustics, speech, and signal processing, vol. 6, pp. 3434–3437. Atlanta (1996)
7. Toyozumi, K., et al.: A study of symbol segmentation method for handwritten mathematical formula recognition using mathematical structure information. In: Proceedings of the 17th ICPR, vol. 2, pp. 630–633. Cambridge (2004)
8. Shi, Y., Li, H., Soong, F.K.: A unified framework for symbol segmentation and recognition of handwritten mathematical expressions. In: Proceedings of the 9th ICDAR, vol. 2, pp. 854–858. Curitiba (2007)
9. Hu, L., Zanibbi, R.: Segmenting handwritten math symbols using adaboost and multi-scale shape context features. In: Proceedings of the 12th ICDAR, pp. 1180–1184. Washington (2013)
10. MacLean, S., Labahn, G.: Elastic matching in linear time and constant space. In: Proceedings of the 9th IAPR Workshop on Document Analysis Systems. (2010)
11. Hu, L., Zanibbi, R.: HMM-based recognition of online handwritten mathematical symbols using segmental K-means initialization and a modified pen-up/down feature. In: Proceedings of the 11th ICDAR, pp. 457–462. Beijing (2011)
12. Álvaro, F., Sánchez, J. A., Benedí, J. M.: Classification of on-line mathematical symbols with hybrid features and recurrent neural networks. In: Proceedings of the 12th ICDAR, pp. 1012–1016. Washington (2013)
13. Davila, K.M., Ludi, S., Zanibbi R.: Using off-line features and synthetic data for on-line handwritten math symbol recognition. In: Proceedings of the 14th ICFHR, pp. 323–328. Crete (2014)
14. Garain, U., Chaudhuri, B.B.: Recognition of online handwritten mathematical expressions. IEEE Trans. Syst. Man Cybern. B Cybern. **34**, 2366–2376 (2004)
15. Álvaro, F., Sánchez, J.A., Benedí, J.M.: Offline features for classifying handwritten math symbols with recurrent neural networks. In: Proceedings of the 22nd ICPR, pp. 2944–2949. Stockholm (2014)
16. Okamoto, M., Miao, B.: Recognition of mathematical expressions by using the layout structure of symbols. In: Proceedings of the 1st ICDAR, pp. 242–250. Saint Malo (1991)
17. Eto, T., Suzuki, M.: Mathematical formula recognition using virtual link network. In: Proceedings of the 6th ICDAR, pp. 762–767. Seattle (2001)
18. Aly, W., Uchida, S., Fujiyoshi, A., Suzuki, M.: Statistical classification of spatial relationships among mathematical symbols. In: Proceedings of the 10th ICDAR, pp. 1350–1354. Barcelona (2009)
19. Álvaro, F., Zanibbi, R.: A shape-based layout descriptor for classifying spatial relationships in handwritten math. In: Proceedings of the ACM Symposium on Document Engineering. Florence (2013)
20. Zanibbi, R., Blostein, D., Cordy, J.R.: Recognizing mathematical expressions using tree transformation. IEEE Trans. PAMI, **24**, 1455–1467 (2002)
21. Rhee, T.H., Kim, J.H.: Efficient search strategy in structural analysis for handwritten mathematical expression recognition. Pattern Recogn. **42**, 3192–3201 (2009)
22. Yamamoto, R., Sako, S., Nishimoto, T., Sagayama, S.: OnLine recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar. In: 10th IWFHR, pp. 249–254. La Baule (2006)
23. Simistira, F., Katsouros, V., Carayannis, G.: Recognition of online handwritten mathematical formulas using probabilistic SVMs and stochastic context free grammars. Pattern Recogn. Lett. **53**, 85–92 (2015)
24. MacLean, S., Labahn, G.: A New approach for recognizing handwritten mathematics using relational grammars and fuzzy sets. Int. J. Doc. Anal. Recogn. **16**, 139–163 (2013)
25. MacLean, S., Labahn, G.: A Bayesian model for recognizing handwritten mathematical expressions. Pattern Recogn. **48**, 2433–2445 (2015)
26. Álvaro, F., Sánchez, J.A., Benedí, J.M.: Recognition of on-line handwritten mathematical expressions using 2D stochastic context-free grammars and hidden Markov models. Pattern Recogn. Lett. **35**, 58–67 (2014)
27. Awal, A.M., Mouchére, H., Viard-Gaudin, C.: A global learning approach for an online handwritten mathematical expression recognition system. Pattern Recogn. Lett. **35**, 6877 (2014)
28. Mouchère, H., Viard-Gaudin, C., Zanibbi, R., Garain, U., Kim, D.H., Kim, J.H.: ICDAR 2013 CROHME: third international

competition on recognition of online handwritten mathematical expressions. In: Proceedings of the 12th ICDAR, pp. 1428–1432. (2013)

29. Mouchère, H., Viard-Gaudin, C., Zanibbi, R., Garain, U.: ICFHR: competition on recognition of on-line handwritten mathematical expressions (CROHME 2014). In: Proceedings of the 14th ICHFR, pp. 791–796. Crete (2014)
30. Anthony, L., Yang, J., Koedinger, K.R.: Evaluation of multi modal input for entering mathematical equations on the computer. In: HI ’05 Extended Abstracts on Human Factors in Computing Systems, pp. 1184–1187
31. Anthony, L., Yang, J., Koedinger, K.R.: A paradigm for handwriting-based intelligent tutors. Int. J. Hum. Comput. Stud. **70**, 866–887 (2012)
32. Le, A.D., Nakagawa, M.: A tool for ground-truthing online handwritten mathematical expressions. In: Proceedings of the 16th International Graphonomics Society Conference, pp. 70–73. (2013)
33. Zhu, B., Gao, J., Nakagawa, M.: Objective function design for MCE-based combination of on-line and off-line character recognizers for on-line handwritten Japanese text recognition. In: Proceedings of the 11th ICDAR, pp. 594–599. (2011)
34. Lari, K., Young, S.J.: The estimation of stochastic context-free grammars using the inside–outside algorithm. Comput. Speech Lang. **4**, 35–56 (1990)
35. Le, A.D., Phan, T.V., Nakagawa, M.: A System for recognizing online handwritten mathematical expressions and improvement of structure analysis. In: Proceedings of the 11th IAPR Workshop on Document Analysis Systems, pp. 51–55. (2014)