Optical recognition of printed mathematical documents

K.Inoue, R.Miyazaki, M.Suzuki[*]

Graduate School of Mathematics, Kyushu University 36,
Fukuoka, 812 Japan

Abstract

This paper describes our new system of OCR, which can handle Japanese scientific documents containing mathematical formulas. Our system consisits of the following two major steps.

After the extraction of the text lines, including mathematical formulas, from a scanned page image, we first segment each line into the Japanese area and the mathematical formula area. This segmentation and the recognition of the Japanese characters are done at the same time by a dynamic programming algorithm. The correction algorithm of the recognition, based on the linguistic morphology, is also implemented, treating the mathematical areas as unknown nouns.

The second part of the process analyzes the mathematical formula area. Here, we have improved considerably the two pioneer works [1] and [2] on mathematical formula recognition. We use the top-down method throughout all the formula recognition process, making use of the recurrent algorithm. By the aid of several other thecnical ideas, this recurrent algorithm allowed the system to recognize correctly even more complicated formulas than we use normally, except at the moment, for the matrices.

The system works reliably on almost noiseless images obtained by 400 dpi scanning from the usual clearly printed documents.

1 Introduction

The recent development of the OCR (optical character recognition) technology made possible its practical use in various applications.

[*] E-mail : suzuki@math.kyushu-u.ac.jp

However, there is no commercial OCR software which can recognize the content of scientific documents including mathematical formulas. In fact, there has been very little research on this subject up to recent years. The lack of this technology presents a serious difficulty in making use of OCR in scientific fields. If one tries to use OCR software to transform some scientific documents into an electronic form of data base, for example, not only are the mathematical formulas not recognized at all, but the recognition results of the text parts to the left and right side of formulae are often inaccurate. This limitation of the applicability of OCR reduces largely the effectiveness of its use in scientific fields, especially in mathematics.

In this paper, we shall describe our new experimental system of optical recognition of Japanese scientific documents including mathematical formulas. The recognition process can be divided into the following three steps:

(1) Segmentation of the Japanese area and the mathematical formula area,

(2) Recognition of Japanese characters,

(3) Recognition of the mathematical formulas.

In our system, steps (1) and (2) are performed at the same time, using the information obtained from the results of character recognition[1] candidates and a dynamic programming algorithm. These steps are described in section 2.

In step (3), mathematical formulas are recognized[2] by the top-down approach with a recurrent algorithm for subformula areas included in formulas such as fractions, integrals, summations, etc. The outline of this step is described in section 3.

In the following, we shall assume that the skew correction of the page image and the segmentation of the page area into columns and lines have already been done.

2 Extraction and recognition of the Japanese characters in a line

In this step, we extract and recognize the Japanese characters in a scanned line image, and separate the line into the Japanese areas and the mathematical areas. The Japanese area contains only Japanese characters (kanji,

[1] As for the Japanese character recognition, we used the OCR engine offerd by RICOH Ltd. We would like to express our hearty thanks here.

[2] As for the character recognition in the formula area, we used an OCR engine developed by our own method, adapted to the recognition of the alphabets and the mathematical symbols.

kana and Japanese punctuation symbols), while the mathematical area covers the complement. Note that alphanumeric characters are all included in the mathematical area. If the character recognition routine always gives correct results, the segmentation is a very simple task. However, it generally assigns a character candidate, even if it may be wrong, and we have to find a method to check it.

2.1 Segment elements

We first cut the scanned line image using vertical lines on which no black pixel exists, and get a sequence of disjoint rectangular areas bounding black images non-separable by these vertical white lines (see fig.1 below). We shall call the rectanglar area units thus obtained "segment elements". In Japanese, there are many characters composed of several segment elements (see fig.1 and fig.2 below). Therefore, the first problem is to determine the groups of the segment elements corresponding to character units.

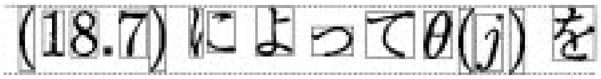


fig.1

Generally, printed Japanese texts have the following properties:

1. The maximum number of segment elements in one character is 7.
2. Aspect ratios of most of the kanji characters are close to 1 : 1.
3. Most of the printed Japanese characters in a line have approximately the same width and height. Only a few kana characters have unusual width or height.
4. The printing pitch of the Japanese characters is nearly constant in a line. The pitch may differ line-by-line.

We use these properties to reduce the number of times of the character recognitions for overlapped regions in the character segmentation algorithm below.

2.2 Virtual character lattice

We first construct a lattice structure as follows.

The vacant spaces between the segment elements will be called "separators". For each separator, we associate a "node".

For each pair of nodes such that the succession of segment elements between the two nodes may correspond to one character in view of the conditions 1 – 3 of the previous section, we join these pair of nodes by an arc and assign the result of the character recognition for the corresponding area and a value which concerns the reliability of the character recognition (fig.2).

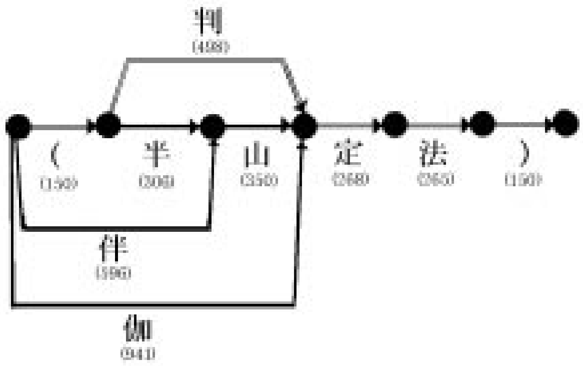


fig.2

The value assigned to each arc is called “cost” and is lower if the recognition result is reliable. These character arcs may overlap each other (fig.2) and may also contain wrongly recognized characters.

2.3 Character segmentation

A succession of arcs with no overlap and no gap is called “path”. The solution of the character segmentation is a “path” from the left end node to the right end node, composed of the arcs corresponding correctly to characters in the line.

We regard this problem of finding the correct path as a shortest-path problem, and apply the dynamic programming algorithm to solve it. The cost attached to each arc expresses the level of suspicion of the character recognition and the path with the smallest sum of the costs is taken to represent the best segmentation.

The most delicate part of this step is how to determine the costs of the arcs. To obtain a correct segmentation, the cost needs to conform to the following conditions:

1. The cost is lower when the recognition is more reliable, whether the character is Japanese or not.

2. For a path P composed of the arcs with reliable recognition results, the cost of the arc joining the two end nodes of P (with probably wrong recognition result) is higher than the sum of the costs of arcs of P (see fig.2).

3. The cost is higher if the aspect ratio of the corresponding area is different from the usual aspect ratio value of the character of the recognition result.

In the implementation, the cost is determined after several operations on the scores returned by the recognition system, taking into account the geometric aspect ratio condition above.

2.4 Segmentation of Japanese/mathematics areas

After the character segmentation, we separate the line into Japanese areas and mathematics areas using the obtained character recognition results.

Since our character recognition system sometimes returns a Kanji character for an arc in the mathematical area, we introduced a pair of supplementary features HCN/VCN to correct the recognition results. The Horizontal Crossing Number(HCN) is the largest number of the black runs in the horizontal lines in the rectangular bounding box of the character, after removin the noise efects. The Vertical Crossing Number(VCN) is defined in the same way. Some types of misjudgements in the distinction between Japanese/mathematics are corrected by using these HCN/VCN features, aspect ratio of the character area and the score ratio of the secondary candidate over the first candidate of the character recognition.

2.5 Correction using Japanese linguistic informations

The correction algorithm of the previous section does not cover the errors of character recognition, especially for characters with complicated shapes. In the character segmentation process of section (2.3), the OCR system assigns several candidates to each arc as the recognition result. We suppose that, even when the first candidate is wrong, the correct result exists in the list of candidates in general. On this assumption, we use a dictionary of Japanese words and morphological information to correct the errors in the recognition.

Japanese sentences are written with no spaces between words. Some parts-of-speech, verbs, adjectives, and auxiliary verbs are conjugated. There is a well-known Japanese morphological analyser, released for researche in parsing Japanese by computer, called "Chasen[3]", which can break the Japanese sentences into a list of words. We made use of its dictionaries and some of its algorithms, but there are two main differences between the method of Chasen and our correction mechanism:

1. In OCR, there are several candidates assigned to one position of character, so that we have to search words substituting some candidates.
2. In our case, mathematical formulas are permitted to be embedded into Japanese sentences. We regard the mathematical area as an unregistered noun in linguistic analysis.

We extended the algorithms of Chasen in these two ways. A line is separated into a list of words by morphological analysis. We select the list which has the least sum of costs by dynamic programming, and correct the candidates and Japanese/mathematics decision according to the list. The cost is computed by taking into account the possible connection between the words.

[3] developed in Matsumoto laboratory at Nara Institute of Science and Technology.

We implemented all these steps and tested it on 50 page images, input by a 400DPI image scanner from clearly printed mathematical documents. The results still contained a few errors in each page.

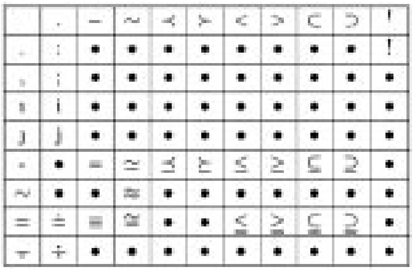
3 Mathematical formula recognition

In this section, we sketch our recognition algorithm of the formulas in the segmented mathematics area as described in the previous section. In the following, all the symbols used in mathematical formulas are called "symbol" including alphabets and numerals.

3.1 Character and Symbol Recognition

First, each connected component is recognised by our own OCR engine. Then, the separate symbols such as "=, i, j", etc. are unified into one symbol using a coupling table (fig.3) in the same way as in [2].

The one-character recognition system which we used in the previous section[4] is mostly adapted to the Japanese character recognition, and is not adapted to the alphabets and the mathematical symbols. Therefore, we developed our own OCR engine, which recognises 344 kinds of symbols including roman and italic alphabets, Greek letters, numerals, and other mathematical symbols including parts of separate symbols. The recognition rate of our system is about 99 % for scanned images of 400DPI of clearly printed documents by TEX, and about 96.5 % for 400DPI images of textbooks printed using system other than TEX[5].



	.	−	∼	≺	≻	<	>	⊂	⊃	!
.	:	●	●	●	●	●	●	●	●	!
,	;	●	●	●	●	●	●	●	●	●
ı	i	●	●	●	●	●	●	●	●	●
ȷ	j	●	●	●	●	●	●	●	●	●
−	●	=	≃	⪯	⪰	≤	≥	⊆	⊇	●
∼	●	●	≈	●	●	●	●	●	●	●
=	≐	≡	≅	●	●	≦	≧	⫅	⫆	●
⨪	÷	●	●	●	●	●	●	●	●	●

fig.3

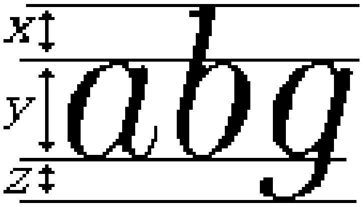


fig.4

3.2 Normalized size and normalized center

In the followings, the notions of the "normalized size" and the "normalized center" of a symbol play a crucial role. In the printed documents, the symbols

[4] see the footnote 1 of the Introduction

[5] The reason of this difference is that we used largely the TEX font images to make our recoginition dictionary. The details of our symbol recognition algorithm will be published elsewhere.

on the same level have approximately the same normalized size[6], and in general the symbols on a same level line are arranged in such a way that the normalized centers are aligned straight. These normalizations are defined by the ascending-descending ratio (x, y, z) in fig.4 above; and the visual height h and the visual center c of the given symbol, as follows:

Normalized size (ns):

$$ns = \begin{cases} h \times \dfrac{x+y+z}{x+y} & \text{ascending symbol} \\ h \times \dfrac{x+y+z}{y} & \text{normal symbol} \\ h \times \dfrac{x+y+z}{y+z} & \text{descending symbol} \end{cases}$$

Normalized center (nc) ([2]) [7] :

$$nc = \begin{cases} c + \dfrac{x}{2(x+y)}h & \text{ascending symbol} \\ c & \text{normal symbol} \\ c - \dfrac{z}{2(y+z)}h & \text{descending symbol} \end{cases}$$

3.3 Structural symbols and subformula area

In mathematical formulas, there are some special symbols such as fraction line, integral symbol, root symbol, $\sum$, $\prod$, $\cup$, $\cap$, lim, min, max, etc. which have subareas to accomodate supplementary formulas. These symbols will be called "structural symbol", and the formulas in the subareas will be called "subformula". Note that subformulas can again have structural symbols. This is the reason why we need a recurrent algorithm to recognize mathematical formula.

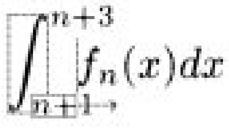


fig.5

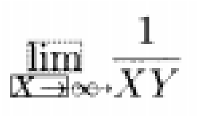


fig.6

The determination of the subformula area is a delicate problem as it is illustrated in fig.5, fig.6 above (see also fig.9 below). In these cases, we take first a temporary subarea and mark it as an "extendable subformula area", while numerator/denominater areas of a fraction are considered as "fixed size subformunla area" for example. The real size of an extendable subformula

[6] There are of course excepional symbols, which must be treated separately.

[7] The notion of "normalized center" can be found in Okamoto [2] where the ratio $x : y : z$ is considered to be $1 : 2 : 1$. In our system, the default value if this ratio is $x : y : z = 15 : 24 : 11$ derived from TEX samples. In order to adapt our system to the other documents (non TEX), an automatic adjusting algorithm of this ratio is also implemented.

area is determined in the recognition process of the subarea called by the recurrent structure of our algorithm. We omit description of our decision criteria for the size of extendable subformula area.

Note that the accent symbols such as tilde, overline, vector symbol, etc. are also considered as structural symbols in our algorithm (with fixed size subformula area).

3.4 Baseline

Given an area in which a mathematical formula is written, the first step is to determine correctly the baseline of the formula. The error of the selection of the baseline breaks the recognition seriously in top-down approach. After checking several evident conditions to determine the baseline, it sometimes still remains undetermined. In such a case, the baseline is decided by the majority of the following list.

1) The normalized center of a symbol at left end of the area.
2) The centerline of the area.
3) The normalized center of the symbol with the largest normalized size.
4) The normalized center of the longest line of fraction.
6) The normalized center of "=", not in a subarea of a structural symbol.
7) The majority of the normalized center of the non-structural symbols with the largest normalized size

3.5 Horizontal connection and subscripts

Each symbol is considered to be on the baseline, if its normalized size is larger than $\frac{2}{3} \times H$ and its normalized center is within the distance

$$d_1 = \min\left\{\frac{x}{2(x+y+z)}, \frac{z}{2(x+y+z)}\right\} \times H$$

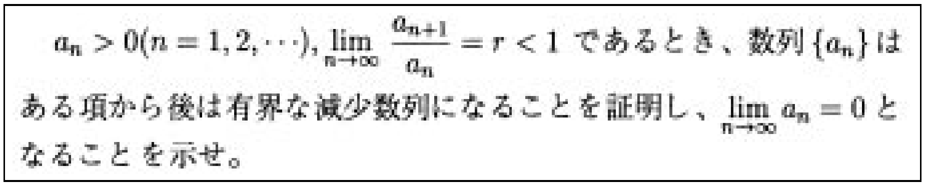
from the baseline, where H is the normalized size of the non-structural symbols on the baseline and (x, y, z) are the ascending-descending ratio.

As for the subscripts, the symbols p between a pair of adjoining symbols on the baseline satisfying the following conditions are considered as subscripts:

$$\begin{array}{rcl} \textbf{subscript symbol} & : & nc(p) > N \text{ and } \alpha \times H < nc(p) - N < \beta, \\ \textbf{superscript symbol} & : & nc(p) < N \text{ and } \alpha \times H < N - nc(p) < \beta, \end{array}$$

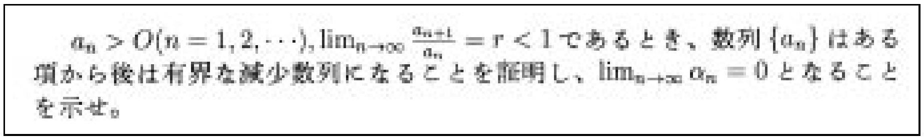
where $nc(p)$ is the normalized center of p, N (resp. H) is the normalized center (resp. size) of the symbols on the baseline and α, β are the threshold defined as follows:

$$\alpha = \frac{t}{2(x+y+z)},$$



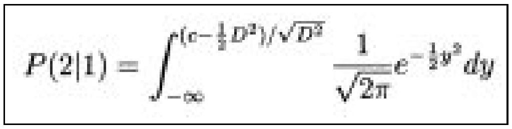
$a_n > 0(n = 1, 2, \cdots)$, $\lim_{n\to\infty} \frac{a_{n+1}}{a_n} = r < 1$ であるとき、数列 $\{a_n\}$ はある項から後は有界な減少数列になることを証明し、$\lim_{n\to\infty} a_n = 0$ となること を示せ。

fig.7(a)



$a_n > O(n = 1, 2, \cdots)$, $\lim_{n\to\infty} \frac{a_{n+1}}{a_n} = r < 1$ であるとき、数列 $\{a_n\}$ はある項から後は有界な減少数列になることを証明し、$\lim_{n\to\infty} \alpha_n = 0$ となることを示せ。

fig.7(b)

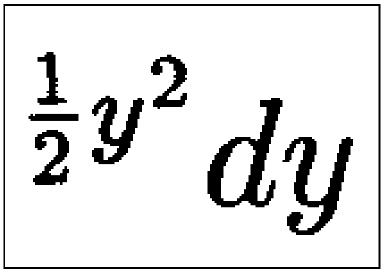


$$P(2|1) = \int_{-\infty}^{(c-\frac{1}{2}D^2)/\sqrt{D^2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} dy$$

fig.8(a)

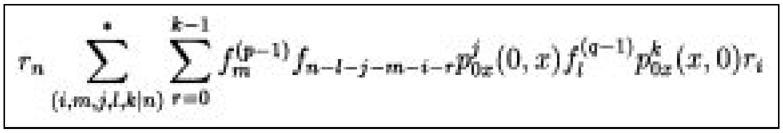
$$P(2|1) = \int_{-\infty}^{(c-\frac{1}{2}D^2)/\sqrt{D^2}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} dy$$

fig.8(b)



$$\frac{1}{2}y^2 \, dy$$

fig.11

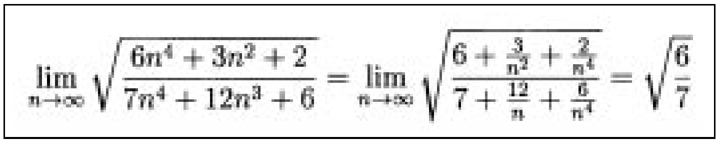


$$r_n \sum_{(i,m,j,l,k|n)}^{*} \sum_{r=0}^{k-1} f_m^{(p-1)} f_{n-l-j-m-i-r} p_{0x}^{j}(0,x) f_l^{(q-1)} p_{0x}^{k}(x,0) r_i$$

fig.9(a)

$$r_n \sum_{(i,m,j,l,k|n)}^{*} \sum_{r=O}^{k-1} f_m^{(p-1)} f_{n-l-j-m-i-r} p_{0x}^{j}(o,x) f_l{}^{(q-1)} p_{0x}^{k}(x,o) r_i$$

fig.9(b)



$$\lim_{n\to\infty} \sqrt{\frac{6n^4+3n^2+2}{7n^4+12n^3+6}} = \lim_{n\to\infty} \sqrt{\frac{6+\frac{3}{n^2}+\frac{2}{n^4}}{7+\frac{12}{n}+\frac{6}{n^4}}} = \sqrt{\frac{6}{7}}$$

fig.10(a)

$$\lim_{n\to\infty} \sqrt{\frac{6n^4+3n^2+2}{7n^4+12n^3+6}} = \lim_{n\to\infty} \sqrt{\frac{6+\frac{3}{n^2}+\frac{2}{n^4}}{7+\frac{12}{n}+\frac{6}{n^4}}} = \sqrt{\frac{6}{7}}$$

fig.10(b)

where $t = z$ (resp. $t = x$) in subscript case (resp. superscript case), and

$$\beta = \frac{h(p)}{2} + \frac{H}{2},$$

where $h(p))$ is the normalized size of p.

The threshold β is defined to prevent subscript and superscript symbols being confused with symbols including subformulas of the structural symbols.

Gathering the subscript symbols between a pair of adjoining symbols on the baseline, we form the subscript area as the circumscribed rectangular box and apply the recognition algorithm recurrently from the determination of the baseline in the subformula area. In case there are both subscript and superscript symbols, the subscript area is divided into the superscript area and the subscript area by the baseline and the recognition algorithm is applied to both areas separately.

A superscript or a subscript formula thus recognized is connected with the nearest symbol on the baseline. The cases connected to the symbol at the right hand side are, for example, combinations or transposed matrices, etc.

4 Experiments and the results

This system was implemented in UNIX-C language on a SUN workstation. In this experiment we input a binary image made from documents printed in TEXformat(A4size,12point) by 400dpi scanning. The recognition results are output in TEXsource file style. In the figures on the previous page, fig.$*$(a) correspond to original images, and fig.$*$(b) are the corresponding recognition results printed using TEX. fig.11 is an enlarged image of the last part of fig.8(a).

References

[1] Richard J.Fateman, Taku Tokuyasu, Benjamin P.Berman, Nicholas Mitchell : "Optical Character Recognition and Parsing of Typeset Mathematics", Computer Science Division, EECS, Dep't, (1995-10)

[2] M.Okamoto, H.Msafiri, "Mathematical Expression Recognition by Projection Profile Characteristics", Trans. IEICE Japan, J78-D-II, No.2, pp.366-370(1995-2)

[3] M.Okamoto, H.Higasi, "Mathematical Expression Recognition by the Layout of Symbols(in Japanese)", Trans. IEICE Japan, J78-D-II, No.3, pp.474-482(1995-3)