International Journal on Document Analysis and Recognition manuscript No.
(will be inserted by the editor)

Quantitative Analysis of Mathematical Documents

S. Uchida[1], **A. Nomura**[2], **and M. Suzuki**[2]

[1] Department of Intelligent Systems, Kyushu University, 6-10-1, Hakozaki, Higashi-ku, Fukuoka-shi, Japan
[2] Department of Mathematics, Kyushu University, 6-10-1, Hakozaki, Higashi-ku, Fukuoka-shi, Japan

Received June 29, 200x / Revised August 13, 200x

Abstract. Mathematical documents are analyzed from several viewpoints for the development of practical OCR for mathematical and other scientific documents. Specifically, four viewpoints are quantified using a large-scale database of mathematical documents, containing 690,000 manually ground-truthed characters: (i) the number of character categories, (ii) abnormal characters (e.g., touching characters), (iii) character size variation, and (iv) the complexity of the mathematical expressions. The result of those analyses clarifies the difficulties of recognizing mathematical documents and then suggests several promising directions to overcome them.

Key words: Mathematical Document – Database – Mathematical Expressions – OCR – Touching characters

1 Introduction

Optical character reader (OCR) for mathematical and other scientific documents, hereafter called *math-OCR*, is a system for converting scanned page images of such documents into a scientific document format, such as XML, LaTeX, Mathematica, or braille [1]. The development of math-OCR is indispensable for both reducing storage size of math documents and for extending their usability. For example, various search services (e.g., keyword search, definition search, and theorem search) across documents can be made possible by math-OCR. Other emerging applications, such as digital libraries [2,3], will also require assistance from math-OCR.

Math documents have many characteristics that differ from those of non-math documents. Therefore math-OCR need to be furnished with special functions. First, since math documents include many math symbols, font variations, and character size variations, a math-OCR needs a special recognizer that can cope with these characteristics. Second, a math-OCR needs to be furnished with a structure analyzer (i.e., parser) of math expressions [4] to facilitate output in one of the scientific document formats. For example, the element "2" of "$\frac{1}{a^2+b^3}$" should be parsed as the right super-script of "a".

Correspondence to: S. Uchida

This paper analyzes 466 pages of actual math documents to quantify various difficulties in recognizing math documents and to suggest possible remedies. For our quantification task, four viewpoints are chosen:

- the number of character categories,
- abnormal characters (e.g., touching and broken characters),
- character size variation
- the complexity of the math expressions.

These viewpoints were chosen because they are suitable for emphasizing the difference between math and non-math documents, and are therefore closely related to the special math-OCR functions.

Although there has been much previous work done on the development of math-OCRs [5], the characteristics of math documents have not been quantitatively revealed. The results of the analysis undertaken in this paper provide many suggestions useful for developments of practical math-OCRs, and certify the necessity for some techniques that have been employed in math-OCRs.

1.1 Terminology

Hereafter, the term *character* means not only ordinary characters (e.g., "A"), but also math symbols (e.g., "+"), unless otherwise noted. The term *category* means the finest level of character classification and the term *type* means a set of categories having a similar property. For example, "A", "B" and "C" are three categories belonging to the same type (Roman). In contrast, "A"(Roman), "A"(italic), "$\mathcal{A}$"(calligraph), "$\mathbb{A}$"(blackboard bold), "$\mathfrak{A}$" (German), and "$\mathcal{A}$" (script) are six categories belonging to different types. Each character belongs to either the *text region* or the *math region*. The math region includes not only numbered equations but also in-line math expressions. Note that many in-line math expressions are composed of a single character, such as "x" in the sentence "The variable x denotes . . .".

Table 1. Contents of database.

type	category examples	#pre-defined categories	text region #cat.	text region #char. (%)	math region #cat.	math region #char. (%)	total #cat.	total #char. (%)
accent	ˆ ˜ ¯ ¨ ` ˇ _	13	1	2 (<0.01)	7	2,699 (1.72)	7	2,701 (0.39)
arrow	← ↔ ↩ ↖	16	2	6 (<0.01)	6	1,111 (0.71)	6	1,117 (0.16)
big symbol	$\sum \int \prod$	18	0	0 (0.00)	11	2,453 (1.56)	11	2,453 (0.36)
blackboard bold	$\mathbb{A}\mathbb{B}\mathbb{C}\mathbb{D}\mathbb{E}\mathbb{F}$	52	0	0 (0.00)	9	427 (0.27)	9	427 (0.06)
calligraphic	$\mathcal{A}\,\mathcal{B}\,\mathcal{C}\,\mathcal{D}\,\mathcal{E}\,\mathcal{F}$	26	0	0 (0.00)	19	592 (0.38)	19	592 (0.09)
German	$\mathfrak{A}\,\mathfrak{B}\,\mathfrak{C}\,\mathfrak{a}\,\mathfrak{b}\,\mathfrak{c}$	52	0	0 (0.00)	25	1,044 (0.66)	25	1,044 (0.15)
Greek	$\Gamma\Delta\Theta\alpha\beta\gamma$	40	5	25 (<0.01)	33	12,802 (8.14)	33	12,827 (1.86)
italic	*A B C a b c fi*	61	55	63,750 (11.96)	52	50,667 (32.21)	56	114,417 (16.57)
extended Latin	Ä Æ è *Ä Æè*	364	38	453 (0.08)	2	13 (0.01)	38	466 (0.07)
numeric	0 1 2 *0 1 2*	20	20	13,060 (2.45)	14	15,470 (9.83)	20	28,530 (4.13)
operator	$+ - \times / < \&$	92	6	149 (0.03)	49	20,391 (12.96)	50	20,540 (2.98)
others	$\# \% \infty \forall \exists \dagger$	39	12	3,571 (0.67)	17	2,598 (1.65)	22	6,169 (0.89)
parenthesis	( ) { } [ ]	20	7	8,200 (1.54)	12	30,351 (19.29)	12	38,551 (5.58)
point	, . ‘ ’ ′	15	9	21,435 (4.02)	9	7,732 (4.91)	12	29,167 (4.23)
Roman	A B C a b c fi	61	56	422,339 (79.24)	54	8,799 (5.59)	56	431,138 (62.46)
script	$\mathcal{A}\,\mathcal{B}\,\mathcal{C}\,\mathcal{D}\,\mathcal{E}\,\mathcal{F}$	52	0	0 (0.00)	7	175 (0.11)	7	175 (0.03)
total		941	211	532,990 (100.00)	326	157,324 (100.00)	383	690,314 (100.00)

Notes: (i) Each “Roman” and “italic” type includes nine double letters (i.e., ligatures), such as “fi”.
(ii) Each “blackboard bold”, “German”, and “script” type is composed of 26 capital and 26 small letters.
(iii) The pre-defined categories of “big symbol”, “extend Latin”, “operator”, and “others” are listed in **Appendix B**.

2 Outline of database

2.1 Data collection

The documents contained in the database are 30 English articles on pure mathematics (published 1970∼ 2000). A list of the articles is in **Appendix A**. The numbers of pages, characters, and math expressions in the database are 466, 690,314, and 20,859, respectively. This database is larger than other databases used in the past research on math-OCR (e.g., about 15,000 characters of [6], about 10,000 characters of [7], and 350 math expressions of [8]). Note that matrices, tables, and figures are excluded from the database.

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 paper, etc. Several page images are noisy and include a lot of abnormal characters, such as touching characters and broken characters.

2.2 Ground truth

The ground truth for each character was attached *manually* by seven students belonging to a university math department. The ground truth of each character is composed of the following attributes:

- type and category
- text or math region
- normal or abnormal character
- size (height and width)
- location in page
- link
- sub/super-script level.

The numbers of types and categories pre-defined on attaching ground the truth were 16 and 941, respectively. All 16 types are listed in **Table 1** and the all the pre-defined categories of “big symbol”, “extend Latin”, “operator”, and “others” are listed in **Appendix B**. Several types are also familiar in non-math documents (e.g., Roman and numeric), while others are particular to math documents (e.g., operator and calligraphic).

Similar-shaped categories are sometimes defined in different types. For example, Σ (capital sigma / Greek) is similar to $\sum$ (sum / big symbol), and $\cup$ (cup / operators) is similar to $\bigcup$ (bigcup / big symbol). These similar-shaped categories were distinguished manually according to their context and/or usage. Similary, $\hat{\mathrm{E}}$ (extended Latin) and $\hat{\mathrm{E}}$ (E /Roman + ˆ / accent) were distinguished.

Bold and non-bold were not distinguished on attaching their ground truths. Thus, for example, both “A” and “**A**” were classified into the same category “A” and both “A” and “$\boldsymbol{A}$” were classified into “A”. This is because the difference between bold and non-bold is often very subtle (even for humans) and document-dependent. Since their discrimination is important for understanding math expressions, a reasonable treatment for boldface has been left for our future work.

The sixth attribute, links, represents the positional relation to the preceding character and was attached for describing the structure of a math expression (as a tree). There are six kinds of links: horizontal, right-superscript, right-subscript, left-superscript, left-subscript, upper, and under. A math expression that includes one or more links other than a Horizontal one has a two-dimensional (2D) structure. **Figure 1** shows a math expression whose structure is represented by 10 links including four non-Horizontal links. Thus, this math expression has a 2D structure.

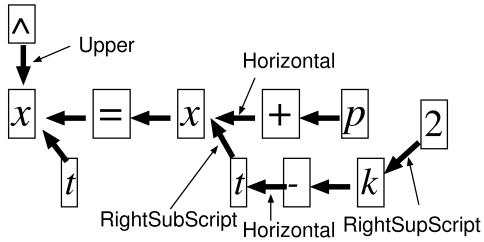

Fig. 1. Links representing the structure of the math expression "$\hat{x}_t = x_{t-k^2} + p$".

$$\lim_{r\to\infty} \frac{\sigma_{D''_{w_i}}(r)}{r^{\varrho+2n-2}} = 0 .$$

Fig. 2. Math expression including high-level sub/super-scripts.

The seventh attribute, the sub/super-script level, describes the depth of sub/super-scripts. For example, in the math expression in **Fig. 1**, "t", "-", and "k" are first-level subscripts, and "2" is a second-level subscript. Note that baselines are assumed on both the numerator and the denominator of a fraction. Thus, in the math expression of **Fig. 2**, σ and r are baseline characters, "D" is a first-level subscript, and "i" is a third-level subscript.

3 Categories and their frequencies

3.1 Number of categories

The contents of the database are summarized in **Table 1**, where it is shown that text, math, and whole regions in the database are composed of 211, 326, and 383 categories, respectively. From this, it is suggested that non-math documents are composed of about 200 ($\sim$ 211) categories. Therefore math documents are composed of about twice as many categories as non-math documents. Thus, not only accurate but also efficient character recognition procedures are required in math-OCR.

3.2 Frequencies of type and category

The importance of italic characters is quantified by **Table 1**. Namely, the frequency of italic characters are very high not only in the text region (11.96%) but also in the math region (32.21%). Since italic characters are slanted and therefore often misrecognized in ordinary OCRs, this result emphasizes the necessity of some special pre-processing, such as [9], to detect and deslant italic characters. Note that many italic characters can be found in the text region as well as the math region because theorems are often printed with italic characters.

Table 1 also shows that 466 characters of extended Latin type were found in the database. This number may be larger than that of ordinary non-math documents, because the characters were found in author names and paper titles in bibliographies and abstracts written in French or other non-English languages. Note that among the above the 466 extended Latin characters, 13 were found in the math region. Some of them were used in function and variable names that originated from a person's name. **Figure 3** shows that an extended Latin character "$\check{S}$" is included in a math expression "$\check{S}(A)$", that originates from a Russian mathematician called "Šilov".

in $C(\bar{\Omega})$ with respect to $\|\ \|_{\bar{\Omega}}$.
we denote by $\check{S}(A)$ the Šilov
eans the set of points on $b\Omega$,

Fig. 3. Math expression including an extended Latin character "$\check{S}$".

$$\begin{array}{c} ( )\ 1\ ,\ =\ -\ 2\ |\ 0\ n\ +\ i\ k\ p\ z\ f\ r\ a\ x^{-}\ t\ s\ j\ G\ S\ \alpha\ w\ \lambda\ \in \\ *\ C\ -\ q'\ \sigma\ /\ M\ \rightarrow\ b\ P\ g\ A\ d\ u\ \partial\ R\ h\ X\ H\ D \end{array}$$

Fig. 4. The 50 most frequent categories (first: "(" $\rightarrow$ 50th:"D") in math region. The character "$-$" next to "$=$" is minus, "$^-$" next to "x" is overline, and "$-$" next to "C" is fraction bar.

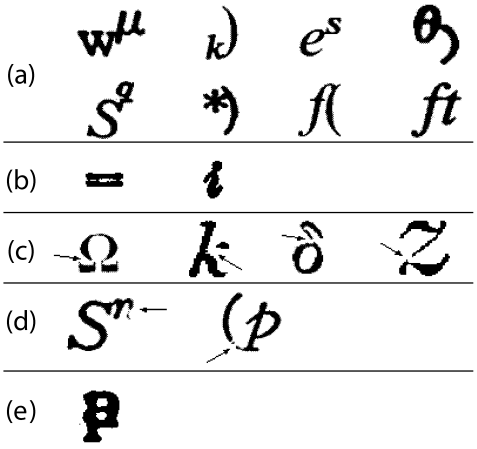

Fig. 5. (a) Touching characters, (b) self-touching characters, (c) broken characters, (d) touching and broken characters, and (e) overlaid characters ("a"+ "p"). Broken points are indicated by arrows.

Figure 4 lists the 50 most frequent categories in the math region. This list shows that not only italic characters but also parentheses, accents, and operators were frequent in the math region.

In math documents, there exist similar-shaped categories (e.g. "r", "r", "γ", "Υ", and "ɾ"). One strategy to improve total recognition accuracy is to use their occurrence rates as empirical prior probabilities in the recognizer. This strategy improves the recognition accuracy of popular characters ("r" and "r"). Rare characters, however, might still be misrecognized, since their occurrence rates are near or equal to zero as indicated by **Table 1**. Another and more promising strategy is the incorporation of document-dependent post-processing. For example, the mutual comparison of characters initially recognized as "r" will be useful to pick out the misrecognition "Υ"$\rightarrow$"r", since the shape of "r" is stable within a document. A self-corrective classifier [10,11] is also a promising document-dependent approach where a recognition procedure is repetitively applied to a document while updating its dictionary according to the results of its recognition.

Table 2. The number of normal and abnormal characters. (Upper: The number of characters. Lower: Percentage.) Note that most of the "non-baseline" characters in the text region are footnote numbers.

	text region			math region			total
	baseline	non-baseline	subtotal	baseline	non-baseline	subtotal	
normal	524,715 (98.47)	102 (99.03)	524,817 (98.47)	118,625 (98.18)	35,230 (96.52)	153,855 (97.79)	678,672 (98.31)
touching	6,630 (1.24)	0 (0.00)	6,630 (1.24)	1,267 (1.05)	292 (0.80)	1,559 (0.99)	8,189 (1.19)
self-touching	82 (0.02)	0 (0.00)	82 (0.02)	103 (0.09)	249 (0.68)	352 (0.22)	434 (0.06)
broken	1,445 (0.27)	1 (0.97)	1,446 (0.27)	819 (0.68)	729 (2.00)	1,548 (0.98)	2,994 (0.43)
touch&broken	13 (<0.01)	0 (0.00)	13 (<0.01)	9 (0.01)	1 (<0.01)	10 (0.01)	23 (<0.01)
overlaid	2 (<0.01)	0 (0.00)	2 (<0.01)	0 (0.00)	0 (0.00)	0 (0.00)	2 (<0.01)
total	532,887 (100.00)	103 (100.00)	532,990 (100.00)	120,823 (100.00)	36,501 (100.00)	157,324 (100.00)	690,314 (100.00)

Table 3. Distribution of abnormal character rates of all 30 documents.

abn. rate(%)	<0.2	0.2∼0.5	0.5∼1	1∼2	2∼3	3∼4	4∼5	>5
#documents	3	4	5	7	2	6	1	2

Table 4. Top 15 categories with high frequencies of (a) touching characters and (b) broken characters in the math region. The middle row (#char) is the number of abnormal characters in the database, and the bottom row (#doc) is the number of documents that include an abnormal character. The 11th and the 14th touching characters are overline and fraction bar, respectively.

(a) Touching character

rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
cat.	(	p	)	∂	r	2	v	ρ	r	V	¯	M	i	$-$	S
#char	261	163	148	62	61	46	43	41	33	32	30	29	28	26	26
#doc	8	9	10	1	3	6	2	3	2	3	5	1	3	4	3

(b) Broken character

rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
cat.	s	p	K	$\mathfrak{N}$	Y	$\prod$	j	k	n	(	2	$\cong$	a	x	1
#char	147	129	79	68	61	56	54	49	48	37	34	28	28	27	24
#doc	3	4	2	1	2	1	2	6	7	14	12	2	6	7	8

4 Abnormal characters

4.1 Distribution of abnormal characters

In this section, abnormal characters are analyzed from several viewpoints. As shown in **Fig. 5**, there are five kinds of abnormal characters: touching, self-touching, broken, touching and broken , and overlaid characters. Overlaid are distinguished from touching characters, because they are caused by typographical errors.

As shown in **Table 2**, the database includes 11,672 (1.69% of all characters) abnormal characters: 8,189 (1.19%) touching , 434 (0.06%) self-touching, 2,994 broken (0.43%), 23 touching and broken (<0.01%), and 2 overlaid characters (< 0.01%). Abnormal characters are found more frequently in the math region (2.21%) than in the text region (1.53%). In the math region, self-touching characters and broken characters are found far more frequently among the non-baseline characters than among the baseline characters.

Table 3 shows the distribution of abnormal character rates of all the 30 documents. This table shows that the rate varies drastically with those documents. In fact, the maximum and the minimum rates were 11.0% and 0.11%, respectively. This variation is due printing conditions (e.g., quality of paper sheet, thickness of ink, fonts, space between characters) that vary with documents. On the other hand, since the printing condition is constant within a document, abnormal characters are often document-specific; namely, abnormal characters of a certain category (e.g., touching characters of "∂") are often found in only one document. Those facts suggest that a document-dependent processing will be effective for detection and normalization procedures in respect to abnormal characters.

Table 4 (a) and (b) shows the top 15 categories yielding many touching and broken characters, respectively, in the

math region. These tables also show the number of documents which include each abnormal character. Several abnormal characters, such as the touching character of parenthesis, are common, and can be found in many documents. Other abnormal characters are often document-specific. The touching characters of "∂" and "M" and the broken characters of "$\mathfrak{R}$", and "$\prod$" are typical document-specific abnormal characters. As noted above, document-specific abnormal characters will be detected and normalized by some document-dependent procedure (like the post-processing discussed in 3.2).

The normalization of the abnormal characters in the math region is very challenging, but it is important for the following two reasons. First, abnormal characters in the math region are hard to recognized without normalization. This is because there is no lexicon for math expressions and thus abnormal characters cannot be recognized correctly by cooperation of linguistic *a priori* knowledge. Second, the performance of structural analysis of math expressions will be degraded by abnormal characters. The structural analysis of a math expression might fail completely due to only one abnormal character because the abnormal character will badly affect the estimation of font sizes and positions. Consequently, detection and normalization of abnormal characters in the math region is indispensable for practical math-OCRs.

4.2 Touching characters in math region

In ordinary OCRs, touching characters are not fatal. This is because word lexicons, horizontal segmentation, and their combination (sometimes called segmentation-by-recognition strategy) will help separation and recognition. However, in math-OCR, touching characters are often fatal in the math region. As noted 4.1, there is no lexicon in math expressions. Further, characters are often touching in non-horizontal directions as shown in **Fig. 5**(a).

The number of non-horizontally touching character pairs were 176 among 760 touching character pairs in the math region. (Namely, there were 176 touching character pairs each composed of a baseline character and a first-level sub/super-script). Thus, at least 23% of the touching characters in the math region can not be separated by conventional horizontal segmentation techniques. This result emphasizes the necessity of segmentation techniques specialized for math expressions, such as [12, 13].

Sometimes, three or more characters touch together. The maximum numbers of characters touching together were 7 in the text region and 5 in the math region.

5 Character size variation

5.1 Frequency of size variation

The analysis of size variation is important to clarify the difficulties in setting prototypes for a math-OCR. This is because character features will often be affected by character size, and therefore multiple prototypes should be set when a large size variation is observed.

Figure 6 shows the frequency for each character of a size (height) variation from the average of its category. Abnormal characters were excluded from this evaluation. This figure quantifies that character size variation in the math region is heavier than that in the text region. Specifically, about 63.4% of characters in the math region are scaled up or down by 5% or more. This variation is mainly due to the existence of sub/super-scripts and of scalable math symbols, such as parentheses. In fact, the character image with the largest size variation (737%) is of parenthesis "|".

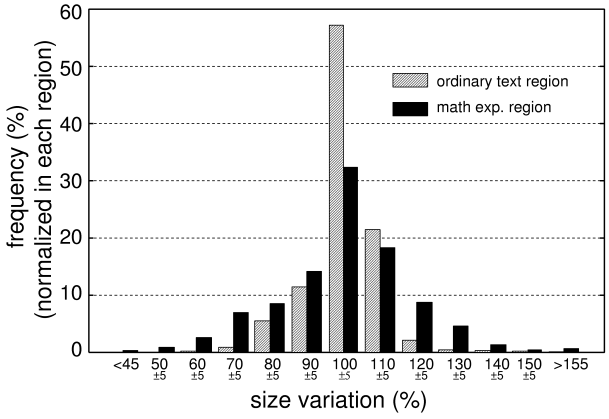

Fig. 6. Frequency of size variation from the average of each category.

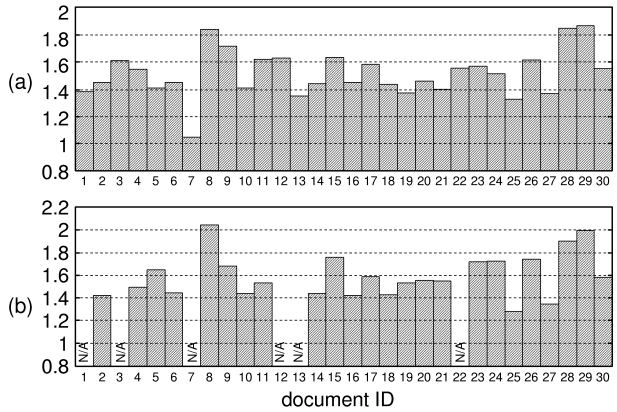

Fig. 7. Ratio between the intra-document average size of first-level sub/super-scripts and that of baseline characters. A ratio equal to 2 means that the baseline characters are twice as large as first-level sub/super-scripts.

5.2 Size of sub/super-scripts

The analysis of size variation is also important to clarify difficulties distinguishing between baseline characters and sub/super-scripts using size information. The most naive approach for the distinction is a thresholding operation on character size using a fixed threshold value. This approach, however, is not sufficient, because the sub/super-scripts of a category in a document are often larger than the baseline characters of the category in another document. In fact, among all 201 categories used both as both baseline characters and first-level sub/super-scripts, 94 (46%) will have trouble being distinquished by this approach.

Table 5. Detail of links. (Upper: The number of links. Lower: Percentage.)

horizontal	right-superscript	right-subscript	left-superscript	left-subscript	upper	under	total
107,495	7,803	15,013	10	2	1,645	4,578	136,546
(78.7)	(5.71)	(11.00)	(0.01)	(<0.01)	(1.20)	(3.35)	(100.00)

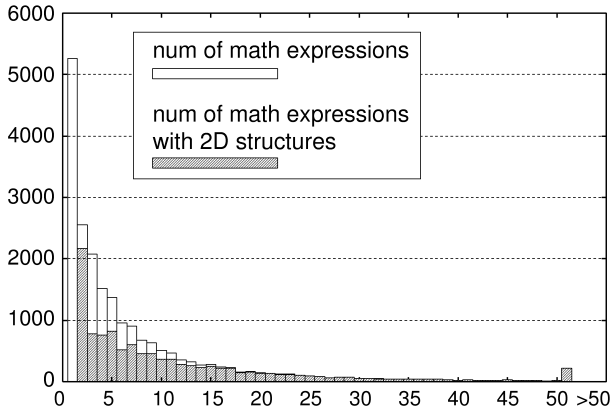

Fig. 8. The distribution of the number of characters per math expression.

An alternative and more promising approach of the distinction of sub/super-scripts is adaptive thresholding, where a threshold is determined for each document according to the average character size of the document. Unfortunately, this approach is also insufficient. **Figure 7** (a) and (b) show the ratio between the average size (height) of first-level sub/super-scripts and that of baseline characters for each document. To obtain the ratios in **Fig. 7** (a), italic small letters with neither ascender nor descender (i.e., a, c, e, m, n, o, r, s, u, v, w, x, z) were used. To obtain the ratios in **Fig. 7** (b), italic capital letters except for "Q" (i.e., A, B, C, ..., P, R, ..., Z) were used. For documents with a ratio that exceeds $1.4 \sim 1.5$, the sub/super-scripts can be distinguished rather easily by adaptive thresholding. For several documents, however, the ratio is close to 1, namely, sub/super-scripts and baseline characters have almost the same size. Especially, the ratio of document #7 is 1.04 in **Fig. 7** (b). The existence of such documents confirms that character size is not sufficient to distinguish sub/super-scripts, and other features such as relative position should be cooperatively utilized.

6 Complexity of math expressions

The database contains 20,859 math expressions (including inline math expressions composed of a single character, such as "x"); **Fig. 8** shows the distribution of the number of characters per math expression. This result is that this distribution can be approximated as an exponential distribution. **Figure 8** also shows that a math expression sometimes becomes very large (i.e., containing over 50 characters). Thus, the structural analysis procedure should be sufficiently computationally feasible for managing such large math expressions.

Figure 8 also depicts the distribution of the number of characters per math expression with 2D structures, i.e., math expressions with one or more non-horizontal links. This distribution shows that most math expressions have 2D structures. In fact, about 70% of math expressions composed of two or more characters have 2D structures. The results of these observations quantify the importance of 2D structural analysis techniques [6, 14–16] in math-OCR.

Table 5 details the links. Most (78.7%) of links were horizontal while about 17% were right sub/super-scripts. Left sub/super-scripts were very rare. Note that 66.1% of upper links and 24.7% of under links were connected to fraction bars.

There were 36,501 sub/ super-scripts in the database. Among them, 33,949 (93.0%) sub/ super-scripts were first-level (e.g., "D"in **Fig. 2**), 2,504 (6.86%) second-level ("w"), and 48 (0.13%) third-level ("i"). No fourth or higher-level sub/super-script was found.

7 Conclusion

About 450 pages of math documents containing 670,000 ground-truthed characters were analyzed to quantify the difficulties involved in the development of a practical math-OCR. The main results of the analysis can be summarized as follows.

- The math, text, and whole regions are composed of 326, 211, and 383 categories, respectively; demonstrating that math documents contain about twice as many categories as non-math ones.
- Italic characters were very frequent in not only the text region (12%) but also the math region (32%).
- The frequency of each category varies drastically; indeed the frequencies of many categories are almost zero. Thus, a naive recognizer where those frequencies are used as empirical prior probabilities will not provide reasonable recognition results.
- The math region includes more abnormal characters (2.21%) than the text region (1.53%).
- Abnormal characters are often document-specific. Namely, all abnormal characters of a category are found in only one document.
- Larger size variations were observed in the math region than in the text region.
- About 23 % of touching characters in the math region are touching non-horizontally. Thus, the separation technique used for touching characters in the text region is not sufficient.
- A document whose sub/super-scripts and baseline characters are nearly equal-sized was noted. Thus, character size is not sufficient to distinguish sub/super-scripts.
- Sometimes complex math expressions composed of over 50 characters exist.
- About 70 % of the math expressions composed of two or more characters were two-dimensional. As well, almost

all the math expressions composed of over 15 characters were two-dimensional.
- Several third-level sub/super-scripts were found, whereas forth (or higher)-level sub/super-scripts were not.

As a future work, the results here will be utilized in the development of a practical OCR, such as INFTY[17].

Acknowledgements. The authors greatly thank the members of the Suzuki Laboratory of Kyushu University for their generous efforts in the development of the database. This work is partially supported by The Ministry of Education, Culture, Sports, Science and Technology of Japan under Kyushu University 21st Century COE Program (Development of Dynamic Mathematics with High Functionality) and a Grant-in-Aid for Scientific Research No.14380182.

References

1. S. Hara, et al. (2000) MathBraille; a system to transform LATEX documents into Braille. SIGCAPH Newsletter, 66:17-20
2. G. O. Michler (2001) Report on the retrodigitization project "Archiv der Mathematik". Archiv der Mathematik, 77:116-128
3. K. Dennis, G. O. Michler, G. Schneider, and M. Suzuki (2003) Automatic reference linking in distributed digital libraries. In: Proceedings Workshop of Document Image Analysis and Retrieval (DIAR-03)
4. D. Blostein and A. Grbavec (1997) Recognition of mathematical notation. In: Handbook of Character Recognition and Document Image Analysis, 557–582, Eds. H. Bunke and P. S. P. Wang, World Scientific
5. K. -F. Chan and D. -Y. Yeung (2000) Mathematical expression recognition: a survey. Int. J. Document Analysis and Recognition, 3(1):3-15
6. H.-J. Lee and J.-S. Wang (1997) Design of a mathematical expression understanding system. Pattern Recognition Letters, 18(3):289–298
7. M. Okamoto, H. Imai, and K. Takagi (2001) Performance evaluation of a robust method for mathematical expression recognition. In: Proceedings of International Conference on Document Analysis and Recognition, 121–128
8. J. Mitra, U. Garain, B. B. Chaudhuri, K. Swamy, and T. Pal (2003) Automatic understanding of structures in printed mathematical expressions. In: Proceedings of International Conference on Document Analysis and Recognition, 540–544
9. B. B. Chaudhuri and U. Garain (2001) Extraction type-based meta-information from imaged documents. International Journal on Document Analysis and Recognition, 3(3):138–149
10. G. Nagy and G. Shelton, Jr. (1966) Self-corrective character recognition system. IEEE Trans. Information Theory, 12(2):215–222
11. H. S. Baird and G. Nagy (1994) A self-correcting 100-font classifier. In: Document Recognition, Proceedings of SPIE, 2181:106–115
12. M. Okamoto, S. Sakaguchi, and T. Suzuki (1999) Segmentation of touching characters in formulas. Doc. Anal. Sys.: Theory and Practice. Third IAPR Workshop, DAS'98. Selected Papers (Lect. Note in Comput. Sci. vol.1655, Springer-Verlag)
13. A. Nomura, K. Michishita, S. Uchida, and M. Suzuki (2003) Detection and segmentation of touching characters in mathematical expressions. In: Proceedings of International Conference on Document Analysis and Recognition, 1:126–130
14. J. Ha, R. M. Haralick, and I. T. Phillips (1995) Understanding mathematical expressions from document images. In: Proceedings of International Conference on Document Analysis and Recognition, 956–959
15. Y. Eto and M. Suzuki (2001) Mathematical formula recognition using virtual link network. In: Proceedings of International Conference on Document Analysis and Recognition, 762–767
16. R. Zanibbi, D. Blostein, and J. R. Cordy (2002) Recognizing handwritten mathematical expressions using tree transformation. IEEE Trans. Pattern Analysis and Machine Intelligence, 24(11):1455–1467
17. `http://www.inftyproject.org`

A List of documents in the database

The documents contained in the database comprise 30 English language articles on pure mathematics:
• Acta Math., 124(1-2), 37-63, 1970. • *ibid.*, 181(2), 283-305, 1998. • Ann. Sci. Ecole Norm. Sup., 4d sér, t.3, 273-284, 1970. • *ibid.*, t.30, 367-384, 1997. • Ann. Inst. Fourier, 20(1), 493-498, 1970. • *ibid.*, 49(2), 375-404, 1999. • Ann. Math., 91, 550-569, 1970. • Ann. Math. Studies, 66, 157–173, 1971. • Arkiv für Matematik, 9(1), 141-163 1971. • *ibid.*, 35(1), 185-199, 1997. • Bull. Amer. Math. Soc., 77(1), 157-159 1971. • *ibid.*, 77(1), 160-163 1971. • *ibid.*, 80(6), 1219-1222, 1974. • *ibid.*, 35(2), 123-143, 1998. • Bull. Soc. Math. France, 98, 165-192, 1970. • *ibid.*, 126, 245-271, 1998. • Invent. Math., 9, 121-134, 1970. • *ibid.*, 138, 163-181, 1999. • J. Math. Soc. Japan, 27(2), 281-288, 1975. • *ibid.*, 27(2), 289-293, 1975. • *ibid.*, 27(2), 497-506, 1975. • J. Math. Kyoto Univ., 11(1), 181-194, 1971. • *ibid.*, 11(1), 373-375, 1971. • *ibid.*, 11(2), 377-379, 1971. • Kyushu J. Math., 53, 17-36, 1999. • Math. Ann., 225(3), 275-292, 1977. • *ibid.*, 315, 175-196, 1999. • Tohoku Math. J., 25, 317-331, 1973. • *ibid.*, 25, 333-338, 1973. • *ibid.*, 42, 163-193, 1990.

B Detail of Categories

B.1 Categories of "big symbol" Type

The "big symbol" type consists of the following 18 pre-defined categories: $\sqrt{\ }$, $\sum$, $\prod$, $\coprod$, $\bigcup$, $\bigcap$, $\bigvee$, $\bigwedge$, $\bigoplus$, $\bigotimes$, $\int$, $\oint$, $\iint$, $\iiint$, $\iiiint$, $\int\cdots\int$, ——(fraction bar), and $\sqrt{\ }$ (continued fraction).

B.2 Categories of "extended Latin" Type

The "extended Latin" type consists of 364 pre-defined categories. Among them, of 110 rather common categories are the following 55 and their italic versions: À, Á, Â, Ã, Ä, Å, Ç, È, É, Ê, Ë, Ì, Í, Î, Ï, Ñ, Ò, Ó, Ô, Õ, Ö, Ø, Ù, Ú, Û, Ü, Ý, ß, à, á, â, ã, ä, å, ą, è, é, ê, ë, ì, í, î, ï, ñ, ò, ó, ô, õ, ö, ø, ù, ú, û, ü, and ý. Among the 466 extended Latin characters in the database, 451 come from the 110 categories noted above. The remaining 15 come from the other 254 pre-defined categories, that are composed of 127 very rare categories and their italic versions. The 15 characters are: Š (8 characters), Č (3), Ş (1), ţ (1), ş (1), and č (1). The 5 most frequent categories of extended Latin are: é (128 characters), É (75), *é* (43), ü (25), and ö (23).

B.3 Categories of "operator" Type

The "operator" type consists of 92 pre-defined categories, which are divided into 69 relational and 23 binary operators. The relational operators are: $<$, $=$, $>$, $\in$, $\ni$, $\subset$, $\supset$, $\Subset$, $\Supset$, $\leq$, $\geq$, $\leqq$, $\geqq$, $\ll$, $\gg$, $\equiv$, $\div$, $\neq$, $\not\subset$, $\not\supset$, $\not\in$, $\not\ni$, $\sim$, $\approx$, $\simeq$, $\cong$, $\prec$, $\succ$, $\propto$, $\parallel$, $\perp$, $\vdash$, $\dashv$, $\lhd$, $\rhd$, $\Colon$, $\eqcolon$, $\subseteq$, $\supseteq$, $\subsetneqq$, $\supsetneqq$, $\subsetneq$, $\supsetneq$, $\subsetneqq$, $\supsetneqq$, $\preceq$, $\succeq$, $\models$, $\not\equiv$, $\not\leq$, $\not\geq$, $\not\leqq$, $\not\geqq$, $\neq$, $\lesssim$, $\gtrsim$, $\not\lesssim$, $\not\gtrsim$, $\Cap$, $\lessapprox$, $\gtrapprox$, $\smile$, $\frown$, $\asymp$, $\nmid$, $\not\subseteq$, $\not\supseteq$, $\nparallel$, $\not\prec$, and $\not\succ$. The binary operators are: $\&$, $*$, $+$, $-$, $/$, $\setminus$, $\pm$, $\mp$, $\times$, $\div$, $\oplus$, $\ominus$, $\otimes$, $\cup$, $\cap$, $\vee$, $\wedge$, $\dotplus$, $\ltimes$, $\rtimes$, $\odot$, $\hat{}$, and $\circ$.

B.4 Categories of "others" Type

The "others" type consists of the following 39 pre-defined categories: !, #, \$, %, : (colon), ; (semicolon), ?, @, ¥, _ (under score), ∞, ∂, ∇, ℓ, $\hbar$, $\Re$, $\Im$, $\aleph$, $\emptyset$, $\forall$, $\exists$, $\neg$, $\angle$, $\triangle$, $\square$, $\measuredangle$, $\blacksquare$, §, †, ‡, ¶, $\flat$, $\natural$, ©, $\wp$, $\star$ (star), − (hyphen), — (long hyphen), and $\hbar$. The 5 most frequent categories of this type are : − (hyphen), : (colon), ∂, ∞, and ; (semicolon).

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 was employed by SECOM Co., Ltd., Tokyo, Japan to work on speech processing. Presently, he is an associate professor at the Faculty of Information Science and Electrical Engineering, Kyushu University. His research interests include image pattern analysis and recognition. Dr. Uchida is a member of IEEE, IEICE, IPSJ, ITE, and ASJ.

Akihiro Nomura received B. Sci. and M. Sci. degrees from Kyushu University in 2002 and 2004, respectively. Presently, he is a Ph. D candidate at the Graduate School of Mathematics, Kyushu University, and works on the development of OCRs for scientific documents.

Masakazu Suzuki received B. Sci. and M. Sci. degree 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 at the Centre National de la Recherche Scientifique (CNRS) from 1975 to 1977 and at Kyushu University from 1977, his main research subjects have been complex analysis and algebraic geometry. He is currently a professor in the Faculty of Mathematics, Kyushu University. In recent years, his research interests have included mathematical document recognition and mathematical knowledge management. Dr. Suzuki is a member of MSJ, JSSAC, JSIAM, IPSJ, and IEICE.