Extraction of inclined Character Strings from Unformed Document Images Using the Confidence Value of a Character Recognizer

Takizawa, Kei
Faculty of Engineering, Kyoto University

Arita, Daisaku
Faculty of Engineering, Kyoto University

Mino, Michihiko
Faculty of Engineering, Kyoto University

Ikeda, Katsuo
Faculty of Engineering, Kyoto University

http://hdl.handle.net/2324/4228

出版情報：IEICE transactions on information and systems. E77-D (7), pp.839-845, 1994-07. The Institute of Electronics, Information and Communication Engineers
バージョン：
権利関係：

PAPER Special Issue on Document Analysis and Recognition

Extraction of Inclined Character Strings from Unformed Document Images Using the Confidence Value of a Character Recognizer

Kei TAKIZAWA[†], Daisaku ARITA[†], *Nonmembers*, Michihiko MINOH[†] *and* Katsuo IKEDA[†], *Members*

SUMMARY A method for extracting and recognizing character strings from unformed document images, which have inclined character strings and have no structure at all, is described. To process such kinds of unformed documents, previous schemes, which are intended only to deal with documents containing nothing but horizontal or vertical strings of characters, do not work well. Our method is based on the idea that the processes of recognition and extraction of character patterns should operate together, and on the characteristic that the character patterns are located close to each other when they belong to the same string. The method has been implemented and applied to several images. The experimental results show the robustness of our method.

key words: *string segmentation, string extraction, layout analysis*

1. Introduction

There are research systems that automatically generate indices of document images for retrieval. Such systems are intended to help us select necessary information quickly from enormous amounts of information on document images. In such systems, it is essential to be able to recognize characters in the documents.

Most of the research [1]–[3] only deals with formed documents, such as visiting cards, papers or order forms, which have only horizontal or vertical character strings.

However, there exist another type of documents which have no specific form at all, for example, catalogs or posters. They often include inclined character strings. We call this type of documents *unformed (free-formed) documents.* A method for dealing with such documents has been proposed [8]. In this method, by applying the Hough transformation to the centroids of connected components, character patterns lying along the same straight line are extracted. This method is suitable for processing images only including simple character patterns such as letters, which are mostly composed of one connected component. This is because the connected components belonging to the same character string are almost collinear.

However, using the method in Ref. [8], it is difficult to deal with complex character patterns such as Chinese characters, which are composed of several connected components. This is because the connected components belonging to the same string are distributed around a straight line on which the character patterns are aligned.

Instead, we propose a new method for processing images including complex character patterns like Chinese characters.

In the proposed method, the connected components are merged into character strings, using the characteristic that the character patterns are located close to each other when they belong to the same string and apart when they do not.

We also deal with the string segmentation problem peculiar to the Chinese character strings. The character strings can not be segmented into character patterns correctly, only using the features such as the width of the patterns. The character recognizer is used to determine whether a pattern is really a character pattern or not.

2. Extraction of Character Strings

To avoid unessential problems in our scheme, we constrain the input image to satisfy the following requirements: (1) the fonts should not be peculiar ones such as italics; (2) the graphics and pictures should be removed beforehand; (3) the character patterns belonging to the same character string should lie along the same straight line; (4) the character patterns belonging to the same character string should have the same orientation.

The proposed method consists of the following two modules:

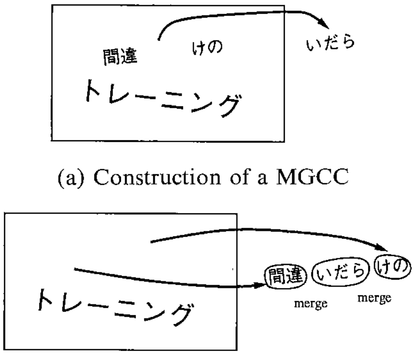
Module-A: Construction of a merged group of connected components (MGCC)
Module-B: Extension of the MGCC

Notice that module-B is not executed after module-A, but called in module-A as a subroutine.

Module-A extracts a part of a group of connected components belonging to the same character string (see Fig. 1(a)). We refer to such a group of connected components as an MGCC (merged group of connected components). Module-A calls module-B, passing the MGCC as an argument. And then, module-B searches the image for the connected components with which the

Manuscript received November 24, 1993.
Manuscript revised March 9, 1994.

[†] The authors are with the Faculty of Engineering, Kyoto University, Kyoto-shi, 606–01 Japan.



(a) Construction of a MGCC

(b) Extension of the MGCC

Fig. 1 Component modules of the proposed method.

MGCC forms a character string and merges them with the MGCC (see Fig. 1(b)). When module-B finishes, module-A is resumed and a new MGCC is generated, and so on. This continues until the input image becomes empty.

In the following two sections, we describe each module in detail.

3. Construction of a Merged Group of Connected Components

In module-A, the connected components belonging to the same character string are extracted and merged together.

If two character patterns are adjacent to each other in the character string, they lie close to each other. On the basis of this property, an MGCC is generated using the expansion operation.

By applying the expansion operation to the image repeatedly, any two of the connected components could meet together. The closer they are located to each other, the earlier they meet together.

Because the expansion operation changes the original image, another image, *the expansion image*, is used, copying the original image into the expansion image. The expansion operation is applied repeatedly to the expansion image until two or more connected components get together and form one new connected component. This new connected component is removed from the expansion image and the corresponding connected components are extracted from the original image as an MGCC. And then, this module calls module-B, passing the MGCC as an argument.

When module-B finishes, the expansion operation is applied to the image repeatedly again. If the inclination of the character string has not been determined in module-B, because the MGCC is too short, the MGCC is restored to the original image before applying the expansion operation. At the same time, the connected component corresponding to the MGCC, which has been removed from the expansion image, is restored to the expansion image.

And then, a new MGCC is generated through the expansion operation, and so on. This continues until the original image becomes empty.

The algorithm of module-A is summarized as follows:

```
Procedure module-A()
    Mg : variable for storing an MGCC;
    orgImg : original image;
    exImg : expansion image;
begin
    while orgImg is not empty do
    begin
        expand exImg;
        if a new MGCC is generated
              then store the MGCC to Mg;
              else goto 10;
        module-B(Mg);
        if the inclination of Mg has not been determined
              then restore orgImg and exImg to the state before
Mg was extracted;
label 10;
    end
end
```

4. Extension of the MGCC

Module-B consists of the following modules.

Module-B.1: Determination of the inclination of the MGCC
Module-B.2 : Segmentation of the MGCC
Module-B.3 : Reconstruction of the MGCC

In module-B.1, the inclination of the MGCC is determined. In module-B.2, the MGCC is segmented into character patterns and recognized. In module-B.3, the original image is searched for the connected components with which the MGCC is merged.

Module-B is composed in the following way:

```
Procedure module-B(Mg:MGCC)
begin
    module-B.1(Mg);
    if the inclination is not determined
        then goto 20;
    while for ever do
    begin
        module-B.2(Mg);
        module-B.3(Mg);
        if Mg has been merged with no connected component
then output Mg as a character string and goto 20;
        module-B.1(Mg);
    end
label 20;
end
```

4.1 Determination of the Inclination of the MGCC

To align the MGCC with the horizontal axis, the angle of its inclination θ_{group} shown in Fig. 2 is determined.

The character patterns are aligned along a line l_1 in Fig. 2.

To determine θ_{group}, the convex hull of the MGCC, which is the smallest polygon surrounding the MGCC, is used.

The convex hull of the MGCC has two long sides which are nearly parallel with each other as shown in Fig. 3. Each side is almost parallel to the line l_1, that is, the angle of each side with respect to the x axis is nearly equal to the angle θ_{group}.

To determine the angle θ_{group} from these sides, a function $A(\theta)(\theta = 0, 1, \ldots, 179)$ is defined so that $A(\theta)$ is the sum of the length of the sides of the convex hull, whose angle with respect to the x axis is θ.

The two long sides are not exactly parallel. And, it is enough to calculate the angle θ_{group} to an accuracy of the order of ten degrees. This is based on the definition of the feature vector of the character recognizer [7]. The feature vector is obtained from the character strokes whose directions are classified into four categories: 0°, 45°, 90°, 180°. The resolution of the direction is rough enough to accept errors of approximately ten degrees, and our experiment confirmed that a character pattern inclination of approximately ten degrees does not affect the performance of the character recognition process.

Thus, the following function $B(\theta)$ is used:

$$B(\theta) = \sum_{\theta'=\theta_s}^{\theta_e} A(\theta') \tag{1}$$

where $\theta_s = (\theta + 170)\ mod\ 180$, $\theta_e = (\theta + 10)\ mod\ 180$.

As $B(\theta)$ has the maximum value around $\theta = \theta_{group}$ owing to the two long sides, we can determine θ_{group} approximately by finding θ_{max} for which $B(\theta)$ takes on its maximum value.

It sometimes happens that the extracted MGCC is a part of a character pattern or a single character pattern. In this case, $B(\theta_{max})$ is not extremely large in comparison with $B(\theta)(\theta \neq \theta_{max})$. Therefore, $B(\theta_{max})$ should be compared with the threshold T_θ, which is 2/3 of the perimeter of the convex hull of the MGCC, so as to determine whether θ_{group} obtained from θ_{max} is valid or not.

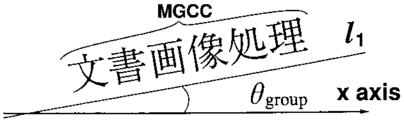


Fig. 2 Inclination angle of the MGCC.

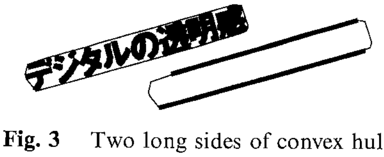


Fig. 3 Two long sides of convex hull.

The coefficient 2/3 is based on the assumption that a character string consists of at least two character patterns. When an MGCC consists of two square-shaped character patterns, its convex hull is a rectangle such that the ratio of its longer side to its shorter one is greater than 2 : 1. The sum of the length of its two longer sides is longer than 2/3 of the perimeter of the convex hull.

Thus, the angle θ_{group} is determined as follows:

```
Procedure module-B.1(Mg:MGCC)
begin
    calculate $A(\theta)$ and $B(\theta)$;
    if $B(\theta_{max}) < T_\theta$
        then $\theta_{group}$ is not determined;
        else find the range of $\theta$ in which $B(\theta) \geq T_\theta$ (if two ranges are found, adopt the smallest range including both of these ranges), and set the angle $\theta_{group}$ to the middle point of this range;
end
```

4.2 Segmentation of the MGCC

The MGCC segmentation problem is transformed into the block merging problem in the following way.

Let z axis be the axis that forms an angle θ_{group} with the x axis. And let $H(z)$ be the projection profile of the MGCC on the z axis. By dividing the MGCC at the intervals whose values of the projection profile are zero, *blocks* are obtained. As shown in Fig. 4, the blocks are assinged symbols $r_1, r_2, \ldots, r_m$ from left to right where m is the total number of the blocks. Each block is either a whole character pattern or a part of it. This is because the zero intervals of $H(z)$ represent the following two types of white spaces: (1) one separating two character patterns which are adjacent to each other; (2) one separating one character pattern into several parts. By merging the blocks properly, we can obtain the character patterns in the MGCC.

Our method has the following features.

First, the character recognizer is used to determine whether a pattern is really a character pattern or not.

It is difficult to segment a Chinese character string including letters and digits. Most of the Chinese characters consist of several narrow connected components. Besides, the letters and digits included in such a string, are usually narrower than the Chinese characters. Hence, it is difficult to distinguish a letter and digit from a part of a Chinese character.

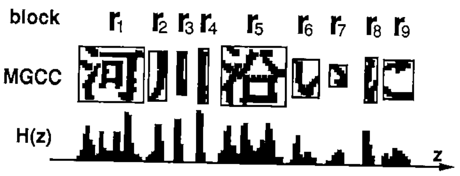


Fig. 4 Division of the MGCC.

Methods to deal with such a string have been proposed in Refs. [5] and [6]. In these methods, it is assumed that the width of letters and digits is almost half of that of Chinese characters. The former [5] uses only the features of the rectangle bounding a connected component. The latter [6] uses a character recognizer as well, in order to classify narrow patterns into three categories, letter, digit and part of Chinese character.

Our method is not based on any assumption about the width of letters and digits, and make use of the character recognizer more effectively. We define a criterion to determine whether a pattern is really a character pattern or not, using the output of the character recognizer. We call this criterion the *confidence value.*

Second, the block merging process is not carried out from one end block to the other end block. The block, from which the process is carried out in both directions to the end blocks, is determined on the basis of the confidence value. This is because the character pattern at each end of the MGCC may be missing a part of it, the part remaining in the original image.

Third, the orientation of the character patterns is determined. There are four orientations of character patterns as shown in Fig. 5. One out of these orientations is selected.

4.2.1 Confidence Value of Character Recognition

The character recognizer operates using the directional element features [7]. The confidence value is defined using the output of the character recognizer .

The character recognizer generates up to the n-th character candidate code word, $C_1, C_2, \ldots, C_n$ with distance, $D_1, D_2, \ldots, D_n$, where D_i represents the city block distance between the input pattern and the standard pattern of C_i in the feature space.

The confidence value is defined using D_1 and D_2. Figure 6 shows the distributions of the term $D_2 - D_1$. In the figure, A represents a set of character patterns whose first character candidates are correct, and B represents a set of character patterns which are parts of character patterns or include two or more character patterns. These patterns were extracted from a digitized document through a string extraction process and string segmentation process, and then recognized. The vertical axis represents the percentage of the patterns, and the horizontal axis represents $D_2 - D_1$. The number of patterns which belong to set A and that of set B are 657 and 397, respectively. Notice that the values of $D_2 - D_1$ are multiplied by a constant. Set B is distributed around the low value of $D_2 - D_1$, whereas set A is distributed around the high value.

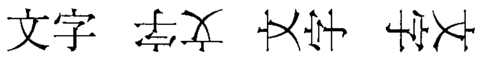


Fig. 5 Four orientations of character pattern.

This experimental result shows that if a character pattern, i.e. neither a part of a character pattern nor a pattern including two or more character patterns, is recognized by the character recognizer, $D_2 - D_1$ tends to grow.

The term $D_2 - D_1$ is useful to define the confidence value. In Ref. [9], we defined the value as follows:

$$c_{old} = \frac{D_2 - D_1}{D_2} \times 100. \tag{2}$$

In the definition, $D_2 - D_1$ is divided by D_2, so that the confidence value ranges from 0 to 100.

However, there are a lot of groups of character patterns which are similar to one another, for example, "は","ば" and "ぱ". The confidence value of such a character pattern tends to be small, because the first and second character candidates are similar to each other.

From the experiments, we found that when no consideration was given to the existence of such groups of character patterns, the confidence value definition causes string segmentation errors [9].

Hence, we change the definition of the confidence value to

$$c = \frac{D_i - D_1}{D_i} \times 100 \tag{3}$$

where i is the minimum subscript for which C_i is not similar to the first character candidate. We define

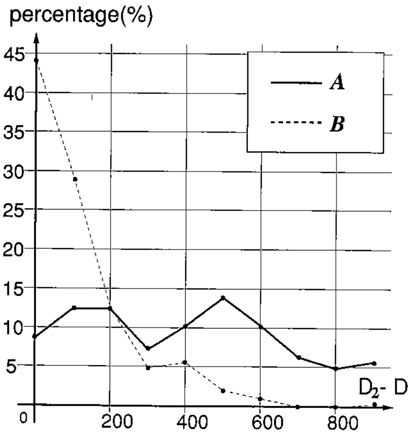


Fig. 6 Distribution of $D_2 - D_1$.

Table 1 Examples of groups of similar character patterns.

No.	Group
1	あぁ
2	c C
3	はばぱ
4	O o 0 〇○
5	I | 1 1
6	ーー一—

seventy groups of character patterns which are similar to one another. We show a part of the groups in Table 1.

4.2.2 Block Merging Process

The group of blocks, from which the merging process is started, is referred to as *the standard block.* This block is obtained in the following way.

Let r_s be the block including the largest number of black pixels. The merged blocks, r_s, $r_{s-1}r_s$, r_sr_{s+1}, $r_{s-1}r_sr_{s+1}$, $r_sr_{s+1}r_{s+2}$ and $r_{s-2}r_{s-1}r_s$ are generated. Among these blocks, the merged block with the maximum confidence value is selected as the standard block, and the standard width w_{stnd} is set to the width of the standard block.

Since the confidence value c is calculated without considering the width of the block, the width should be considered for the block merging process. Thus, a *confidence value for the block merging process* c' is defined as

$$c' = c \times f\left(\frac{w}{w_{stnd}}\right). \tag{4}$$

Where

$$f(x) = \begin{cases} \frac{x+1}{2} & (0 \leq x < 1) \\ -x+2 & (1 \leq x \leq 2) \\ 0 & (x < 0, x > 2) \end{cases}$$

and w is the width of the block. As the width of the block w approaches w_{stnd}, $f(\frac{w}{w_{stnd}})$ increases.

From the standard block, the block merging process is carried out in the following way.

Figure 7 shows the MGCC during the block merging process. The process is being carried out from the standard block towards the right side. Let r_i be the block which is adjacent to the standard block. The group of blocks having maximum confidence value c' is selected as a character pattern from the groups, r_i, r_ir_{i+1}, $r_ir_{i+1}r_{i+2}$, $\cdots$ and $r_i \cdots r_{i+l+1}$ where l is the maximum integer for which the width of $r_i \cdots r_{i+l}$ is shorter than w_{stnd}. In the case of Fig. 7, the group of blocks is selected from r_6, r_6r_7, $r_6r_7r_8$, $r_6r_7r_8r_9$, $r_6r_7r_8r_9r_{10}$. And then, the process continues from the end of the selected group of blocks.

The left side of the standard block is processed in the same way.

As mentioned before, there are four orientations of character patterns. Hence, the block merging process is carried out with respect to each orientation. Four groups of extracted patterns, which are corresponding to four orientations, are obtained. And then, the total confidence values of the groups are calculated. The total confidence value C_{total} is defined as follows:

$$C_{total} = \frac{1}{n}\sum_{i=1}^{n}(c'_i)^2 \tag{5}$$

where n and c'_i denote the number of the patterns extracted from the MGCC, and the confidence value of the i-th pattern from the left, respectively.

The group having maximum total confidence value is selected.

4.3 Reconstruction of the MGCC

The MGCC is merged with the connected components meeting the following conditions: (1) 90% of the area is included in the region shown in Fig. 8 with half tone (the two straight lines are the tangents of the MGCC and parallel to its direction); (2) the distance from either end of the MGCC is less than T_r times as long as the average width of the character patterns which are extracted from the MGCC.

5. Experimental Results

The proposed method was implemented on a SUN SPARC workstation in C language. The test images were digitized with a resolution of 300 dpi. The threshold T_r was set to 3, which was experimentally determined.

Figure 9 shows an example of an unformed document image. Notice that in the figure, some of the character strings are numbered for the following explanation. The proposed method was applied to this image, and most strings were correctly extracted and divided into character patterns. However, some errors were observed at underlined places in Table 2. The errors are due to the inadequacy of the merging process in module-B.3. In module-A, the connected components included in string 4, "坂上是則", met together earlier than those in string 3, "あさぼ...", extracted as an MGCC. There is a character pattern "月", which is contained in string 3, in the direction of this MGCC inclination, merged with the MGCC in module-B.3.

We also applied the method from Ref. [9] to this test image, and the results are shown in Table 2, as well.

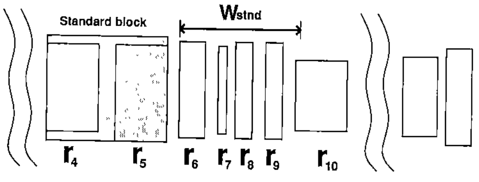


Fig. 7 Block merging process.

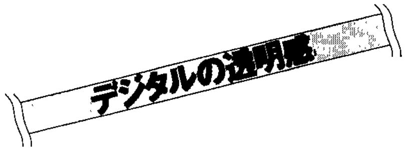


Fig. 8 Reconstruction of the MGCC.

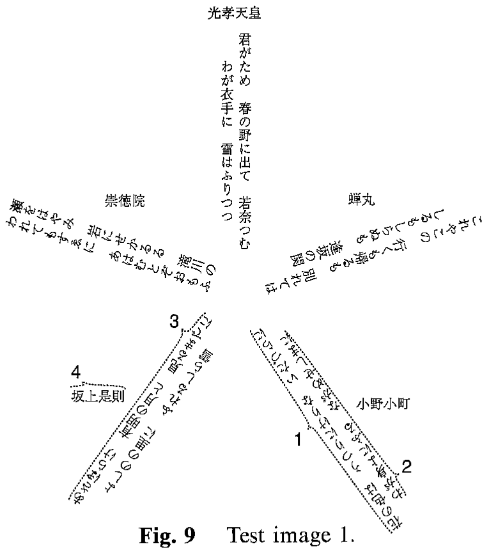


Fig. 9 Test image 1.

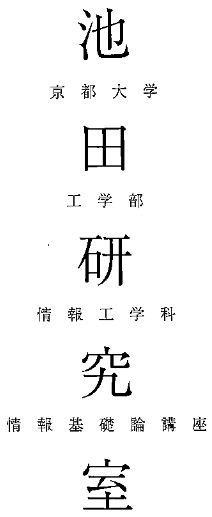


Fig. 10 Test image 2.

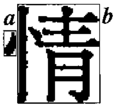


Fig 11 Oversegmented character pattern.

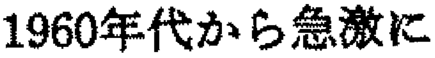


Fig 12 Test image 3.

Table 3 Effect of the new confidence value.

Result A	１９６０年代から急激に
Result B	ユ０年代から急激に

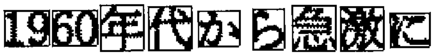


Fig 13 String segmentation result by the method not using the confidence value.

Table 2 Results of applying our method to test image 1.

String no.	First character candidates
1	花の色$\overline{\text{はう}}$つりにけり$\overline{\text{ない}}$たづらに
2	わが身よにふる$\overline{\text{な}}$がめせしまに
3	あさ$\overline{\text{ぼ}}$らけ有明の_と見るまでに
4	坂上是則$\underline{\text{ァ}}$

The method adopts definition (2) as a confidence value. Errors were observed at the overlined places as well as at the underlined ones.

The proposed method was then applied to the image shown in Fig. 10. This image includes a vertical character string and horizontal character strings. The horizontal character strings are located in the character-to-character spaces of the vertical character string.

By applying the expansion operation to the image repeatedly, the connected components included in the horizontal character strings are extracted together earlier than the ones included in the vertical character string. Thereby, after the horizontal character strings were extracted, the vertical character string was collectly extracted.

The character strings were segmented correctly except for the character string "情報基礎論講座". The leftmost character pattern "情" were oversegmented into two patterns, which are labeled *a* and *b* in Fig. 11. The patterns *a* and *b* were recognized as " J " and "清", respectively. Since the pattern *b* is similar to the character pattern "清", its confidence value is larger than that of the pattern "情" including *a* and *b*. Thus, the pattern labeled *a* and *b* were extracted as separate character patterns.

Finally, we show an example of experimental results, which show the effect of the new confidence value (3). The proposed string segmentation method and the one proposed before[9] were applied to the image shown in Fig. 12. The results are shown in Table 3. Result *A* and *B* are corresponding to the new method and the old one, respectively. In result B, the pattern " 1 9 6 " were extracted as a character pattern, which was recognized as "ユ". The confidence value of the pattern " 1 " was small, because the first and second character candidate of the pattern were " 1 " and " l ", respectively. This led to a string segmentation error.

The definition of the similar character pattern group 5 in Table 1 increased the confidence value of the pattern " 1 ". Thereby, the character patterns were extracted correctly in result A.

Figure 13 shows the results obtained by applying the method only considering the width of patterns. The numeral patterns were not extracted correctly. This is because the method is based on the assumption that a character pattern is square-shaped.

6. Conclusion

A new algorithm which is able to extract and recognize character strings from an unformed document image has been proposed. The algorithm consists of two parts.

Module-A:Construction of a merged group of connected components(MGCC)
Module-B:Extension of the MGCC

Module-A extracts a part of a character string as a merged group of connected components(MGCC). An MGCC is generated by gathering the connected components which are close to each other through the expansion operation.

Module-B merges the connected components left in the image with the MGCC, so that the MGCC forms a complete character string. This module also includes the process segmenting the MGCC into character patterns. The process is carried out using a confidence value of the character recognizer as a criterion to determine whether a pattern is really a character pattern or not.

The proposed method was applied to several images and the experimental results show that almost all the strings and characters are extracted and recognized correctly. This proves the robustness of our method.

However, the proposed method has the following problem.

Errors could occur in the case where the character strings are located close to each other. The connected components, which lie close to each other though they do not belong to the same character string, could be mismerged in both modules, A and B. They could be joined together by the expansion operation, and determined to belong to the same character string, in module-A. They could be merged together in module-B, if the threshold T_r is set to a large value.

This problem is left for the future study.

References

[1] Ohsumi, J., Shimizu, N., Nakamura, Y., Itonori, K. and Miyake, H., "Document Image Recognition and Reuse System," *IEICE Technical Report*, PRU91-27, 1991.

[2] Kakeno, H., Sugie, N., Nozawa, S., Takeshita, T. and Inagaki, H., "A Segmentation Method for Document Image Using Fusion of the Image," *IEICE Technical Report*, PRU91-7, 1991.

[3] Luo, Q., Watanabe, T. and Sugie, N., "Structure Recognition of Newspapers Using Production System," *IEICE Technical Report*, PRU91-8, 1991.

[4] Hirazawa, H. and Sato, M., "Document Image Analysis by Maximal Stroke Extraction Method," *IEICE Technical Report*, IE88-136, 1989.

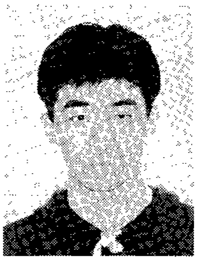
[5] Sato, M. and Kida, H., "A Method of Character Extraction from Printed Documents Including Variable Character Spacing," *IEICE Technical Report*, IE88-138, 1989.

[6] Akiyama, T., Naitou, S. and Masuda, I., "A Method of Character Extraction from Printed Documents Guided by Positions of Non-overlapping Characters," *Trans. IEICE*, vol.J67-D, no.11, pp.1194–1201, 1984.

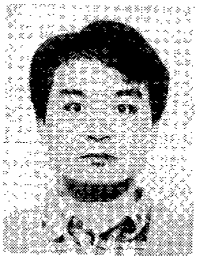
[7] Sun, N., Tabara, T., Aso, H. and Kimura, M., "Printed character recognition using directional element feature," *Trans. IEICE*, vol.J74-D-II, no.3, pp.330–339, 1991.

[8] Fletcher, L.A. and Kasturi, R., "A Robust Algorithm for Text Strings Separation from Mixed Text/Graphics Images," *IEEE Trans. Pattern Anal. Match. Intell.*, vol.10, no.6, pp.910–918, 1988.

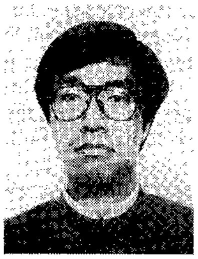
[9] Takizawa, K., Arita, D., Minoh, M. and Ikeda, K., "Extraction of Character Strings from Unformed Document Images," *Proc. of 2nd ICDAR*, pp.660–663, 1993.



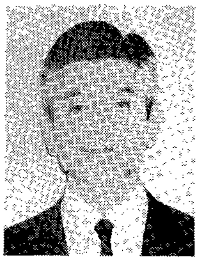
Kei Takizawa was born in Kanagawa, Japan in 1970. He received the B.E. degree in information science from Kyoto University in 1992. Now he is working toward the M.E. degree from Kyoto University. His research interests include document image analysis and post-processing of character recognition. He is a member of IPSJ.



Daisaku Arita was born in Chiba, Japan in 1968. He received the B.E. degree from Kyoto University in 1992. He is currently a graduate student at Department of Information Systems, Kyushu University. He has been engaged in document analysis and image understanding.



Michihiko Minoh was born in 1956 in Kyoto, Japan. He received the B.E., M.E. and D.E. degrees from Kyoto University, in 1978, 1980 and 1983, respectively. He has engaged in research in a variety of Image Processing and Artificial Intelligence. He is currently an Associate Professor of Integrated Media Environment Experimental Laboratory, Kyoto University. He is a member of IPSJ, IEEE and ACM.



Katsuo Ikeda was born in 1937 in Shiga, Japan. He received the B.E. and M.E. degrees in electronic engineering and the D.E. degree in information science from Kyoto University in 1960, 1962 and 1978, respectively. He is currently a Professor of the Department of Information Science, Kyoto University. His primary research interests include construction of intelligent information media environment. He is the author of *Structure of a Computer Utility* (in Japanese; Shokodo, 1974) and *Data Communication* (in Japanese; Shokodo, 1993), and the translator of *System Programming* (J.J. Donovan, 1974) and *Operating System* (J.J. Donovan, 1976). He is a member of IPSJ, IEEE, ACM and the editorial board of Information Processing Letters, Elsevier.