PAPER

Automatic Classification of Spatial Relationships among Mathematical Symbols Using Geometric Features

Walaa ALY[†a)], ***Nonmember***, **Seiichi UCHIDA**[†b)], ***and*** **Masakazu SUZUKI**[††], ***Members***

SUMMARY Machine recognition of mathematical expressions on printed documents is not trivial even when all the individual characters and symbols in an expression can be recognized correctly. In this paper, an automatic classification method of spatial relationships between the adjacent symbols in a pair is presented. This classification is important to realize an accurate structure analysis module of math OCR. Experimental results on very large databases showed that this classification worked well with an accuracy of 99.525% by using distribution maps which are defined by two geometric features, relative size and relative position, with careful treatment on document-dependent characteristics.

key words: *spatial relationships, mathematical documents, geometric features, mathematical symbols*

1. Introduction

Automatic recognition of mathematical expressions is considered a basic process in converting scientific and engineering documents into an electronic form. This process is composed of two parts: (i) recognition of characters and mathematical symbols and (ii) structure analysis of mathematical expressions. Although there have been many attempts to recognize mathematical documents [1], there are still many unsolved problems toward the realization of practical recognition systems.

In this paper, we consider the structure analysis part, especially, the automatic classification of spatial relationships between each adjacent pair of characters and/or mathematical symbols, such as, "σ_i", "a^x", and "$\int f$." Hereafter, this task is simply called *classification task*. Classification of spatial relationships is very important to recognize mathematical expressions because the same set of symbols convey different meaning depending on the spatial relationships. For example, "ab", "a_b", and "a^b" have the same symbols but introduce different meaning in mathematical expressions.

We assume there are five classes of spatial relationships: horizontal ("xt"), subscript ("x_t"), superscript ("x^t"), upper ("$\overset{k}{\sum}$"), and lower ("$\underset{k}{\sum}$") classes.

Throughout this paper, we assume the correct category is given for every character and symbol, that is, we assume that recognition of characters and mathematical symbols has already been done. This assumption is rather realistic when we focus on the structure analysis part; in most math OCRs, in fact, the structure analysis is done after recognizing individual characters and symbols.

The main contribution of this paper is to tackle the classification task by a statistical decision method grounded by a deep analysis of huge databases. In the proposed method, the importance of using document-dependent characteristics and symbol types will be fully emphasized as well as reasonable feature extraction for specifying the spatial relationships. Experimental results revealed that the classification can be done almost perfectly (~ 99.525%) by the proposed method.

In the proposed method, we will use two features called *relative size* H and *relative position* D between a pair of adjacent symbols. These features are very important to specify the spatial relationships. Intuitively speaking, the relative size H is useful to discriminate between baseline and non-baseline classes and the relative position D is useful to discriminate among subscript, superscript, upper, and lower classes.

Figure 1 shows an overview of the proposed method. Again, our task is the classification of spatial relationships between each adjacent pair (parent-child) in mathematical expressions, where the parent and the child are the first and the second symbol (or character) of each pair, respectively. This classification task is done on *distribution maps* which plot the feature vectors (H, D) in a two-dimensional space. Specifically, the relationship between each adjacent pair is classified using Bayesian classifier which classifies each point on the distribution maps into one of the 5 classes. It is interesting to note that even this simple Bayesian classifier could achieve the above high recognition rate, 99.525%.

As will be detailed in this paper, the proposed method introduces several new techniques to deal with huge variations in the classification task; for example, *symbol types* are introduced to compensate the variation of symbol sizes. Each symbol has a type according to its size and position. Different distribution maps are prepared for individual symbol types. That is, different classification tasks will be done according to the symbol types. In addition, document-dependent processing is also introduced to improve the performance of the classification task. Furthermore, very large databases are used in the classification task. These databases are suitable for giving a strong ground to the proposed

Manuscript received March 4, 2009.
Manuscript revised June 30, 2009.

[†] The authors are with the Department of Intelligent Systems, Kyushu University, Fukuoka-shi, 819–0395 Japan.
[††] The author is with the Department of Mathematics, Kyushu University, Fukuoka-shi, 812–8581 Japan.
a) E-mail: walaa@human.is.kyushu-u.ac.jp
b) E-mail: uchida@human.is.kyushu-u.ac.jp

DOI: 10.1587/transinf.E92.D.2235

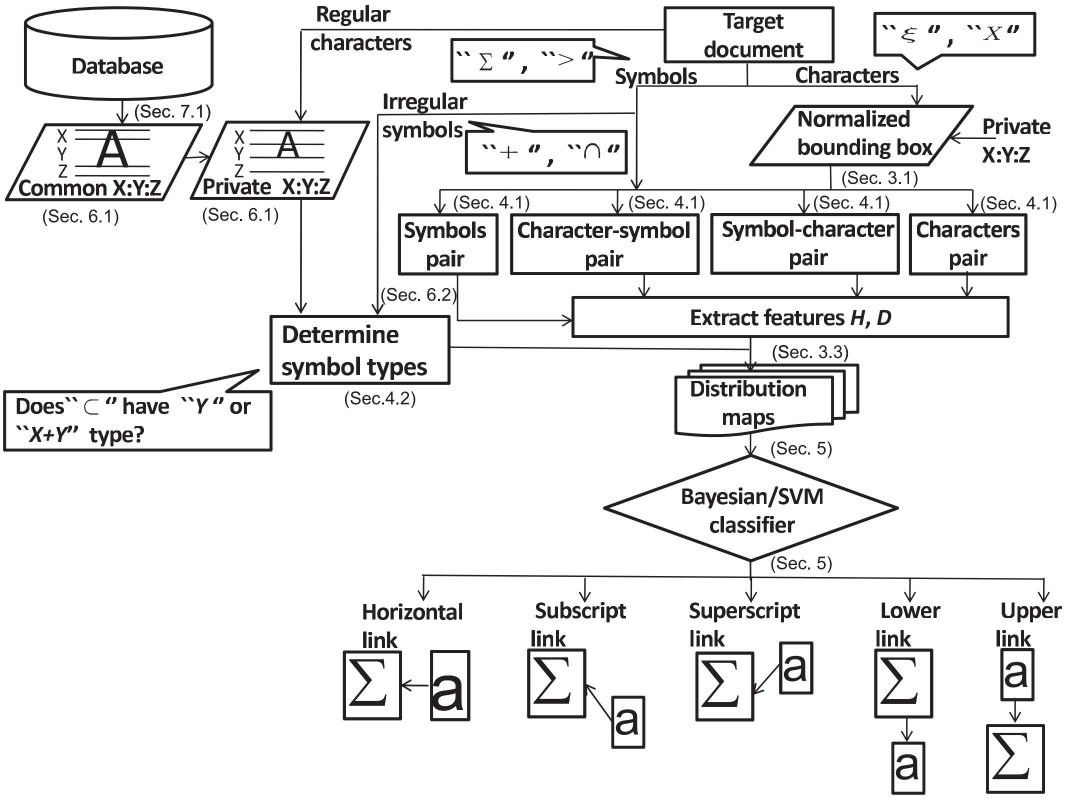


Fig. 1 Outline of the proposed method.

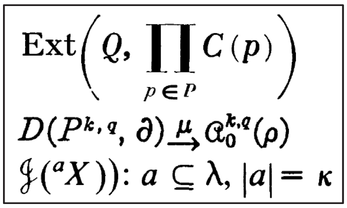

$\mathrm{Ext}\left(Q, \prod_{p \in P} C(p)\right)$
$D(P^{k,q}, \partial) \xrightarrow{\mu} \mathfrak{a}_0^{k,q}(\rho)$
$\mathfrak{J}({}^{a}X)): a \subsetneq \lambda, |a| = \kappa$

Fig. 2 Examples of mathematical expressions.

method.

Our initial work on this classification task was reported in [14]. This is very preliminary work because only the spatial relationships among characters were focused upon. Thus, the lower pair of "∏" and "p" and the horizontal pair of "D" and "(" in Fig. 2 were out of its scope. The present work is a large extension of the previous work because we consider both symbols and characters. In other words, the classification task in this paper deals with all the adjacent pairs in the mathematical expressions. Furthermore, as noted above, the symbol types and the document-dependent processing are newly introduced for this extension.

The remainder of this paper is organized as follows: Section 2 presents a brief review of the related work. Section 3 introduces the two spatial features, H and D, which were used in the distribution map. Section 4 gives details about character/symbol combination types, as well as symbol types, for dealing with the large variation of symbols. Section 5 describes how to use the features in the classification task. Section 6 presents a document-dependent processing. Section 7 shows experimental results with very large databases. Finally, Sect. 8 presents a conclusion and future work.

2. Related Work

The structure analysis has been discussed by many researchers started from Anderson's purely syntactic approach [2] for parsing mathematical expressions. Most of other past attempts have relied on geometric information to classify the spatial relationships of adjacent symbols for the structure analysis. On this meaning, the proposed method has very close relation to the past attempts. For example, Okamoto and his colleagues [3], [4] classified the spatial relationships using geometric information like relative size and relative position. A similar methodology can be found in Suzuki et al. [5] and Garaian et al. [6], [7]. Zanibbi et al. [8] also used some geometric information in their system for recognizing typeset and handwritten mathematical expressions.

Unfortunately, the above attempts did not give details about the classification task[†]; they gave only the total performance of the system and specified neither quantitative nor

[†]Moreover, in other attempts [9]–[13], the detail of structure part is completely concealed as a black box of a large math OCR system.

qualitative analysis of their results. This may be because (i) the classification task is one module of a large math OCR system, (ii) it employs many heuristics whose details are often hidden from readers, and (iii) it should be evaluated with a large-scale database, which was not available in the past.

In contrast, in this paper, we concentrate on the classification task, give the detail of the methodology to tackle the task with an experimental backup by huge database, and show the fact that almost perfect classification rate (99.525%) can be achieved by the method. In addition, we will observe the effect of two important techniques, i.e., document-dependent processing and symbol type-dependent processing, which are newly introduced in this paper.

3. Features Extraction for Discriminating the Spatial Relationships

Generally, symbols (e.g., "$\int$", "$\sum$", "$\in$") have more variation in their heights, widths, and positions than characters (e.g., "x", "2", "α"). Thus, different techniques are applied in extracting the features of characters and symbols as follows.

3.1 Normalized Bounding Box

As stated in [14], for extracting the two features, relative size H and relative position D, from a pair of adjacent characters, we must care about the difference of character sizes. We will use a *normalized bounding box* for each character instead of the actual bounding box to compensate the differences in the character sizes. Figure 3 (a) shows the actual bounding box for character "α" and Fig. 3 (b) shows the normalized bounding box for character "α."

For setting the normalized bounding box of each character, a virtual ascender or a virtual descender or both are added to the actual bounding box. This addition depends on the character category. For example, the normalized bounding box of the characters without ascender and descender (e.g., "a", "c", "e") needs the virtual ascender and descender. Similarly, the normalized bounding box of the characters without descender (e.g., "b", "d", "h") needs the virtual descender. The heights of the virtual descender and ascender are determined by the technique detailed in Sect. 3.2.

Unfortunately, this technique is not applicable to symbols. For example, it is very difficult to set the normalized height for "=" and "−." It is also difficult to set the normalized height for "[" and "|" because those elongated symbols often stick out of both ascender and descender parts.

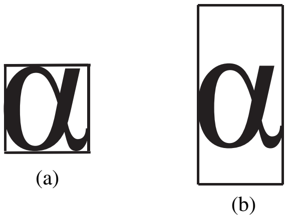


Fig. 3 (a) Original bounding box for character "α." (b) Normalized bounding box for character "α."

Consequently, normalized bounding box can compensate the variation of *character* sizes and cannot compensate the variation of *symbol* sizes because the latter have more size variations. Instead, we will classify the symbols into 6 types according to their heights and positions and then perform the classification task on each type independently. In other words, we give up to normalize all the symbols for performing one unified classification task and instead prepare different classification tasks according to the symbol types. The detail of the symbol types will be discussed in Sect. 4.2.

3.2 $X:Y:Z$ Ratio

On setting the normalized bounding box for characters, we must know the height of the virtual ascender and descender. This can be derived from the height ratio of three regions, called X, Y, and Z regions. Figure 4 shows X, Y and Z regions. The regions X and Z corresponds to the ascender and the descender parts, respectively.

If we have the ratio of the heights of X, Y and Z (hereafter called $X:Y:Z$ ratio), we can calculate the heights of ascender/descender or equivalently the height of the normalized bounding box. For example, the height of the normalized bounding box of the character "δ" can be obtained by estimating the height of its virtual Z and it will be calculated by multiplying the actual height of the "δ" by the ratio between X+Y+Z and X+Y.

In the estimation of the $X:Y:Z$ ratio of a document, the heights of X, Y and Z for each baseline character of a document are first measured. At the measurement, we need to refer to its category (i.e., recognition result). For example, if the category of a character is "A", the height of the X+Y region is measured. Then the measured heights are averaged to calculate the $X:Y:Z$ ratio for all the baseline characters of the document. Note that X:Y:Z ratio is different in each document. This important fact will be discussed in Sect. 6.1.

The $X:Y:Z$ ratio for non-baseline characters is also estimated in the same way. This is because they often have their own X:Y:Z ratio (which is slightly different from the X:Y:Z ratio of the baseline characters) due to their own font shapes[†].

3.3 Feature Extraction

For the classification task robust to the variations of characters, the two features H and D are defined by a normalized

[†] Readers may be confused by the fact that we need to discriminate between baseline characters and non-baseline characters for estimating their own $X:Y:Z$ ratio during the process toward our final goal, i.e., the classification task. For this discrimination we used a predetermined X:Y:Z ratio which calculated from all characters contained in the database. Of course, the result from this discrimination includes some errors. These errors do not affect the estimation seriously because we use the average of the heights.

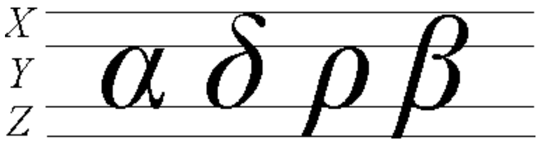


Fig. 4 *X*, *Y* and *Z* regions.

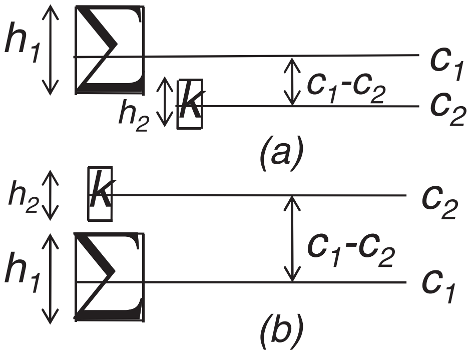


Fig. 5 (a) Normalized sizes (h_1, h_2), normalized centers (c_1, c_2) for adjacent pair of subscript class. (b) Normalized sizes (h_1, h_2), normalized centers (c_1, c_2) for adjacent pair of upper class.

size and a normalized center. For characters, the normalized size and the normalized center are equal to the height and the center of the *normalized* bounding box, respectively. For symbols, the normalized size and the normalized center are equal to the height and the center of the *actual* bounding box[†], respectively.

Let h_1 and h_2 denote the normalized sizes of the parent and the child, respectively. Similarly, let c_1 and c_2 denote the normalized centers of those pairs. Figures 5 (a) and (b) illustrate these parameters. The relative size H and the relative position D can be extracted for each adjacent pair as follows[††]:

$$H = \frac{h_2}{h_1}, \tag{1}$$

$$D = \frac{c_1 - c_2}{h_1}. \tag{2}$$

4. Combination Types and Symbol Types

4.1 Combination Types

Clearly from the above discussion, the (H, D) features of characters and symbols are very different from each other. This fact implies that the distribution of the (H,D) features are also different according to the combination of the character and/or the symbol of the adjacent pair. For example, in the superscript class, the (H,D) feature of an adjacent pair by a parent symbol and a child character, such as "Σ^n" and "$)^n$", may be different from that of an adjacent pair by a parent symbol and a child symbol, such as "Σ^+" and "$)^+$." (Recall that the height of each character is measured at its normalized bounding box, whereas that of each symbol is measured at its actual bounding box. Thus, their H and D features will be different).

Table 1 Symbol types.

Types	Examples of symbols				
X+Y+Z	$\leq$	$\int$	$]$	$\sum$	$\oplus$
X+Y	$<$	$>$	$\wedge$	$\in$	$\subset$
Y	$-$	$=$	$\sim$	$\leftarrow$	$\smile$
Y+Z	$\perp$	$+$			
X	$\hat{a}$	$\check{a}$	$\breve{a}$	$\tilde{a}$	$\dot{a}$
Z	$\underline{a}$	$\underbrace{a}$			

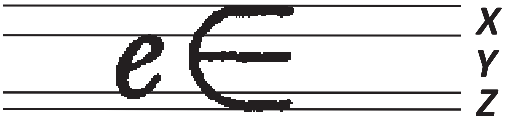


Fig. 6 Determining the type of symbol "∈."

Consequently, the classification task should be done for each of four parent-child *combination types*, such as "character (parent)-character (child)" and "symbol-character." As we will see later, the combination types "character-symbol" and "symbol-character" are majority and the remaining combination types "character-character" and "symbol-symbol" are minority in the mathematical expressions.

4.2 Symbol Types

To compensate the variation in positions and heights of symbols, we will define a type for each symbol. Within symbols, there are huge variations in their positions and heights. (So, we gave up to normalize their bounding boxes). Thus, the distribution of the (H, D) features of large symbols (e.g., "$\sum$", "$\int$") will be very different from that of small symbols (e.g., "$\rightarrow$", "$=$").

Consequently, the classification task should be done for each of *symbol types* in addition to the combination types. We define 6 symbol types such as, "*X*", "*X+Y*", "*X+Y+Z*", "*Y*", "*Y+Z*", and "*Z*" according to *X*, *Y*, and *Z* regions of its neighboring characters. Table 1 shows some examples of symbol types.

Figure 6 shows an example of determining the type of symbol "∈". In this example, the symbol "∈" will have the "*X+Y+Z*" type. Note that for each symbol we should specify not only its type at baseline but also its type at non-baseline because symbols often have different font shapes according to their sizes.

5. Classification Using the Features

The classification task of the spatial relation is done on the distribution map, which is the two-dimensional feature

[†] Exceptionally, accent and minus symbols have the width of the actual bounding box instead of the heights.

[††] We can use other definitions of H and D. For example, we can use "$D = (C_1 - C_2)/((C_1 - C_2) + 1/2(h_1 + h_2))$." The best result by this D is 98.99% and thus the simpler D of Eq. (2) outperforms it.

space spanned by the relative size H and the relative position D. From the discussion in Sect. 4, it is clear that different distribution maps are prepared necessary according to both symbol types and combination types. Since we have 6 symbol types and 4 combinations types, we have, in principle, 49 distribution maps ($= 1 \times 1 + 1 \times 6 + 6 \times 1 + 6 \times 6$). In the databases used in the following experiment, however, several combinations were not found or very rare and therefore we had only 23 distribution maps in the experiment.

The spatial relationships are classified using Bayesian classifier which classifies each point in the distribution maps into one of 5 classes (i.e., "superscript", "horizontal", etc.). Especially, we assumed that each of the five classes has a two-dimensional Gaussian distribution on the distribution map. This assumption reduces Bayesian classifier to a quadratic classifier. All data of each class were used to estimate the parameters (i.e., the empirical mean vector and the empirical covariance matrix) of the Gaussian distribution[†].

We can also use other nonlinear classifiers; one possible choice is SVM. The results of SVM will be shown in Sect. 7 along with the results of the Bayesian classifier.

6. Document-Dependent Processing

Each document has its own characteristics in its type setting. Accordingly, we must normalize them or introduce special consideration on them for the classification task. These treatments on document-dependent characteristics will improve the performance of the classification task. In this section, we will introduce two document-dependent processing.

6.1 Private $X : Y : Z$ Ratio

As stated before, we use the $X : Y : Z$ ratio for normalizing bounding boxes of characters and for setting symbol types. The simplest way to prepare the X:Y:Z ratio is just to use a predetermined *common* X:Y:Z *ratio*, which is estimated by averaging X:Y:Z ratios of all the documents in the database. Another and document-dependent way is to use a *private* X:Y:Z *ratio*, which is estimated for each document by using characters in the target document. The use of the private $X : Y : Z$ ratio outperforms the common $X : Y : Z$ ratio as will be shown in the experimental results. This fact implies that each document use its own font setting and type setting.

6.2 Irregular Symbols and Characters

A deep observation on mathematical documents reveals that some symbols have different symbol types in different documents. These symbols called *irregular symbols*. For example, "$\prec$", "$\subset$", and "$\in$" occupy only Y region in some documents and occupy X+Y regions in other documents, yet occupy the entire X+Y+Z regions in the other documents. Thus, they have 3 different types across documents. Figure 7 shows these types for symbol "$\in$."

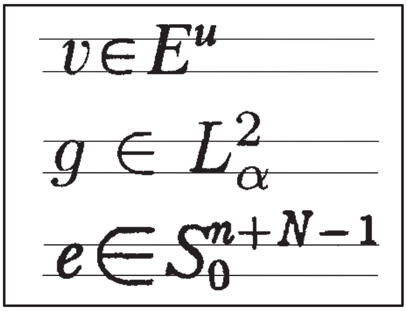


Fig. 7 Symbol "$\in$" from different document.

Similar irregularity can be found in characters. That is, some characters occupy different X:Y:Z regions in different documents [14]. These characters called *irregular characters*. Figures 8 shows some examples of irregular characters. Irregular symbols are more common than irregular characters and have more variations in documents. In the following experiment, 36 irregular symbols (e.g., "$\in$", "$\prec$", "$\wedge$") and 18 irregular characters (e.g., "2", "i", "ψ") were defined.

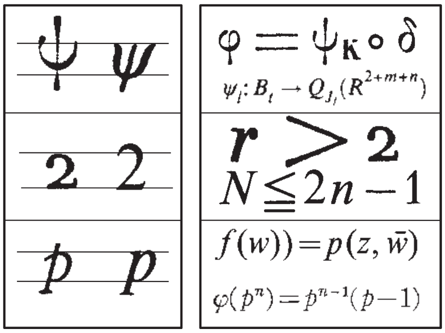


Fig. 8 Examples of irregular characters. The characters on the left part were extracted from the mathematical expressions on the right part.

Special treatments are applied for irregular characters and irregular symbols. Specifically, the following two treatments are applied within each document: (i) elimination of the irregular characters when evaluating the private $X : Y : Z$ ratio, and (ii) estimation of the actual $X : Y : Z$ occupation of each irregular character/symbol by choosing the most probable type according to its size and position from possible types. Using the estimated $X : Y : Z$, the symbol type is estimated for the irregular symbol, and the normalized bounding box is estimated for the irregular character.

7. Experimental Results

7.1 Database

The classification task was conducted on 158,308 adjacent pairs of symbols and characters. Table 2 shows the number of pairs for each parent-child combination type. This huge number of adjacent pairs was extracted from two large databases, InftyCDB-1 [15], [16] and InftyCDB-2 [17], which together consist of 65 English articles (pub-

[†] The prior probability of a Bayesian classifier for each class is estimated as the ratio between the number of samples in this class and the total number of all samples in the database.

Table 2 Number of pairs in the database for each combinational type.

Combination type parent-child	#pairs
character-character	37,263
character-symbol	54,924
symbol-character	52,057
symbol-symbol	14,064

lished between 1949 and 2000), 4 French articles (published between 1974 and 1988), and 7 German articles (published between 1956 and 1987) on pure mathematics. The total number of pages in the databases is 908.

To the authors' best knowledge, these databases are the largest of those used in past attempts on the classification task. For example, they are larger than the database used in [18], which consists of 297 pages. Such large databases are well suited to derive general properties of mathematical expressions and thus also suited to design the classifiers for the spatial relationships.

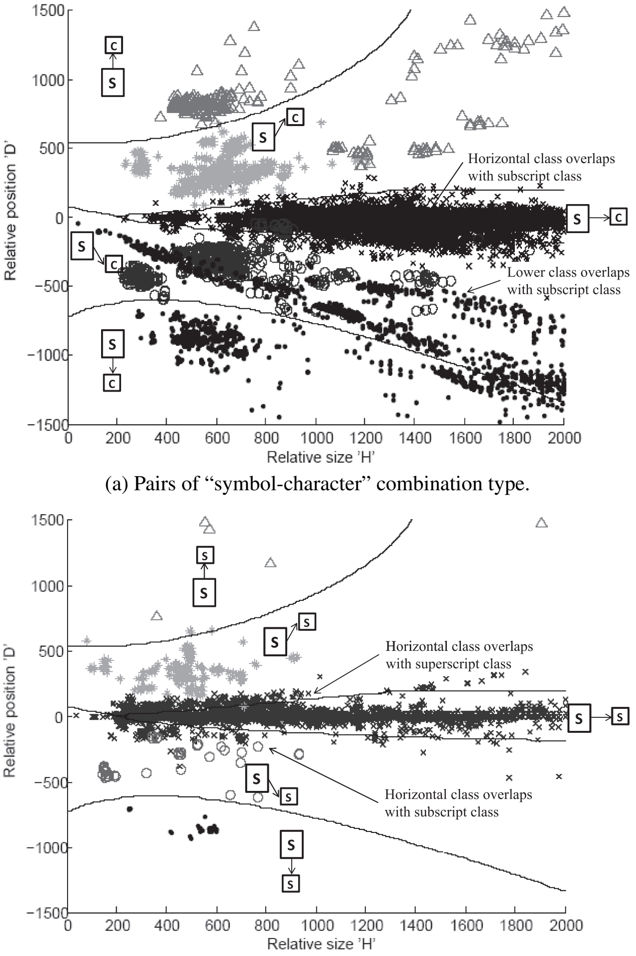
All the mathematical expressions in the database were used, except matrices and fraction expressions. In these two types of expressions, the size of font is often very irregular and thus they will distribute our observations. Matrices and fraction expressions can be detected easily and treated separately from other mathematical expressions.

7.2 Analysis of Distribution Maps

Figures 9 and 10 illustrate several distribution maps whose details will be discussed later. In these maps, each of “×”, “○”, “∗”, “△”, and “•”-shaped dots corresponds to a pair of horizontal, subscript, superscript, upper, or lower class, respectively.

Figures 9 (a) and (b) show some examples of distribution maps when symbol types were *not* specified. Heavy overlaps between the classes can be observed on these maps and thus good classification accuracy cannot be expected. These overlaps come from the variation of the sizes and positions of the actual bounding boxes of symbols. For example, some pairs of lower class overlapped with subscript class and most of those pairs have accent symbols; this is because accent symbols have more variations in their heights and positions. It was also revealed that the overlap between horizontal class and subscript class includes, for example, pairs of a baseline symbol of “X:Y:Z” type and a baseline symbol of “Y+Z”, such as “)+.” These pairs of horizontal class may be misclassified as subscript class.

Figures 10 (a) and (b) show some examples of distribution maps which were separately according to symbol types. That is, symbols of different types such as, “X”, “X+Y”, “X+Y+Z”, “Y”, “Y+Z”, and “Z” were plotted in different distribution maps. The overlaps were drastically decreased because the variations in symbol sizes not large within each symbol type. Consequently, this improvement shows that specification of symbol types are very useful for the classification task with (H, D) features.



(a) Pairs of “symbol-character” combination type.

(b) Pairs of “symbol-symbol” combination type.

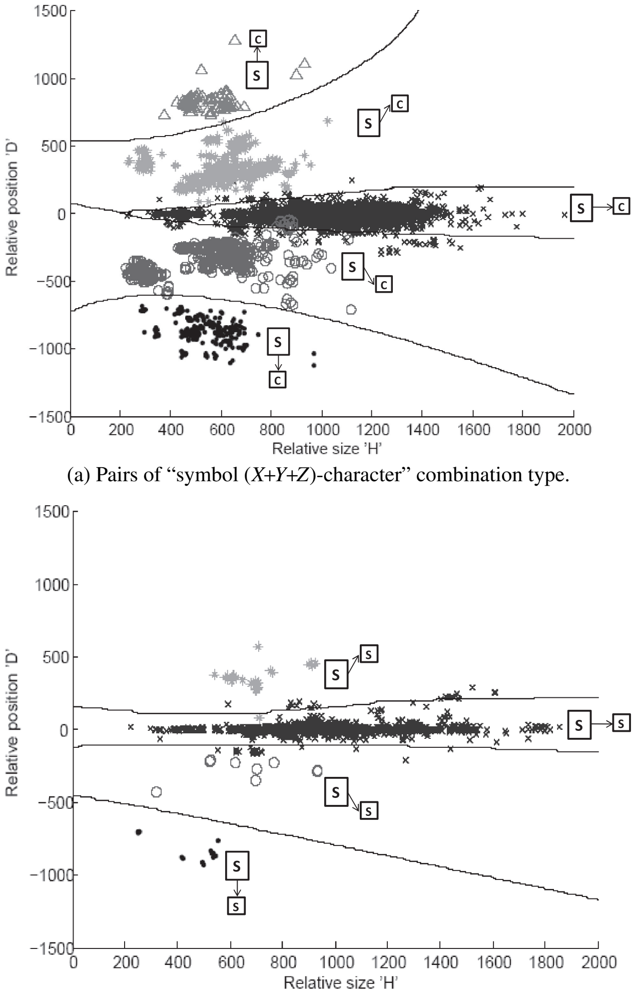
Fig. 9 Distribution maps without symbol type specifications.

There are, however, still small overlaps. A closer investigation of the points in the overlaps on Fig. 10 (a) revealed that, (i) most of them are pairs including irregular symbols and and/or irregular characters, (ii) other pairs have very big variations in the heights of their rectangular boxes. For examples, some parentheses and big symbols are very high and thus their relative size H deviated from usual distribution.

Figure 11 shows some example of misclassified symbols in Fig. 10 (b). In the first expression, the non-baseline parentheses of “(n)” have very low position and the pair “)$^($” was miss-classified to the horizontal class instead of the superscript class. In the second expression, the symbol “$\int$” has a very big X-part and thus the pair “= $\int$” was miss-classified to the superscript class instead of the horizontal class. In the third expression, the symbol “$\subsetneq$” has a very big Z-part and thus the pair “) $\subsetneq$” was miss-classified to the subscript class instead of the horizontal class.

7.3 Recognition Accuracy

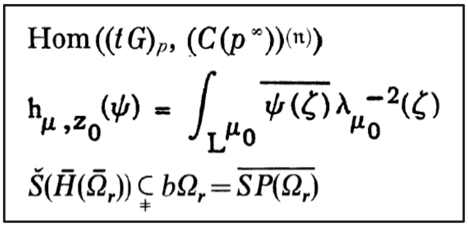
In order to evaluate the proposed methods, we employed two nonlinear classifiers; Bayesian classifier and SVM classifier. In these classifiers, the evaluation was done by using all the



(a) Pairs of "symbol (*X*+*Y*+*Z*)-character" combination type.

(b) Pairs of "symbol (*X*+*Y*+*Z*)-symbol(*X*+*Y*+*Z*)" combination type.

Fig. 10 Distribution maps with symbol type specifications.



$$\mathrm{Hom}((tG)_p, (C(p^{\infty}))^{(n)})$$

$$h_{\mu,z_0}(\psi) = \int_{L\mu_0} \overline{\psi(\zeta)}\,\lambda_{\mu_0}^{-2}(\zeta)$$

$$\check{S}(\bar{H}(\bar{\Omega}_r)) \subsetneq b\Omega_r = \overline{SP(\Omega_r)}$$

Fig. 11 Examples of misclassified symbols.

data for train and all the data for test[†].

7.3.1 Bayesian Classifier

Table 3 Classification accuracy (%) without discriminating combination type and symbol type for Bayesian classifier.

No. of symbol types	Document-dependent processing		Char-char combination	
	irregular treatment	private *X*:*Y*:*Z* ratio	excluded	included
1	−	−	98.651	98.482
1	−	+	98.585	98.424
1	+	−	98.677	98.532
1	+	+	98.635	98.572

Table 4 Classification accuracy (%) with discriminating combination type for Bayesian classifier.

No. of symbol types	Document-dependent processing		Char-char combination	
	irregular treatment	private *X*:*Y*:*Z* ratio	excluded	included
1	−	−	98.620	98.879
1	−	+	98.553	98.823
1	+	−	98.743	98.972
1	+	+	98.755	99.013
6	−	−	98.873	99.072
6	−	+	98.874	99.068
6	+	−	99.275	99.378
6	+	+	99.426	99.525

Table 3 shows the recognition rates without discriminating the combination types and symbol types, i.e., the recognition rates without any treatment on symbols. (They can be also considered as the recognition rates by the past method [14] applied not only for characters but also symbols.) The recognition rates are not high in this case. This result simply indicates that the discrimination task is not an easy task and the individual treatment on the combination type is important for dealing with all the adjacent pairs in mathematical expressions.

Table 4 shows the accuracy rate with discriminating the combinations when the four combinations of character/symbol pairs (e.g., "character parent" and "symbol child") are treated separately and discriminated on different distribution maps. Firstly, the comparison between Tables 3 and 4 confirms the above consideration that the combination type should be introduced for higher accuracy. Secondly and more importantly, Table 4 shows the remarkable importance of the symbol type specification. When the six symbol types were specified, the recognition rates have been improved. Although the number of the distribution maps is increased by the symbol types specification, the classification task became an easy problem.

From Table 4, we notice that, using symbol types improved the performance and we notice also the effect of applying document-dependent processing; that is, the use of the private *X*:*Y*:*Z* and the special treatment to irregular characters/symbols. These results prove the importance of using both symbol types and document-dependent processing in the classification task.

The highest recognition rate was 99.525%; this rate indicates that we can classify the spatial relationship between each adjacent pair almost perfectly if we have correct character/symbol recognition results. Even if the adjacent pairs of the "character-character" combination were excluded for

[†] We also used 6-fold cross-validation method in the evaluations. However, the two results did not have big differences. Both of them emphasize the importance of symbol types and document-dependent processing. The results of using 6-fold cross-validation using Bayesian classifier will be shown in Appendix.

Table 5 Classification accuracy (%) with discriminating combination type for SVM classifier.

No. of symbol types	Document-dependent processing		Char-char combination	
	irregular treatment	private X:Y:Z ratio	excluded	included
1	−	−	99.245	99.400
1	−	+	99.238	99.391
1	+	−	99.297	99.431
1	+	+	99.284	99.429
6	−	−	99.804	99.827
6	−	+	99.763	99.793
6	+	−	99.867	99.868
6	+	+	99.839	99.853

evaluating the classification performance in a more severely experimental setup, the recognition rate still remains at a very high rate, 99.426%. These high rates will be satisfactory because the databases include symbols with large size variations, many symbols show document-dependent characteristics, and the classifier employed is just a simple quadratic classifier based on the Gaussian assumption.

7.3.2 SVM Classifier

Table 5 shows the accuracy rate by quadratic SVM classifier. We choosed the quadratic kernel for easier comparison with the quadratic Bayesian classifier of Sect. 7.3.1. Soft margin was introduced on training SVM. Its parameter was optimized experimentally to achieve the best accuracy. From this table, it is confirmed that symbol type classification is important for the SVM classifier as well as the Bayesian classifier. It also emphasizes the importance of applying document dependent-processing by using special treatment to irregular characters and symbols. However, the effect of using private $X : Y : Z$ is slightly minimized. This is due to the saturation of recognition accuracy.

These results also prove the importance of our features H and D. The classification almost perfectly as it reached to 99.853%. As discussed in Sect. 7.2, the remaining mis-classifications were mainly due to irregular symbols and irregular characters and variations larger than expected. Again, Fig. 11 shows several examples.

8. Conclusion

In this paper, the spatial relationships between each adjacent symbol pair of mathematical expressions is classified into one of five classes (horizontal class, subscript class, superscript class, upper class, and lower class) for realizing an accurate structure analysis module part of math OCR. In order to deal with many variations of symbols, specifications of both symbol types and character/symbol combination types, as well as document-dependent processing, were introduced in the classification.

The classification task result on very large databases was almost perfect as it reached to 99.525%. In addition, experimental results showed that symbol types and document-dependent processing have improved the performance while these two points were overlooked in the past attempts.

Future work will focus on the secondary use of the proposed classification method. Specifically, if the (H, D) feature of a certain adjacent pair deviates in its corresponding distribution map, there are two possibilities; one possibility is that the character/symbol recognition is failed and the other is that the pairing of two character/symbol is failed. Thus, by checking the deviation, we will be able to detect those failures for better performance of math OCR.

References

[1] K. Chan and D. Yeung, "Mathematical expression recognition: A survey," Int. J. Document Analysis and Recognition, vol.3, no.1, pp.3–15, 2000.

[2] R.H. Anderson, "Syntax-directed recognition of hand-printed two-dimensional mathematics," in Interactive Systems for Experimental Applied Mathematics, ed. M. Klerer and J. Reinfelds, pp.436–459, Academic Press, 1968.

[3] M. Okamoto and B. Miao, "Recognition of mathematical expressions by using the layout structure of symbols," Proc. 1st Int. Conf. Document Analysis and Recognition, pp.242–250, 1991.

[4] H. Twaakyondo and M. Okamoto, "Structure analysis and recognition of mathematical expressions," Proc. 3th Int. Conf. Document Analysis and Recognition, pp.430–437, 1995.

[5] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, and T. Kanahori, "INFTY- An integrated OCR system for mathematical documents," Proc. Int. Conf. ACM Symposium on Document Engineering, pp.95–104, 2003.

[6] U. Garain and B.B. Chaudhuri, "A syntactic approach for processing mathematical expressions in printed documents," Proc. Int. Conf. Pattern Recognition, vol.4, pp.523–526, 2000.

[7] U. Garain and B.B. Chaudhuri, "An approach for recognition and interpretation of mathematical expressions in printed documents," Int. J. Pattern Analysis and Applications, vol.3, pp.120–131, 2000.

[8] R. Zanibbi, D. Blostein, and J.R. Cordy, "Recognizing mathematical expressions using tree transformation," Int. J. Pattern Analysis and Machine Intelligence, vol.24, no.11, pp.1455–1467, 2002.

[9] J. Ha, R.M. Haralick, and I.T. Phillips, "Understanding mathematical expressions from document images," Proc. 3rd Int. Conf. Document Analysis and Recognition, vol.2, pp.956–959, 1995.

[10] J.-Y. Toumit, S. Garcia-Salicetti, and H. Emptoz, "A hierarchical and recursive model of mathematical expressions for automatic reading of mathematical documents," Proc. 5th Int. Conf. Document Analysis and Recognition, pp.119–122, 1999.

[11] Y. Guo, L. Huang, C. Liu, and X. Jiang, "An automatic mathematical expression understanding system," Proc. 9th Int. Conf. Document Analysis and Recognition, vol.2, pp.719–723, 2007.

[12] H.J. Lee and M.C. Lee, "Understanding mathematical expressions using procedure-oriented transformation," Int. J. Pattern Recognition, vol.27, no.3, pp.447–457, 1994.

[13] D. Blostein and A. Grbavec, "Recognition of mathematical notation," in Handbook of Character Recognition and Document Image Analysis, World Scientific, pp.557–582, 1997.

[14] A. Walaa, S. Uchida, and M. Suzuki, "A large-scale analysis of mathematical expressions for an accurate understanding of their structure," Proc. 8th Int. Document Analysis Systems, pp.549–565, 2008.

[15] M. Suzuki, S. Uchida, and A. Nomura, "A ground-truthed mathematical character and symbol image database," Proc. 8th Int. Conf. Document Analysis and Recognition, pp.675–679, 2005.

[16] S. Uchida, A. Nomura, and M. Suzuki, "Quantitative analysis of mathematical documents," Int. J. Document Analysis and Recognition, vol.7, no.4, pp.211–218, 2005.

[17] M. Suzuki, C. Malon, and S. Uchida, "Databases of mathematical documents," Research Reports on Information Science and Electrical Engineering of Kyushu University, vol.12, no.1, pp.7–14, 2007.

[18] U. Garain and B.B. Chaudhuri, "A corpus for OCR research on mathematical expressions," Int. J. Document Analysis and Recognition, vol.7, no.4, pp.241–259, 2005.

Appendix: Evaluation by Cross-Validation

Tables A· 1 and A· 2 show the accuracy rate for the Bayesian classifier by using the 6-Fold cross-validation method. These results are comparable to the results shown in Tables 3 and 4, respectively.

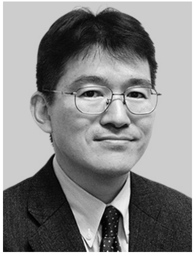
Table A· 1 Classification accuracy (%) without discriminating combination type and symbol type for Bayesian classifier using 6-fold cross-validation.

No. of symbol types	Document-dependent processing irregular treatment	 private *X*:*Y*:*Z* ratio	
1	−	−	98.195
1	−	+	98.149
1	+	−	98.223
1	+	+	98.303

Table A· 2 Classification accuracy (%) with discriminating combination type for Bayesian classifier using 6-fold cross-validation.

No. of symbol types	Document-dependent processing irregular treatment	 private *X*:*Y*:*Z* ratio	
1	−	−	98.654
1	−	+	98.613
1	+	−	98.740
1	+	+	98.793
6	−	−	98.874
6	−	+	98.854
6	+	−	99.225
6	+	+	99.366

Walaa Aly received her B.Sc. and M.Sc. degrees in electrical engineering from South Valley University, Aswan, Egypt, in 2000, and 2006, respectively. Since 2000 she has been associated with the Department of Electrical Engineering, Faculty of Engineering in Aswan, South Valley University as Demonstrator. Since 2007, she has been a Ph.D. student in the Department of Intelligent Systems of the Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan. Her research interests include pattern recognition, character recognition, and mathematical document analysis.



Seiichi Uchida received B.E., M.E., and Dr.Eng. degrees from Kyushu University in 1990, 1992 and 1999, respectively. From 1992 to 1996, he joined SECOM Co., Ltd., Tokyo, Japan where he worked on speech processing. Currently, he is a professor at Faculty of Information Science and Electrical Engineering, Kyushu University. His research interests include pattern recognition and image processing. He received 2002 IEICE PRMU Research Encouraging Award, MIRU2006 Nagao Award (best paper award), 2007 IAPR/ICDAR Best Paper Award, and 2009 IEICE Best Paper Award. Dr. Uchida is a member of IEEE and IPSJ.

Masakazu Suzuki received B.Sci. and M.Sci. degrees from Kyoto University in 1969 and 1971 respectively and degree of D. d'Eta és Sci. at Univ. Paris VII in 1977. During his career in CNRS from 1975 to 1977 and in Kyushu University from 1977, his main research subjects were complex analysis and algebraic geometry. He is currently a professor at Faculty of Mathematics, Kyushu University. In recent years, his research interests include mathematical document recognition and mathematical knowlege management. Dr. Suzuki is a member of MSJ, JSSAC, JSIAM, and IPSJ.