A Document Analysis System Based on Text Line Matching of Multiple OCR Outputs

Yasuaki Nakano[1], Toshihiro Hananoi[1], Hidetoshi Miyao[2], Minoru Maruyama[2], and Ken-ichi Maruyama[3]

[1] Kyushu Sangyo University, Fukuoka, 813-8503 Japan
{nakano,hananoi}@is.kyusan-u.ac.jp
[2] Shinshu University, Nagano, 380-8503 Japan
{miyao,maruyama}@cs.shinshu-u.ac.jp
[3] MediaDrive Corporation, Kumagaya, 360-0037 Japan

Abstract. It is well known that integration of multiple OCR outputs can give higher performance than a single OCR. This idea was applied to the printed Japanese recognition and better performance was obtained. In the conventional experiments, however, the zoning, i.e. the extraction of the text region, was done manually and this has been a serious problem from the practical point of view. To solve the problem, an approach to match automatically the classified regions outputted by multiple OCRs was proposed. By the proposed method, a high recognition rate of 98.8% was obtained from OCR systems whose performance is no better than 97.6%.

1 Introduction

It is well known that integration of multiple OCR systems can give higher recognition rate than every single OCR system constituting the integrated system. The idea is not new and a patent claiming the idea was filed in Japan at early 1960's [10]. In the period the idea could not be installed by the cost problem, but the recent progress of the so-called "Software OCR" has made it more practical. Many papers [1-3] describe the effect of the integration. Rice et al. [1] reported the effect of the integration in printed English document recognition using six OCRs with a voting logic. Matsui et al. [3] suggested the same idea in handprinted numeral recognition.

Tabaru et al. reported experimental results of the idea applied to printed Japanese document recognition [4].

In Rice's or Tabaru's systems, the zoning, or the extraction of the text regions to be recognized, was executed manually and the extracted regions were fed to each OCR. In this sense, the applicability of the method has not been large, because of human intervention.

Recently Klink et al. [6] presented a full automated system which can integrate segmentation results using a voting logic. The documents processed in this paper are

S. Marinai and A. Dengel (Eds.): DAS 2004, LNCS 3163, pp. 463–471, 2004.

printed in European languages and the applicability for the Japanese documents is unknown.

In this paper a further experimental result of the majority logic using JEIDA Japanese printed document database is presented. Besides, a method to integrate multiple OCR zoning results is explained and its performance is shown.

2 Majority Logic Using Multiple OCR's

2.1 Tabaru's Results

One of the authors reported the effect of integration of multiple OCR devices using majority logic [4]. In this report, a not large dataset consisting of 1,476 Japanese characters was used. So, intensive experiments using lager datasets may be necessary to ensure the reliability.

2.2 Layout, Typeface, Font Sizes, and Numbering

In this section the result of an intensive experiment is presented using six commercial printed Japanese software OCRs which are listed in **Table 1**.

Table 1. OCRs Used in this Section.

Software Products	Producer Names
Yonde! Koko ver.3™	A.I.Soft™
OmniPagePro 6.0j™	Caere™ and Canon™
Yomitorimonogatari EX v2.5™	Ricoh™
Ninshikikohboh Wide 97™	RIOS SYSTEM™
e.Typyst bilingual 97™	MEDIADRIVE™
WinReader PRO v3.5™	MEDIADRIVE™

Samples of the printed Japanese documents were selected arbitrarily from JEIDA document database [7]. The document IDs and characters included in them are listed in **Table 2**. All of them include Kanji as well as Latin characters. The character numbers are sum of these two kinds.

The sampling pitch used in scanning of the JEIDA database is 600 dpi and the images are fed to each OCR with this resolution. The "zoning" was done manually. In other words, the text regions to be recognized were determined by pointing devices on the screen by an operator.

To apply majority logic to the multiple OCR outputs, the faulty segmentation becomes a severe problem. If there are deleted or inserted characters caused by the faulty segmentation in some OCRs, the correspondence between different OCRs will give meaningless results. In many Japanese characters, especially kanji, have several components (radicals) in them. So, sometimes a character is divided into two or more simple characters. On the contrary, two simple characters are merged and segmented as a single complex character. By the reason, deletion or insertion occurs rather frequently and the matching character strings of different lengths is very important.

Table 2. Document Samples Used.

Document Ids	Category	Characters
P010102 and others	Newspapers	10,915
P020101 and others	Novels	5,495
P030101 and others	How-to books	3,339
P040101 and others	Elementary School Textbooks	2,326
P070101	Manuals	289
Total: 25 documents		22,364

Since Japanese sentences are printed without spaces between words, the correct correspondence should be taken not among the words but among the printed lines. A method used by Tabaru [4] to find the best match between lines resembles to the dynamic-programming-based string matching algorithm [8], though not exactly same. In this paper, Tabaru's method is adopted also.

In the experiment, six OCR recognition results were compared after the adjustment by the matching. A forced decision scheme is adopted in each OCR and the first candidates made by all OCRs were treated as the results and no rejections were considered.

Table 3. Experimental Results: the Numbers of Errors and Error Rate.

OCR	Substitutions	Insertions	Deletions	Recognition Rate (%)
A	541	77	6	97.2
B	1,349	103	46	93.3
C	442	31	23	97.8
D	453	48	13	97.7
E	413	17	11	98.0
F	373	124	24	97.7
Majority	112	15	0	99.4

Simple majority logic was used to give the final decision. When three-three or two-two-two ties were found, the earliest result was adopted. Since the forced recognition mechanism was adopted, the errors were classified into substitution, insertion and deletion. No rejection scheme is used.

The recognition errors of six OCRs and the virtual recognition system by the majority logic are shown in **Table 3**. Number of total recognized characters are 22,364 as shown in **Table 2**. In the calculation of recognition rates, the sum of three type errors divided by total number is subtracted from 100%.

As shown in **Table 3**, it is obvious that the majority logic yields a very good performance. Especially the improvement in deletion is remarkable.

3 Automatic Matching of Text Regions

3.1 Practical Viewpoints

Commercial OCR software can analyze a document image and segment it into the layout structures consisting of texts, mathematical expressions, figures, tables, photo-

graphs, captions etc. Logic structure extraction, e.g. header/footer/title/reference identification, is one of research targets now.

As stated in the previous chapter, many reports on the integration of multiple OCRs seem to be based on the manual zoning. In order to bring the integration technique into practical use, it may be necessary to avoid or minimize human intervention.

A fully automated integration of several OCR outputs, therefore, require matching extracted regions which are labeled differently or disorderedly. Some regions may be deleted, merged or divided into several regions with some OCRs.

So far, Kling et al. [4] propose a technique called MergeLayouts to integrate faulty segmentation made by multiple OCRs. Their method is for European languages, and can use the fact that words are printed with spacing between them.

On the contrary, Japanese sentences are printed without spaces between words and a new technique will be necessary. Furthermore MergeLayouts uses font and size information outputted by OCRs. But it is not common to output such information in Japanese OCRs. To develop a method to integrate OCR outputs only by character codes is another purpose of this paper.

3.2 Difficulties in Region Matching

In this research, text regions are only considered and figures and/or photographs are not taken into account. If a nontext region, say a photograph, is recognized as a text region by all OCRs, the outputted texts may be integrated. But such a case does not seem to occur frequently and the misrecognized text regions exist in only a part of OCR set. As a result, we can suppose that all regions successfully matched among many OCRs are text ones.

Under the consideration above, regions to be matched consist of text lines. So, the matching must be done on the basis of character string correspondence.

By observing the outputs from printed Japanese OCRs, the problems stated below are noted.

- A line sometimes divided into multiple text regions, especially when a large space area exists in it.
- The sequence of the text regions differs OCR by OCR. This phenomenon occurs notably when vertical and horizontal lines exist in a document simultaneously. (In Japanese documents, it is not rare to mix horizontal and vertical lines in a page.)
- In extreme cases, column extraction fails and several text lines in different columns are merged into one horizontal line.
- Captions of figures/tables, headers and footers are sometimes recognized as a part of another region.

From these problems caused by the document image analysis, it is not easy to match same text regions on document images. An example of erroneously analyzed documents is shown in **Figure 1**. Furthermore, in the character recognition subsystem, deletion or insertion occur frequently as stated in the previous section.

3.3 Text Matching Algorithm

In this research, text region matching is done on the basis of line similarities.

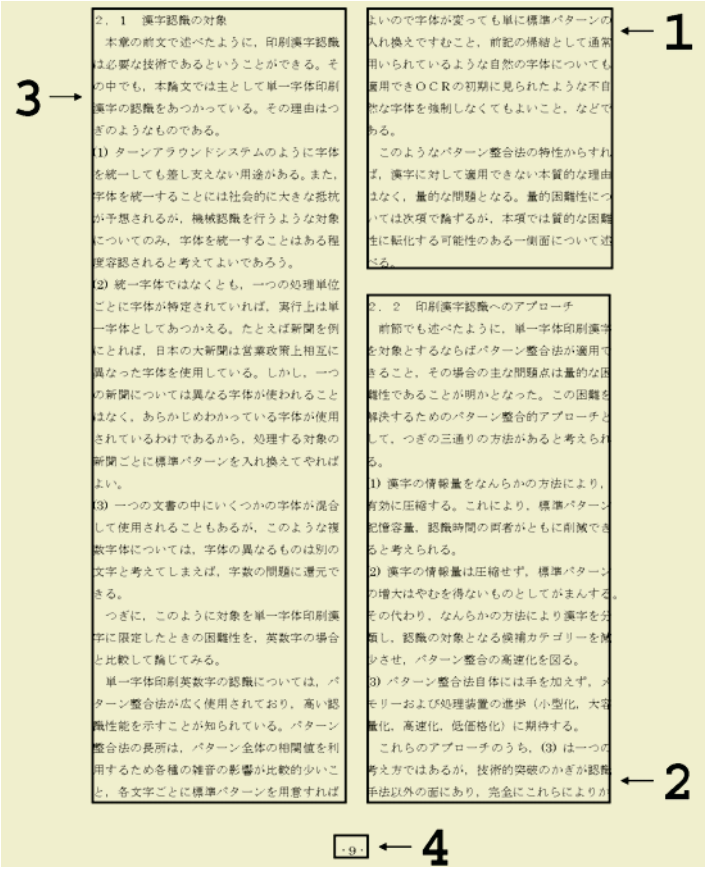

Fig. 1. An example of erroneously analyzed document images: the sequence of the regions is not considered appropriate.

Distance of two lines is calculated by the dynamic programming (DP) method [8]. DP matching algorithm is widely used in string matching, speech recognition and character recognition, so it is explained very simply here.

Denote two text lines as character strings $\{a_1, a_2, ..., a_m\}$ and $\{b_1, b_2, ..., b_n\}$. The distance $d(i, j)$ between two character substrings $A_i=\{a_1, a_2, ..., a_i\}$ $(i= 1 \text{ to } m)$ and $B_j=\{b_1, b_2, ..., b_j\}$ $(j = 1 \text{ to } n)$ when two substrings are best fitted is evaluated iteratively as follows:

$$\text{initial values:} \begin{cases} d(0,0) = 0 \\ d(i,0) = d(i-1,0)+1, \text{ for } 1 <= i <= m \\ d(0,j) = d(0,j-1)+1, \text{ for } 1<= j <= n \end{cases}$$

$$\text{iteration:} \qquad d(i,j) = \min \begin{cases} d(i-1,j-1)+h_{ij} \\ d(i,j-1)+1 \\ d(i-1,j)+1 \end{cases}$$

where h_{ij} denotes the distance between characters a_i and b_j; $h_{ij} = 0$ when two characters are same and $= 1$ otherwise. Based on the iteration the distance between two total substrings A_m and B_n is calculated.

3.4 Construction of a Standard OCR

Based on the distance of text lines, distance between two text regions is calculated. The following is simplified explanation of the procedure.

First it is necessary to determine a "standard OCR" as the most highly supported one by other OCRs. The standard OCR is determined as follows. Denote two OCR as OCR(i) and OCR(j). In this process, all lines in all regions are collected from OCR outputs and numbered arbitrarily.

1. Calculate distance d(k) between line 1 of OCR(i) and line k of OCR(j).
2. The line k-min of OCR(j) whose distance is smallest from line 1 in OCR(i) is allotted to the line 1 in OCR(j). When the smallest distance is greater than a threshold, line 1 of OCR(j) is not determined.
3. Similarly, for all lines in OCR(i) corresponding lines in OCR(j) is determined.
4. For every OCR(i) the value of distance for all line pairs between two OCR is evaluated and average distance by dividing line numbers and is calculated.
5. After the calculation of the averaged distance of every OCR from others, an OCR which has the smallest average is set as the standard OCR.

3.5 Judgment of Analysis Failure by Text Line Sorting

Document analysis result of each OCR is checked by a human operator if it corresponds to the human judgment. The results are compared with those of the automatic region correspondence method.

Automatic region correspondence is done only by the text line sequence. First the standard OCR is constructed by the method stated in the previous section. Then the text lines in outputs of other OCRs are sorted according to those of the standard OCR using the text line similarity.

When all lines can be sorted according to those of the standard OCR, the region correspondence is considered as a success. Otherwise, the analysis is considered as a failure.

3.6 Voting

The method to vote OCR results at character level is almost same as the one used in section **2.2.** In this case, however, a rejection scheme is introduced.

Using the standard OCR as a criterion, all lines in another OCR can be sorted based on the correspondence of each line before the majority logic is applied.

4 Experimental Results

4.1 Environment

In the experiment explained in this chapter, five software OCRs listed in **Table 4** (from A to E) are used. OCR F is omitted because it was supplied by the same maker as OCR E.

For a part of the documents in JEIDA'93 which were analyzed successfully by our method, the voting method was applied to the OCR results. The performances were almost as same as those in **Table 3**.

4. 2 Environment Document Samples

As same as **Section 2.2**, JEIDA'93 was used and 77 images were selected from the database. In the dataset, 51 images contain figure and tables and 26 do not contain them. Since JEIDA'93 includes very tough samples from the document analysis viewpoint, easier samples were added. As easier samples, 50 academic papers, some of which contain figures, were used.

4.3 Experimental Results

Table 4 shows an experimental result of region recognition of each OCR. OCRs from A to E are five OCRs used in the experiment, but the order has nothing to do with that in **Table 3**. In this table "Failure" means that the final extraction of regions failed or the erroneous sequence of regions was obtained. In **Table 4**, OCRs A-E do not coincide with the sequence in **Table 3**.

Table 4. Failures in Region Recognition.

OCR	JEIDA'93 without Figures	JEIDA'93 with Figures	Academic Papers	Total
A	10	4	4	18
B	14	8	4	26
C	7	1	10	18
D	14	10	20	44
E	9	4	6	19
Total	51	26	50	127

Table 5 shows the failures in the virtual OCR which integrates the results of OCR A-E by the proposed method.

From **Table 4**, it is evident that some OCRs are poor in the region recognition.

Table 5 shows that the proposed method gives good performances and it is promising for simple structure documents such as academic papers.

In this experiment, the total number of characters is 4,813. In the calculation of the recognition rate, rejection is weighed as 1/2 of substitution.

Table 5. Failures in Region Recognition by the Proposed Method.

OCR	JEIDA'93 without Figures	JEIDA'93 with Figures	Academic Papers	Total
Our Method	3	7	2	12
Total	51	26	50	127

Figure 3 shows a sample of recognition results by the proposed method. Here, R shows that the input character is rejected and candidates for the rejected character are shown in braces.

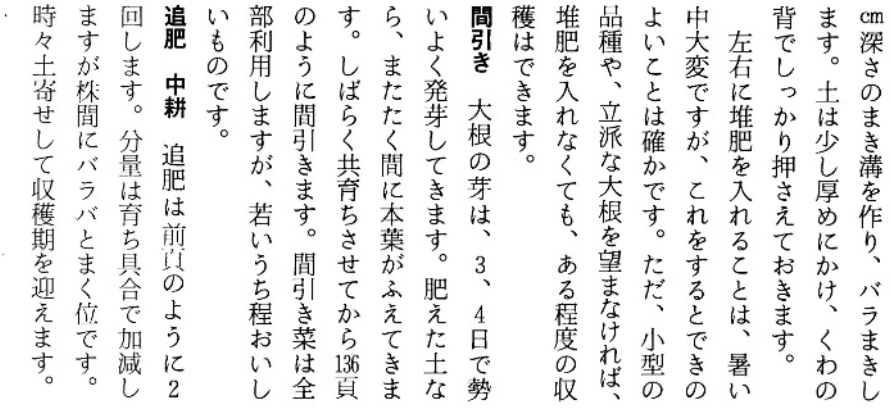

Fig. 2. A region in an input image (JEIDA'93) extracted by OCR's and successfully matched by the proposed algorithm.

5 Concluding Remarks

In this paper we showed that integration of multiple OCR results is useful both at the document analysis stage and the character segmentation and recognition stage.

A simple method to integrate the faulty document analysis and character segmentation and recognition is proposed in this paper. An experimental result is shown using commercial OCRs and printed Japanese documents, which include JEIDA'93 and academic papers.

The proposed procedure to match the regions outputted from multiple OCRs seems to work well for simply structured documents. The character recognition rate in the successfully analyzed text regions is improved from 92.1-97.6 % to 98.8% and proved the high performance of the proposed method.

In order to make the proposed method more robust, more sophisticated matching process between regions will be necessary, especially when figures and tables are included in the documents.

The idea of integrating multiple OCRs is applicable to wide area. Some of the authors have already reported an approach using the resembled principle to English cursive handwritten word recognition [9].

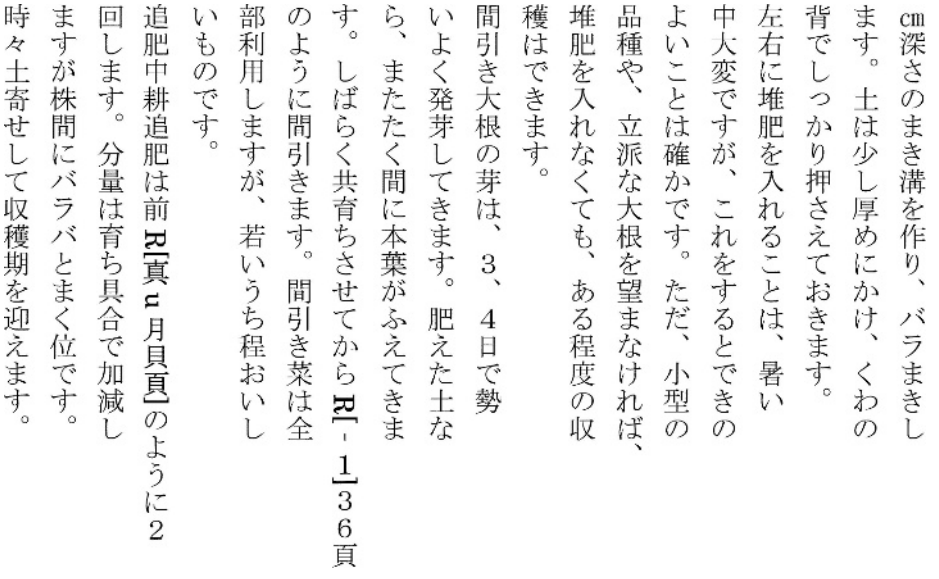

Fig. 3. Recognition result of the region shown in Fig. 3. "R" marks in the recognition result show rejection errors and the characters in the braces are candidates for each rejection.

The authors appreciate the contribution of Messrs. Hirosato Tabaru and Atsuhiko Tani of the Graduate School of Shinshu University.

References

1. Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121
2. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.): Theoretical Aspects of Computer Software. Lecture Notes in Computer Science, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (1997) 415–438
3. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)
4. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)