A Large-Scale Analysis of Mathematical Expressions for an Accurate Understanding of Their Structure

Walaa Aly, Seiichi Uchida, Masakazu Suzuki
Kyushu University, 744 Motooka, Nishi-ku, Fukuoka-shi, 819-0395, Japan
{walaa, uchida}@human.is.kyushu-u.ac.jp

Abstract

A wide variety of mathematical expressions printed in scientific and technical reports can be recognized by analyzing a two-dimensional layout structure. In this paper, the position relation between adjacent characters is analyzed for the purpose of automatic discrimination among baseline, subscript, and superscript characters. This analyzing is one of the most important parts of structure analysis. The proposed method is very promising, as the results reached up to (99.76%) over a very large database by using distribution map. This distribution map is defined by two important features, i.e., relative size *and* relative position.

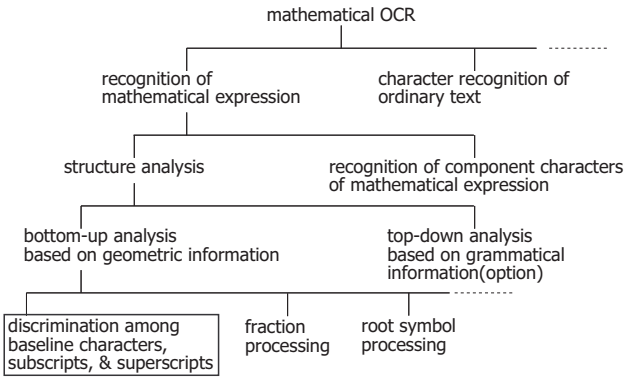


Figure 1. Major modules of math OCR.

1. Introduction

There is a wealth of mathematical expressions which can be very useful in many applications. These materials have been existed in physical, mechanical, and mathematical books, which are references in the domain from many years ago. Unfortunately, most of these materials are not available in electronic form and therefore there is an urgent need to build systems to manipulate them.

Math OCR is a system for converting scanned page images into machine-editable text formats, such as Latex and XML. There are many attempts to recognize mathematical documents. This recognition is very necessary for reducing the storage size, making various search services, and getting digital libraries. The previous attempts were overviewed in [1]. Figure 1 shows the major modules of math OCR. The main module of math OCR is the recognition of mathematical expressions, which is decomposed into two other modules; recognition of component characters of mathematical expressions and structure analysis.

Recognition of component characters of mathematical expressions is more difficult than that of ordinary text characters. This fact is due to many reasons; mathematical symbols and characters have huge numbers of categories. For example, Suzuki et al. [2] have defined about 1500 categories for mathematical symbols and characters. In addition, these categories have many similar shapes such as "v", "v" (italic "v"), "$\boldsymbol{v}$" (bold-italic "v"), "$\mathcal{V}$" (Calligraphic of "V"), "υ" (upsilon), "ν" (nu), and "$\vee$" (logical OR operator). Moreover, these categories have many variation in their sizes. Due to these reasons, the recognition of mathematical expressions became a difficult and interesting part in pattern recognition. However, this part is not the scope of this paper.

Structure analysis is a special module for math OCR. The analysis of a two-dimensional structure of mathematical expressions is done in this module. The structures of mathematical expressions are often represented by some graphical models, such as a directed graph or undirected graph. The nodes of the graph correspond to individual characters and the edges correspond to the relationships between adjacent characters. These relations are determined in this module, for example, subscripts and superscripts are detected in this module.

The structure analysis module is further decomposed into two modules; top-down and bottom-up module. The purpose of the top-down module is to regulate the analysis result by using some grammar. Many researchers have

been fascinated with the top-down module from Anderson's famous trial four decades ago [3]. This fascination may come from the fact this module is a practical application of two-dimensional grammars (such as tree grammar and web grammar). The details of this module can be found in [1, 4, 5].

Bottom-up module, which is the main topic of this paper, is also important in the structure analysis module. This module determines the relative structure of individual characters by using geometrical information, such as position and size. In some sense, bottom-up module is far more essential than the top-down module, because the latter can be considered as just a verification step of the former. In other words, the performance of the bottom-up module is very crucial for the total performance of structure analysis.

In this paper, we discuss about the main function of the bottom-up module, that is, the automatic discrimination among baseline, subscript, and superscript characters (hereafter, simply called *discrimination task*). Readers might think that the discrimination task can be easily completed by just checking the upper and lower relation between adjacent characters. This strategy, however, is totally insufficient. Careful considerations are necessary for this discrimination due to wide variation of printing style. Figure 2 shows some examples of mathematical expression from different mathematical documents. We can notice from this figure the variation of the relative sizes and the positions between baseline characters and sub/superscript characters. For example, the position of the subscript μ is very low and the superscripts 2 in Z^2 and $)^2$ are located at different positions.

As explained in Section 2, the discrimination task has been tackled with some heuristics. Unfortunately, these heuristics did not evaluated precisely. In fact, the previous attempts did not give details about this task and they only gave total performance of the structure analysis module. Furthermore, they illustrated neither qualitative nor quantitative analysis of their result. Therefore, previous attempts are not well-grounded.

The main contribution of this paper is to evaluate how the geometrical information is useful for the discrimination task through a very large-scale experiment and focused inspections about the experimental result. As the geometrical information, we will use two features called *relative size* and *relative position*. On the extraction of those features, careful considerations will be given for compensating the difference among various print styles. Experimental results revealed that the discrimination can be done almost perfectly ($\sim 99.76\%$).

In this paper, the discrimination task was done for the alphanumeric characters; mathematical symbols are excluded from this evaluation. The evaluation results, however, are still meaningful. That is, alphanumeric characters have the majority over mathematical symbols in mathematical expressions. Therefore, the results are enough to show the importance of the proposed method.

As noted above, careful considerations are given for dealing with this delicate discrimination task. For example, a character size normalization is introduced to avoid the variation of character shapes. In addition, a special treatment for characters whose sizes are not stable is introduced. A document- specific consideration is also introduced to improve the performance.

The remainder of this paper is organized as follows. Section 2 outlines a brief review on the discrimination task. Section 3 presents the database which was used in the investigation. Section 4 describes the task of the proposed method. Section 5 introduces the features which were used in the distribution map. Section 6 presents the characteristics of some unusual characters. Section 7 shows experimental results with a very large database through qualitatively and quantitatively analysis. Finally, Section 8 presents a concussion and future work.

2. Related work

As noted before, there have been many attempts which tackled the discrimination task. For example, Okamoto [6] and Tian [7] have checked the relative position and size of adjacent character pairs. It is a reasonable strategy for the discrimination task.

In [8], a normalized bounding box, which is also employed in the proposed method, was introduced. It is a bounding box with a virtual ascender and/or descender and can stabilize the discrimination by suppressing the bad effect of character shape variation. Mitra et al. [9] also have employed an normalization scheme on the discrimination. They have also provided a parameter table which will be useful for the discrimination. They, unfortunately, have provided neither enough justification on the parameter table nor experimental results focused on the discrimination task.

Pottier and Lavirotte [10] used OCR system to deduce the relative size and position. They also, unfortunately, have provided neither details nor experimental results focused on the discrimination task.

Again, all of these attempts did not give details about the discrimination task; they gave only the total performance of the system and specified neither quantitative nor qualitative analysis of their results (e.g., [11, 12, 13]). This may be because (i) the discrimination 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.

In contrast, this paper concentrates its attention on the discrimination task. This task is not trivial one; it requires many considerations about the characteristics of mathematical expressions, which often depends on the printing style

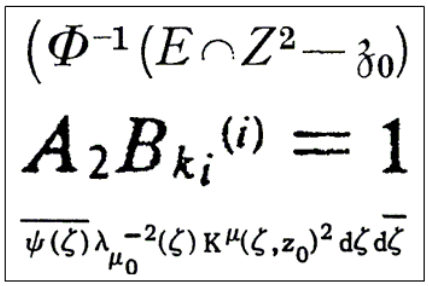


Figure 2. Examples of mathematical expressions.

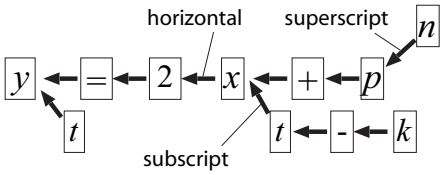


Figure 3. Links representing the structure of the expression "$y_t = x_{t-k} + p^n$."

of each document. In other words, through this task we can understand the characteristics, which will be useful in many other modules in math OCR. Experimental evaluation is done quantitatively and qualitatively with 47,963 ground-truthed adjacent character pairs extracted in mathematical expressions.

The proposed method was inspired by Eto et al. [14], They collected all adjacent pair of characters and plot their relative size and position on a two-dimension space. They, however, did not specify any qualitative or quantitative evaluation like other past attempts. In addition, their method relies on some empirical manual operation at the discrimination task.

3. Database

The discrimination task was applied on 47,963 pair of adjacent alphanumeric characters in mathematical expressions. This huge number of characters were extracted from two large databases, InftyCDB-1 [2, 15] and InftyCDB-2 [16]. These databases consist of 65 English articles (published in 1949 ~ 2000), 4 French articles (published in 1974 ~ 1988), and 7 German articles (published in 1956 ~ 1987) on pure mathematics. The total number of pages in the databases is 908.

To the authors' best knowledge, those databases are the largest databases used in past attempts on the discrimination task. For example, they are larger than the database used

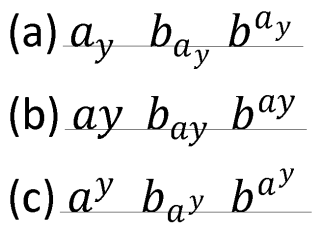


Figure 4. The three classes in the discrimination task: (a) subscript class, (b) horizontal class, and (c) superscript class. Here, the adjacent pair of "a" and "y" is illustrated.

in [17], which consists of 297 pages. Such large databases are very suitable to derive universal properties (e.g., the discrimination task) of mathematical expressions.

In InftyCDB-1 and InftyCDB-2, all pages were scanned in 600 dpi and binarized automatically by the same commercial scanner (RICOH Imagio Neo 450). The quality of the resulting page images varies with the quality of original print and/or copy. The mathematical expressions shown in Figure 2 were examples from the databases.

A ground truth for each character in the mathematical expressions was attached manually by seven university mathematics students. The ground truth of each character consists of the many attributes [2], such as character category, size, location, and link to its adjacent character.

From the link attribute, it is possible to represent each mathematical expressions as a tree. Figure 3 shows a structure mathematical expression, which has seven horizontal links and three non-horizontal links. Note that this is a directed tree where each link connects a *child* character to a parent character.

In the experimental evaluation, we select adjacent character pairs (i.e., linked character pairs) which are comprised of alphanumeric characters such as "A", "B", "c", "δ", "$\boldsymbol{E}$", "$\boldsymbol{f}$, and "2". Mathematical symbols are excluded such as"$\int$". This exclusion is not so vital because of the majority of alphanumeric. About 57.1% of characters in mathematical expressions are alphanumeric (179,457 alpha-numerals from 314,114). Therefore, the investigation of alphanumeric characters for the discrimination task is still meaningful by itself and useful for the discrimination task including mathematical symbols.

Note that we assume that we know the correct category of each character, which is taken from the ground truth. This assumption is reasonable since the structure analysis is performed after recognition of component characters in most math OCRs.

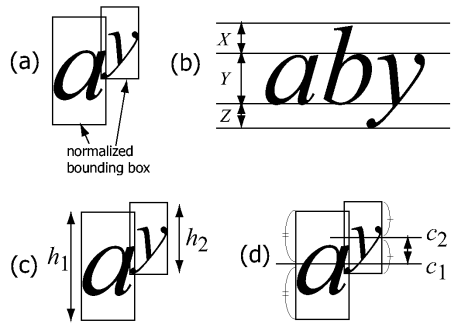


Figure 5. (a) X, Y and Z regions. (b) Normalized bounding box. (c) Normalized size (h_1 and h_2). (d) Normalized center (c_1 and c_2).

4. The discrimination task

Our task is the automatic discrimination among baseline, subscript, and superscript characters. This task is equivalent to the classification problem of each adjacent character pair into one of three classes, horizontal class, subscript class, and the superscript class. A character pair in the horizontal class are linked by a horizontal link (Fig. 3). Similarly, a pair in the subscript class are linked by a subscript link and a pair in the superscript class are linked by a superscript link.

Figure 4 illustrates these three classes by the parent character "*a*" and the child character "*y*". The numbers of pairs in all the 47,963 pairs are 16,896 (horizontal), 24,185 (subscript), and 6,882 (superscript). Note that a parent character may be a sub/superscript character. Among the 47,963 pairs, 4,900 pairs have sub/superscript parent characters. (Their children will be double sub/superscript characters.)

5. Feature extraction for the discrimination task

5.1. Normalized bounding box

For the extraction of two features, relative size and relative position, we must care about the difference of character sizes. For example, the sizes of "a" and "A" are very different. For compensating this difference, we will use a *normalized bounding box* for each character instead of the actual bounding box. Figure 5 (a) shows the normalized bounding box.

For setting the normalized bounding box of each character, a virtual ascender or a virtual descender or both are added to the actual bounding boxes. 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"), need the virtual ascender and descender. Similarly, the normalized bounding box of the characters without descender (e.g., "b", "d", "h") need the virtual descender.

5.2. $X : Y : Z$ ratio

On setting the normalized bounding box, we must know the height of the virtual ascender and descender. This can be derived from the ratio of three regions, called X, Y, and Z regions. Figure 5 (b) shows X, Y and Z regions. From the ratio of the heights of X, Y and Z, 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 "A" can be obtained by estimating the height of its virtual Z and it will be calculated by multiplying the actual height of the "A" by the ratio between 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 have the $X : Y : Z$ ratio for all the baseline characters of the document. The $X : Y : Z$ ratio for sub/superscript 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[1].

The $X : Y : Z$ ratio is common for all characters within a document. In other words, each document has its own $X : Y : Z$ ratio. In this sense, we hereafter call the ratio *private* $X : Y : Z$ *ratio*. In contrast, we also can estimate the $X : Y : Z$ ratio common for any document by using all the characters of a large-scale multi-document database. We hereafter call the ratio *common* $X : Y : Z$ *ratio*. If the performance with the common ratio is better or comparable to that with the private ratio, the common ratio is far more useful than the private ratio since we can always use the same ratio without any estimation at each document. Unfortunately, as experimentally shown in a later section, the private ratio outperforms the common ratio.

[1]Readers may be confused by the fact that we need to discriminate between baseline characters and sub/superscript characters for estimating their own $X : Y : Z$ ratio during the process toward our final goal, i.e., the discrimination among the baseline characters and the sub/superscript characters. For the estimation, we only need to perform a "rough" discrimination. Of course, the result from this rough discrimination includes some errors. These errors do not affect the estimation seriously because we use the average of the heights.

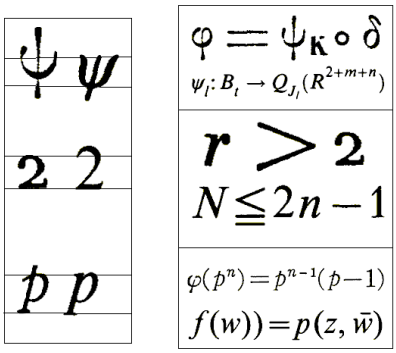


Figure 6. Examples of irregular characters. The characters on the left part were extracted from the mathematical expressions on the right part.

5.3. Feature extraction

The two features used for the discrimination task are defined by a normalized size and a normalized center of the normalized bounding box. The normalized size and the normalized center is defined as the height and the center of the normalized bounding box, respectively. Figures 5 (c) and (d) illustrate them. Let h_1 and h_2 denote the normalized sizes of the parent and child, respectively. Similarly, let c_1 and c_2 denote the normalized centers of those characters.

From the normalized size and the normalized center, the relative size H and the relative position D can be extracted for each pair of adjacent characters as follows:

$$H = \frac{h_2}{h_1}, \quad (1)$$

$$D = \frac{c_1 - c_2}{h_1}. \quad (2)$$

6. Irregular characters

On setting the normalized bounding box, a special treatment must be done for *irregular characters* which may have different sizes and occupy different X,Y, Z regions in different documents. Figure 6 shows several irregular characters. For example, "ψ" occupies Y and Z regions in a document and occupies all $X : Y : Z$ regions in another document. Through a rough observation of the database documents, the following 18 categories are considered as irregular characters: 7 Roman characters (e.g., "i", "t"), 3 Greek characters (e.g., "ϕ", "ψ"), and 8 numeric characters (e.g., "0", "5").

Obviously, these irregular characters badly effect the estimation of the $X : Y : Z$ ratio. In order to suppress the effect, a special treatment is applied for the irregular characters. This special treatment is composed as follows: (i) the elimination of the irregular characters when evaluating the X, Y and Z ratio, (ii) the estimation of the actual $X : Y : Z$ occupation of each irregular character by choosing the most probable occupation from possible occupations, and (iii) the fixation of the normalized bounding box according to this estimation.

7. Experimental results

7.1. Qualitative evaluation with distribution maps

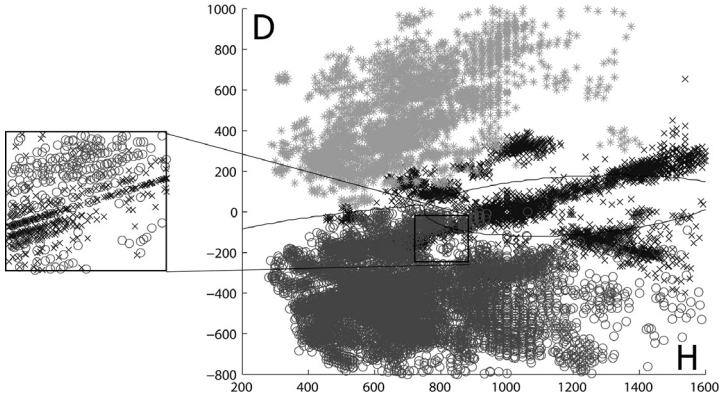
The discrimination performance can be evaluated qualitatively by observing the *distribution map* which shows the distribution of (H, D)-features in a two-dimensional space for pairs of adjacent characters. If there is small overlap among horizontal, subscript and superscript classes, we can expect better discrimination by some appropriate discrimination functions.

Figure 7 illustrates several distribution maps whose details will be discussed latter. In these maps each"$\times$"-shaped dot corresponds to a character pair of the horizontal class (e.g., "xy" and "2α"). Each "$\circ$"-shaped dot corresponds to a pair of the subscript class (e.g., "$\boldsymbol{M}_2$" and "ϵ_X"). Finally, each "$*$"-shaped dot corresponds to a pair of the superscript class (e.g., "$\mathbb{H}^3$").

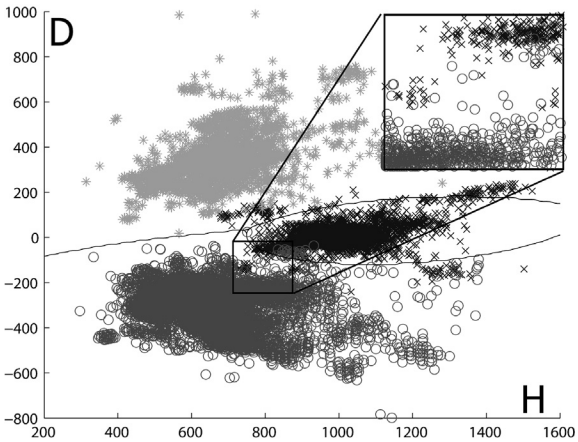
Figure 7 (a) shows the distribution map with the actual bounding boxes. Heavy overlaps among the three classes can be observed on this map. Especially, horizontal ("$\times$") and subscript ("$\circ$") classes are confused and cannot be distinguished. These overlaps come from the variation of the sizes and positions of the actual bounding boxes, that is, the difference of original character shapes (such as the difference between "a" and "A").

Figure 7 (b) shows the distribution map with the normalized bounding box (with the private $X : Y : Z$ ratio determined at each document). The overlaps were decreased drastically and thus we can conclude that the normalization is very powerful for the discrimination task with (H, D)-features. There, however, are still small overlaps. A close investigation of the points in the overlaps revealed that most of them are pairs including irregular characters. These irregular characters might have wrong normalization.

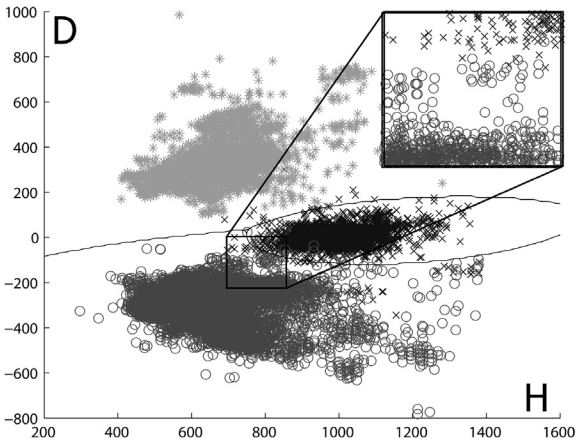
Figure 7 (c) shows the distribution map after applying the special treatment of irregular characters. The overlaps became far smaller than those of (b). Consequently, the special treatment is not trivial but necessary for accurate discrimination. (As shown later, the special treatment reduces 50% of mis-discrimination.) Note that (c) is the best case among those five maps. of Figure 7.



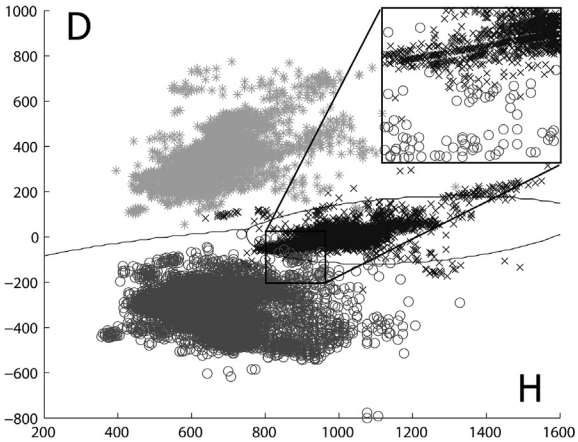
(a) Without any normalization.



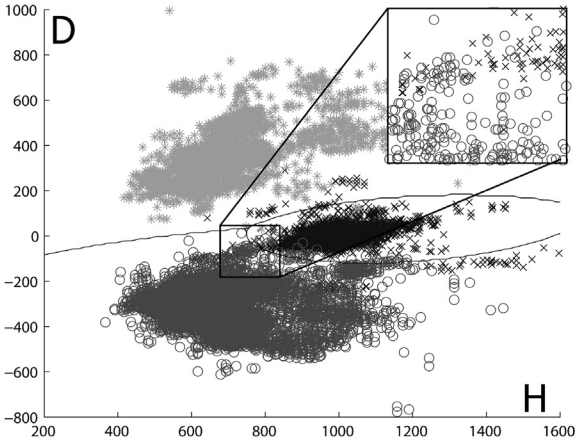
(b) Normalization with private $X : Y : Z$.



(c) Special treatment of irreg. char. + (b).



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



(e) Special treatment of irreg. char. + (d).

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

Table 1. Discrimination accuracy (%) by quadratic classifier on H-D space.

normalize	-	private $X:Y:Z$		common $X:Y:Z$	
special treat.	-	-	+	-	+
accuracy	80.38	99.31	99.76	99.54	99.25

As discussed in 5.2, we can use the common $X:Y:Z$ ratio instead of the private $X:Y:Z$. Figures 7 (d) and (e) show the distribution maps when the common $X:Y:Z$ was used at the normalization. The difference between them is the usage of the special treatment of the irregular characters. The map of (d) has a little bit smaller overlaps than map of (b). That is, the common $X:Y:Z$ outperformed the private $X:Y:Z$ without the special treatment. This is because the private $X:Y:Z$ ratio is more sensitive to irregular characters than the common $X:Y:Z$ ratio.

The special treatment was then applied in Figure 7 (e). Different from the improvement from (b) to (c), we recognize the enlargement of the overlap by the application. This is because the normalized bounding box of irregular characters was wrongly estimated by the common $X:Y:Z$ and then those irregular characters failed at the discrimination.

Consequently, the smallest overlaps occur at Figure 7 (c) where the normalized bounding box with the private $X:Y:Z$ ratio and the special treatment to irregular characters were used. The pairs at the overlaps of the best case mainly includes characters which have styles peculiar to a document. Figure 8 shows two mathematical expressions from two different documents. For example, "θ" and "i" generally occupy the $X:Y$ region and the upper expressions includes such "θ" at its baseline ("θ" in "$d\theta$"). However, all "θ"s of superscripts have different situations; "θ" in the upper expression does not occupy full of the $X:Y$ region and "θ" in the lower expression exceeds the $X:Y$ region. This example shows the necessity of further consideration about document-specific print styles as our future work.

7.2. Quantitative evaluation through quadratic discrimination

Table 1 shows the discrimination accuracy rate by using a simple Bayesian classifier. We assumed that each of the three classes has a two-dimensional Gaussian distribution on the $H-D$ space. This assumption reduced 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. Figure 7 shows the distribution boundary of the quadratic classifiers.

The evaluation results illustrated in this table coincide with the qualitative evaluation in 7.1. The highest accuracy was achieved with the normalized distribution map using private $X:Y:Z$ and special treatment of irregular character. Although we used a simple quadratic classifier to determine the discrimination boundary, the highest accuracy was 99.76%. That is, the discrimination was done almost perfectly by using the two feature H and D.

$$\lim_{r\to\infty}\frac{n_w(r)}{r^\varrho}=\frac{\varrho}{r^\varrho}\int_0^{2\pi}h(e^{i\theta}w)d\theta$$

$$\zeta=\log\frac{1+ze^{-i\theta}}{1-ze^{-i\theta}}$$

Figure 8. The character θ which could not be discriminated correctly from two different documents.

8. Conclusion

This paper discussed a discrimination task of classifying each pair of adjacent characters in a mathematical expression into one of the three classes (horizontal class, subscript class, and superscript class) for realization an accurate structure analysis module of math OCR. For this task, we used two features, relative size and relative position, for describing the relation between the adjacent characters. A large-scale experiment was conducted to evaluate the usefulness of those features, which convey geometrical information of component characters in mathematical expressions.

Experimental results were observed as distribution maps and showed that the two features are sufficient for the discrimination task and could provide 99.7 % accuracy. Through the experiment, we emphasized the importance of special considerations about irregular characters and the use of private $X:Y:Z$ ratio, which is used for making the normalized bounding box. These two points were overlooked in the past attempts, while they give us an important aspect that document-specific characteristics is necessary on the structure analysis.

Future work will focus on the following points.

- Apply our discrimination task on non-alphanumeric symbols, such as operators and parentheses. Although alphanumeric characters are the major part in most

mathematical expressions, non-alphanumeric symbols are also important in the mathematical expressions.

The discrimination of these non-alphanumeric symbols might be more difficult than that of alphanumeric characters, because there is no general normalization scheme for non-alphanumeric specially for parentheses. However, if we can extract any document-dependent characteristics from each individual document, they may be usefully in the discrimination of non-alphanumeric symbols.

- Another utilization of the distribution map. We already know the distribution of baseline, subscript, and superscript characters. If there is any unusual position of these characters on the distribution map, we can detect errors in these characters (e.g., category error, pairing error, broken or touching characters, etc.) Therefore, the proposed method may 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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

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

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

[12] 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.

[13] 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.

[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.