Math.comput.sci. *Online First*

DOI 10.1007/s11786-008-0051-9

Mathematics in Computer Science

Identifying Subscripts and Superscripts in Mathematical Documents

Walaa Aly, Seiichi Uchida, and Masakazu Suzuki

Abstract. In mathematical OCR, it is necessary to analyze two-dimensional structures of the component characters and symbols in mathematical expressions printed in scientific documents. In this paper, we analyze the positional relationships between adjacent characters for the purpose of automatic discrimination between baseline characters, subscripts, and superscripts, which is one of the most important and delicate parts of structure analysis. It has been proven through a large-scale experiment, that this discrimination can be carried out almost perfectly ($\sim 99.89\%$) by using the relative size and position of adjacent characters.

Mathematics Subject Classification (2000). ???

Keywords. Mathematical documents, structure analysis of mathematical expression, subscript and superscript.

Do you want to provide data of Math. Classification?
The runningtitle is too long. Please shorten it.

1. Introduction

Mathematical (hereafter, simply called math) OCR is a system for converting scanned page images into machine-editable text formats. Recently, there have been many attempts to implement math OCR to reduce the storage size of math documents and provide various search services. Math OCR is also essential for digital libraries. An overview of previous attempts is given in [1].

As shown in Figure 1, a main module of math OCR is math expression recognition. This module is further decomposed into two modules: component character recognition and structure analysis. Recognition of the component characters of a math expression is generally more difficult than ordinary character recognition, since there is a huge number of categories of math symbols. In fact, Suzuki et al. [3] have predefined about 1,500 categories for describing their math document databases. In addition, there are many similar-shaped categories. For example, it

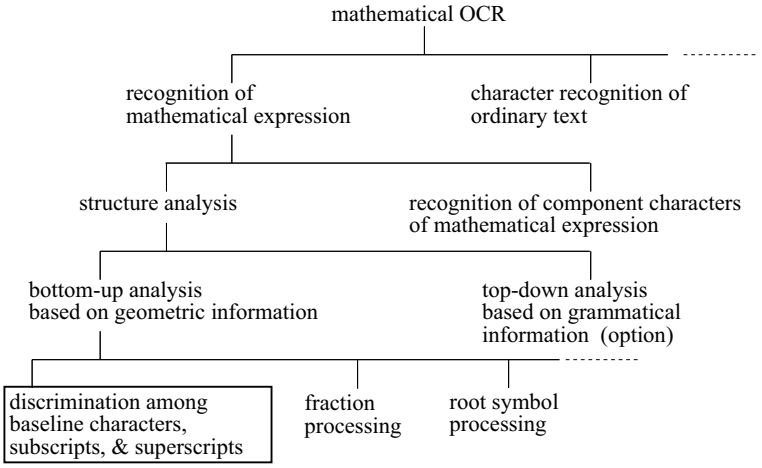


FIGURE 1. Major modules of math OCR.

is necessary to distinguish very similar symbols "v", "v" (italic "v"), "$\boldsymbol{v}$" (bold-italic "v"), "$\mathcal{V}$" (calligraphic "V"), "υ" (upsilon), "ν" (nu), and "$\vee$" (logical OR operator). Furthermore, the variation in symbol size is also a problem since the variation often changes the shape of the symbols slightly. Accordingly, recognition of the component characters is still a challenging and interesting pattern recognition problem, although not in the scope of this paper.

The other module, structure analysis, is more peculiar to math OCR. The aim of this module is to analyze the two-dimensional structure of math expressions. For example, subscripts and superscripts are detected and related to their parent characters in this module. The structure of math expressions is often represented by some graphical model, such as a directed graph or even an undirected graph. The nodes of the graph correspond to individual characters, while the edges (also referred to as "links") correspond to the parent-child relationships between adjacent character pairs.

The structure analysis module is often further decomposed into top-down analysis based on grammatical information and bottom-up analysis based on geometric information. Top-down analysis has been the subject of study since Anderson's famous trial four decades ago [4]. It is a post-process for verifying the result of bottom-up analysis, which is described below. Thus, top-down analysis is optional. Many researchers have, however, been fascinated by this topic because of its powerful ability to regulate the structure analysis module by some formal grammar representing the rules of math expressions. For further discussion on top-down analysis, readers can refer to past attempts in [1, 5, 6].

Bottom-up analysis, which is the main topic of this paper, is indispensable to structure analysis. Its role is to establish the parent-child relations among the component characters based on their geometric information, such as position and size. That is, bottom-up analysis is necessary to build the graph representation.

In this paper, we address one of the most important and delicate problems in bottom-up analysis, the automatic discrimination between baseline characters, subscripts, and superscripts (hereafter, simply called baseline analysis) using geometric information. It might seem that discrimination can be done by simply checking upper-lower relationships between adjacent character pairs. Unfortunately, this simple check is hugely insufficient. In fact, the discrimination must be done with careful consideration of the variation in printing styles. Figure 2 depicts several superscripts and subscripts in various math expressions. These clearly show, for example, that the relative sizes between baseline characters and superscripts are different in these expressions.

As reviewed in Section 2, baseline analysis has in the past been based on certain heuristics, which seem to work well. However, these heuristics have not been evaluated precisely; that is, the evaluations in the literature have typically been done with respect to the total performance of the structure analysis module and have therefore, not evaluated the individual performance of the discrimination. In other words, these past attempts are not well-grounded.

The main contribution of this paper is to prove that discrimination can be performed almost perfectly ($\sim 99.89\%$) using two features, namely, *relative size* and *relative position* of adjacent character pairs. The usefulness of these features will be well-grounded by qualitative and quantitative evaluation through experiments on very large databases of math documents. In this paper, we concentrate on the evaluation of alphanumeric text in math expressions; thus excluding math symbols from the experiment. The evaluation results are, however, still meaningful because alphanumeric characters form the major part of math expressions and there are sufficient evaluation results to show the effect of the proposed discrimination strategy. An extended experiment will be presented elsewhere.

Several techniques are introduced in the proposed discrimination method. For example, a character size normalization scheme is introduced to increase robustness with respect to the wide variety of inherent character shapes. The normalization includes special treatment for characters whose sizes are not stable. A document-specific consideration is also introduced in the normalization for better discrimination performance.

The remainder of this paper is organized as follows. After a brief review in Section 2, the database used in this investigation is outlined in Section 3. In Section 4, we describe the task of baseline analysis to clarify our purpose and contribution. Next we provide details of the extraction of features for discrimination in Section 5. In Section 6, we present the characteristics of some unusual characters. In Section 7, experimental results from the large database are presented to prove, both qualitatively and quantitatively, that the discrimination can be done almost perfectly. Finally, our conclusions and future work are described in Section 8.

2. Related work

As noted above, many past attempts have tackled the problem of baseline analysis. Since the discrimination often requires several heuristics and is only a module of a math OCR system, some of the literature documenting previous attempts does not provide details on this topic (e.g., [2, 7, 8]).

Okamoto [9] and Tian [10] have checked the relative position and size of adjacent character pairs. This is a reasonable strategy for the discrimination task. A normalized bounding box, which is a bounding box with virtual ascender and/or descender and can stabilize the discrimination by suppressing the bad effect of character shape variation, is introduced in [11]. We make use of this normalized bounding box in the proposed method, as described in Section 5. Mitra et al. [12] also employ a normalization scheme for the discrimination. They have also defined a parameter table which is useful for the discrimination, but have, unfortunately, provided neither sufficient justification for the parameter table nor experimental results focused on the discrimination task. Pottier and Lavirotte [13] use an OCR system to deduce the relative size and position. They too have, however, provided neither details nor experimental results of the discrimination task.

Our proposed method was inspired by Eto et al. [14], who tackle the discrimination task using an instance-based strategy; they collect adjacent character pairs and plot their relative positions and sizes on a two-dimensional space, called a *distribution map*. While it seems a reasonable strategy for the discrimination task, the use of a distribution map is still not well-grounded in any qualitative or quantitative evaluation. In addition, their discrimination boundaries are drawn as polygons designed manually.

Stated again, the main contribution of this paper is to provide a solid grounding to the fact that the baseline analysis can be done almost perfectly by using only the relative position and size of adjacent characters. This fact is proven by qualitative and quantitative evaluation through experiments with 41,581 adjacent character pairs as discussed below.

3. Database

To observe the ability of baseline analysis, we used 41,581 pairs of adjacent alphanumeric characters in math expressions. This huge number of characters was extracted from two large-scale databases, InftyCDB-1 [3, 15] and InftyCDB-2 [16], which together consist of 65 English articles (published between 1949 $\sim$ 2000), 4 French articles (published between 1974 $\sim$ 1988), and 7 German articles (published between 1956 $\sim$ 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 research on math document recognition. In fact, they are larger than the database used in [17], which consists of 297 pages. Such large databases are

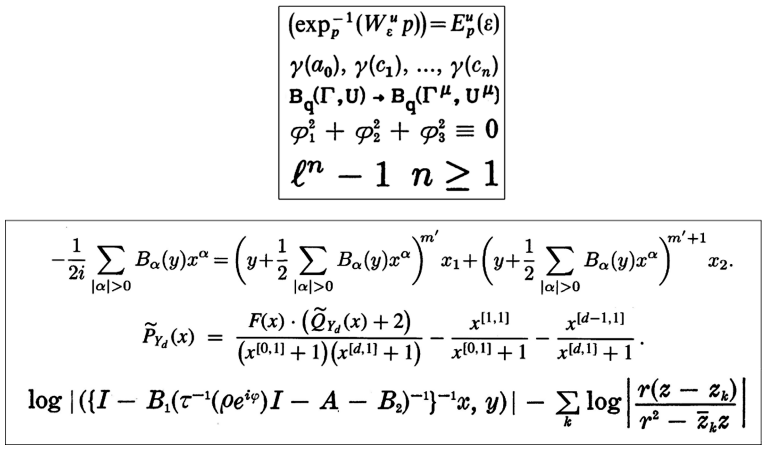


FIGURE 2. Examples of math expressions. Upper: simple expressions. Lower: complex and/or large expressions.

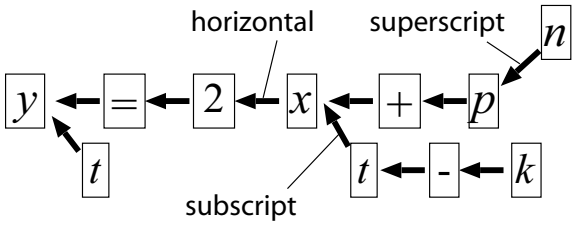


FIGURE 3. Links representing the structure of the expression "$y_t = 2x_{t-k} + p^n$".

extremely well suited to deriving universal properties (e.g., baseline analysis) of math expressions.

All the pages of InftyCDB-1 and InftyCDB-2 were scanned in 600 dpi and automatically digitized by the same commercial scanner (RICOH Imagio Neo 450). The quality of the resulting page images varies according to the quality of the original prints and/or copies. The math expressions shown in Figure 2 are examples from the databases.

A ground truth was *manually* attached to each character in the math expressions by seven university mathematics students. The ground truth of each character consists of the following attributes [3]:

- character category which denotes the finest level of character classification. For example the operator type consists of 92 predefined categories such as $\div$, $\times$, and $\pm$.

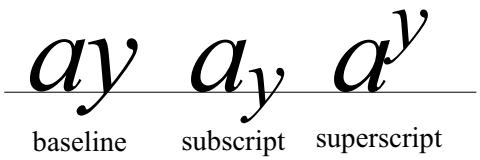


FIGURE 4. The discrimination task addressed in this paper. Since we assume that the parent character lies on the baseline, the task is to discriminate whether the child character also lies on the baseline or in the subscript or superscript level.

- type which denotes a set of categories having a similar property such as Roman alphabet, Greek alphabet, numeral, operator, and parenthesis.
- font such as upright, italic, bold, and bold italic.
- text area or math expression area.
- quality (normal, touching, broken, etc.).
- size (height and width).
- location in the entire math expression image.
- link to parent character.
- sub/super-script level.

One of the most important attributes for this work is the eighth attribute, namely, the link, which represents the positional relation between a pair of adjacent characters. Hereafter, the first character of each pair is called the parent character and the second character the child character. Link relations encode the structure of each math expression as a tree. Figure 3 shows a math expression whose structure is represented by seven horizontal links and three non-horizontal links. The non-horizontal links indicate subscripts and superscripts.

In the following experiment, we assume that we already know the correct class of each character. This correct class is given by the ground truth. This is a reasonable assumption as symbol recognition is, in fact, performed in advance of structure analysis in most math OCRs.

4. The task

In this paper, the task of baseline analysis is somewhat simplified by imposing the following two conditions.

- The parent of each adjacent character pair lies on the baseline. In other words, the parent character is neither a subscript nor a superscript. Under this condition, our task is to judge whether the child is a baseline character or a (first-level) subscript or (first-level) superscript. Figure 4 shows the

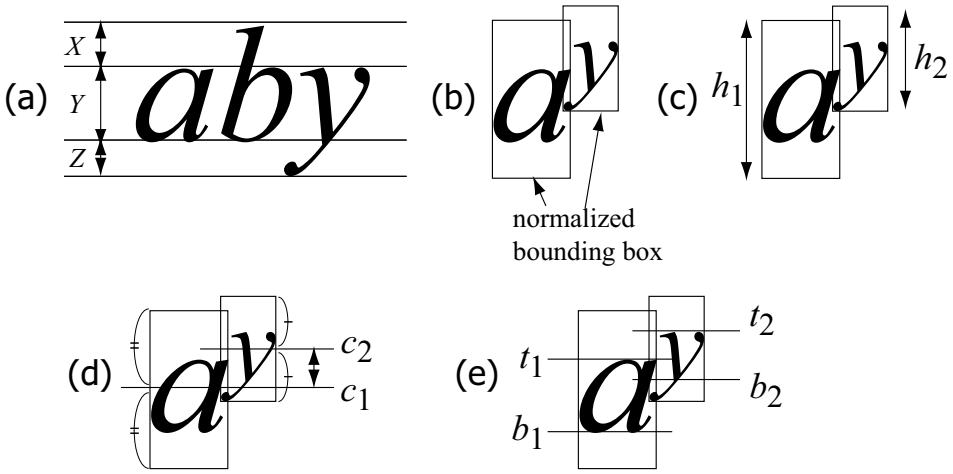


FIGURE 5. (a) X, Y and Z regions. (b) Normalized bounding box. (c) Normalized size (h_1 and h_2). (d) Normalized center (c_1 and c_2). (e) Baselines (b_1 and b_2) and top-lines (t_1 and t_2).

three situations to be discriminated. Note that first-level denominators and numerators are also considered as baseline characters.

- The target characters are alphanumeric e.g., "A", "B", "c", "δ", "$\boldsymbol{E}$", "$\mathcal{F}$", "$\mathfrak{G}$", "$\mathbb{H}$", and "2". For example, "$\int_k$" is excluded from the present investigation due to the symbol "$\int$". This condition is not that critical, because approximately 57.1% of the characters in math expressions are alphanumeric (179,457 alpha-numerals in the total 314,114 characters). Alpha-numerals are, therefore, in the majority and an investigation of their discrimination is still very meaningful in itself.

As noted before, there are 41,581 adjacent character pairs satisfying these conditions in the databases, including 21,718 base–sub, 13,627 base–base, and 6,236 base–sup points.

Consequently, our aim is to prove experimentally the possibility of near-perfect discrimination of a child's location (either the baseline, subscript or superscript level) by using positional and size relations between the parent on the baseline and a child.

5. Feature extraction for discrimination

5.1. Estimation of partial heights of characters

The estimation of the partial heights X, Y and Z are crucial for the discrimination task. This can be done by calculating X, Y and Z for every baseline character

of a document while referring to its category information (i.e., recognition result). The median or mode of the calculated X, Y and Z will be a good estimate of the partial heights of the document.

It is also necessary to estimate the partial heights of subscripts and superscripts in addition to the baseline characters. This requirement might seem unsatisfiable because we need subscripts and superscripts to estimate the partial heights X, Y and Z in advance of the baseline analysis. However, it is fairly easy to discriminate subscripts and superscripts "roughly" from the baseline characters in the math expressions without using the proposed discrimination method. Of course, the result of the rough discrimination includes baseline characters wrongly discriminated as sub/superscripts. We can, however, estimate X, Y and Z accurately for subscripts and superscripts from the rough discrimination results by taking the median or mode.

5.2. Normalized bounding box

It seems reasonable to use the bounding box size (vertical size) and center positions to apply baseline analysis. However, these sizes and positions vary depending on the font style and character category. For example, the bounding box sizes of "a" and "A" are very different.

Consequently we consider the *normalized bounding box* instead of the actual bounding box of each character. The difference between the two bounding boxes is that the former is defined as the total size of the character including a virtual ascender X and virtual descender Z. For example, the normalized bounding box of a character with neither an ascender nor descender (e.g., "a", "c", "e") is defined by attaching a virtual ascender and descender to the actual bounding box. Similarly, the normalized bounding box of a character without a descender (e.g., "b", "d", "h") is defined by attaching a virtual descender.

5.3. Feature extraction

From the normalized bounding box, a normalized size and a normalized center are calculated for each character. Figures 5 (c) and (d) illustrate these. The normalized size and center are defined as the height and the center of the normalized bounding box, respectively. The relative size H and relative position D of a pair of adjacent characters are then calculated from the normalized size and center line. Let h_1 and h_2 denote the normalized sizes of the first and second characters, respectively. Similarly, let c_1 and c_2 denote the normalized centers of these characters. Then, H and D can be calculated as follows:

$$H = \frac{h_2}{h_1}\,, \tag{5.1}$$

$$D = \frac{c_1 - c_2}{h_1}\,. \tag{5.2}$$

The ability to discriminate the two features, H and D, is confirmed by a large-scale experiment in Section 7.

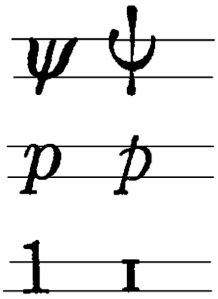


FIGURE 6. Irregular characters.

6. Irregular characters

In the above normalization process, care must be taken with irregular characters that have different sizes and occupy different X, Y and Z regions in different documents. From a cursory observation, we have detected 18 categories whose sizes and $X : Y : Z$ regions differ in various documents. These categories include 7 Roman characters (e.g., "i", "t"), 3 Greek characters (e.g., "ϕ", "ψ"), and 8 numeric characters (e.g., "0", "5"). Figure 6 depicts several irregular characters. For example, "ψ" occupies only Y and Z regions in some documents, yet occupies the entire $X : Y : Z$ region in others and therefore it will have two different cases.

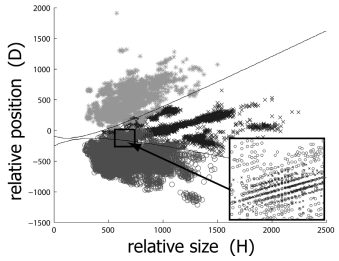
Clearly, such irregular characters seriously affect the estimation of the heights X, Y and Z. The situation becomes worse if we rely on the heights X, Y and Z estimated for each individual document. If a document includes many irregular characters, the estimation of X, Y and Z will be distorted by the irregular characters. Note that the presence of irregular characters from only a single character category is often rare in a document; in fact, the number of occurrences of the character ψ is not large. The total number of irregular characters, however, can be large because there are several categories which produce irregular characters. Consequently, the detrimental effect of irregular characters is noticeable.

In the following experiments, we will emphasize that special treatment of irregular characters is important for better discrimination performance. Our method for this special treatment is described in Section 7.

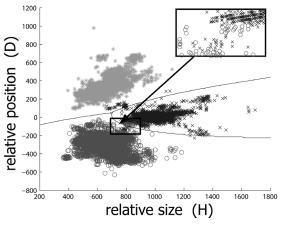
7. Evaluation of discrimination ability

7.1. Qualitative evaluation with distribution maps

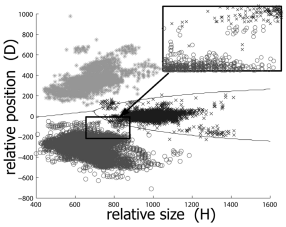
The ability of baseline analysis was evaluated qualitatively by observing the distribution map [14], that gives the distribution of (H, D)-features of adjacent character



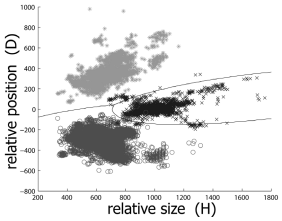
(a) Without any normalization.



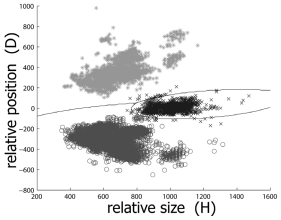
(b) Normalization with common $X:Y:Z$.



(c) Normalization with common $X:Y:Z$ and special treatment of the irregular characters.



(d) Normalization with private $X:Y:Z$.



(e) Normalization with private $X:Y:Z$ and special treatment of the irregular characters.

FIGURE 7. Distribution maps for different cases. The curves show the decision boundaries using a quadratic classifier. The values of H and D are multiplied by 1000.

pairs in the database. The smaller the overlap in the baseline analysis, the better is the discrimination using (H, D)-features and an appropriate discrimination function.

Figure 7 shows several distribution maps, whose details are discussed later. In the distribution maps, each "$\times$"-shaped dot corresponds to a baseline character pair which we call the baseline characters, such as "xy" and "2α". Each "$\circ$"-shaped dot corresponds to a pair consisting of a baseline character and a subscript, which we call subscript characters, such as "$\boldsymbol{M}_2$" and "ϵ_X". Finally, each "$*$"-shaped dot corresponds to a pair consisting of a baseline character and a superscript, which we call superscript characters, such as "$\mathbb{H}^3$."

Figure 7 (a) shows the distribution map when a normalized bounding box is not used. It is obvious that there are heavy overlaps between the three distributions and therefore we cannot distinguish them. Baseline characters and subscripts are particularly confused on the map. These overlaps stem from the size and positional variations of the bounding boxes without normalization.

Figure 7 (b) shows the distribution map using a normalized bounding box. The overlap is less than that in Figure 7 (a). Accordingly, we conclude that the normalization is very effective and the discrimination can be done well using (H, D)-features. However, a small overlap can still be observed. The character pairs in this overlap often include irregular characters. The obvious hypothesis is that normalization of these irregular characters is incorrect, causing their (H, D)-features to vary.

Figure 7 (c) verifies this hypothesis. This figure depicts the distribution map after special treatment of irregular characters. The special treatment comprises (i) removal of an irregular character when estimating the heights X, Y and Z, (ii) estimation of true $X : Y : Z$ occupation of the irregular character by choosing the best case according to its size from the possible cases of the category and (iii) setting the normalized bounding box according to the estimation (i.e., the possible $X : Y : Z$ occupation). The effect of the special treatment is significant as detailed below.

In the two distribution maps given in Figures 7 (b) and (c), the heights X, Y and Z are estimated by averaging all the heights estimated for the individual documents. Thus, a common $X : Y : Z$ is used in the normalization. Now, a question arises: Does the overlap decrease if a "private" $X : Y : Z$ for each individual document is used instead of the common $X : Y : Z$? The answer is that it depends on the special treatment of irregular characters. Without any special treatment, the private $X : Y : Z$ values of several documents were adversely affected by the irregular characters in those documents. Figure 7 (d) shows the map with a private $X : Y : Z$ and without any special treatment of the irregular characters. The overlap is slightly larger than in (b). In contrast, with special treatment, the private $X : Y : Z$ values of several documents were not affected by the irregular characters. On the contrary, an estimated private $X : Y : Z$ is better in that it incorporates document-specific characteristics. (That is, there is

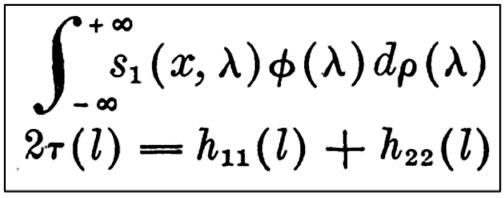


FIGURE 8. The characters ρ and τ which cannot be discriminated correctly.

TABLE 1. Error rate (%) using a quadratic classifier in H-D space. The parenthesized number is the rate for D' instead of D.

normalize	-	common $X:Y:Z$				private $X:Y:Z$			
special treat. of irreg. chars.	-	not applied		applied		not applied		applied	
base–base& base–sub	6.74	0.24	(0.32)	0.10	(0.16)	0.41	(0.44)	∼0	(0.12)
base–base& base–sup	0.67	0.35	(0.37)	0.29	(0.33)	0.35	(0.39)	0.13	(0.11)
base–sub& base–sup	∼0	0	(∼0)	0	(0)	0	(∼0)	0	(0)
all	6.05	0.32	(0.42)	0.21	(0.27)	0.43	(0.50)	0.11	(0.13)

only a small difference in $X:Y:Z$ among documents. In fact, the overlap in (e) is smaller than in (c) and we are finally able to suppress overlaps almost completely.

Several instances of ρ, π and τ in an old document (published in 1946) still appear confused in the distribution map. Two of these characters are shown in Figure 8. In the first equation, the character "ρ" is the subscript to "d" and "1" is the subscript to "s". Generally, the character "ρ" occupies Y and Z regions whereas "1" occupies X and Y regions. Thus, the top of "ρ" should be lower than that of "1". In this equation, however, the general relation does not hold. Consequently, all the subscripts of "ρ" in the document are closer to baseline characters than subscripts and therefore they form a small overlap.

7.2. Quantitative evaluation through quadratic discrimination

Table 1 shows the discrimination error rates using a simple Bayesian classifier. The classifier was trained by assuming that each of three classes (baseline characters, subscripts, and superscripts) forms a two-dimensional Gaussian distribution in the $H-D$ space. It is well-known that under this assumption the Bayesian classifier is reduced to a quadratic classifier. The parameters of the Gaussian distribution, i.e., a mean vector and a 2×2 covariance matrix, were estimated empirically using

all the data of each class. Figure 7 shows the discrimination boundary using the quadratic classifier. The error rate was evaluated in a closed manner; i.e., the discrimination test was done with the same data that were used in the estimation.

The evaluation results listed in the table coincide with the qualitative evaluation in Subsection 7.1. The lowest discrimination error was attained for the distribution map using the normalized bounding box, special treatment of irregular characters, and a private $X : Y : Z$. Although the discrimination boundary was limited to a simple quadratic function, the lowest error was 0.11% (i.e., 99.89% accuracy). In other words, we were able to apply baseline analysis almost perfectly ($\sim$ 99.89%) using the two features, H and D.

A supplementary experiment was conducted with another relative position feature defined as

$$W = \sqrt{(t_1 - t_2)^2 + (b_1 - b_2)^2}/h_1 \tag{7.1}$$

$$D' = \begin{cases} -W & \text{if } t_2 > t_1, \\ W & \text{otherwise} \end{cases} \tag{7.2}$$

where t_1 and t_2 are the top-lines and b_1 and b_2 are the baselines as shown in Figure 5(e). The error rates achieved for this D' are listed in Table 1 as parenthesized numbers. A slight degradation compared to the original D can be observed.

8. Conclusion

Through a large-scale experiment using 41,581 pairs of adjacent alphanumeric characters in math expressions, we have proved that baseline analysis can be done almost perfectly (99.89%). For this discrimination, relative position and size features, calculated after a normalization process, are used. Although these features are not themselves novel, this paper is the first attempt to prove their effective performance through a well-focused and large-scale experiment. It is somewhat surprising that such high-performance discrimination can be expected for *any* math document; that is, the simple quadratic discrimination boundary derived in the experiment will still be effective on other math documents, because the boundary is very reliable across the various math expressions in the large database.

Results of the discrimination experiment reveal that special treatment of irregular characters can improve the discrimination performance. The results also reveal that the use of character size normalization specialized for each individual document is effective. Although the improvement using this "private" size normalization is not that large (99.79%→ 99.89%), document-specific characteristics can be considered a key factor in achieving perfect discrimination.

Future work will focus on the following aspects.

- Experimental evaluation of non-alphanumeric symbols, such as operators and parentheses: Although most symbols in math expressions are alphanumeric as noted before, non-alphanumeric symbols are also important in math expressions. Their discrimination might be more difficult than the discrimination

of alpha-numerals because there is no general size normalization scheme for non-alpha-numerals (especially for parentheses). However, if we can extract document-specific characteristics from each individual document, these may help the discrimination of non-alpha-numerals.

- Experimental evaluation of adjacent characters of non-baseline parents: For example, the discrimination of the relations between "a" and "y" in "x^{ay}" and "x^{a^y}", where a is a non-baseline parent.
- Secondary utilization of the distribution maps: We already know the distribution of baseline characters, subscripts, and superscripts on the distribution maps. If the features (H and D) of a pair of adjacent characters do not lie in the right position of the distribution, it is possible to hypothesize the existence of some error about these characters (e.g., category error, pairing error, broken or touching characters, etc.). From these misplaced characters our technique may be able to help text line segmentation.

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] D. Blostein, and A. Grbavec, "Recognition of mathematical notation", In Handbook of Character Recognition and Document Image Analysis, pp. 557–582, 1997.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] X. Tian, and H. Fan, "Structural analysis based on baseline in printed mathematical expressions", Proc. 6th Int. Conf. Parallel and Distributed Computing Applications and Technologies, pp. 787-790, 2005.

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

[12] J. Mitra, U. Garain, B. B. Chaudhuri, K. Swamy, and T. Pal, "Automatic understanding of structures in printed mathematical expressions", Proc. 7th Int. Conf. Document Analysis and Recognition, pp. 540–544, 2003.

[13] S. Lavirotte, and L. Pottier, "Optical Formula Recognition", Proc. 4th Int. Conf. Document Analysis and Recognition, pp. 357–361, 1997.

[14] Y. Eto and M. Suzuki, "Mathematical formula recognition using virtual link network", Proc. 6th Int. Conf. Document Analysis and Recognition, pp. 762–767, 2001.

[15] 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.

[16] 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.

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

Walaa Aly
Graduate School of Information Science and Electrical Engineering
Kyushu University
744 Motooka, Nishi-ku
Fukuoka-shi, 819-0395
Japan
e-mail: walaa_ata@yahoo.com

Seiichi Uchida
Faculty of Information Science and Electrical Engineering
Kyushu University
744 Motooka, Nishi-ku
Fukuoka-shi, 819-0395
Japan
e-mail: uchida@is.kyushu-u.ac.jp

Masakazu Suzuki
Faculty of Mathematics
Kyushu University
6-10-1 Hakozaki, Higashi-ku
Fukuoka-shi, 812-8581
Japan
e-mail: suzuki@math.kyushu-u.ac.jp

Received: December 15, 2007.
Revised: March 14, 2008.
Accepted: June 15, 2008.