International Journal of Fuzzy Logic and Intelligent Systems, vol. 8, no. 3, September 2008 pp. 225-230

Recognition of the Printed English Sentence by Using Japanese Puzzle

Young-Sun Sohn

Dept. of information & communications engineering, Tongmyong University

Abstract

In this paper we embody a system that recognizes printed alphabet, numeral figures and symbols written on the keyboard for the recognition of English sentences. The image of the printed sentences is inputted and binarized, and the characters are separated by using histogram method that is the same as the existing character recognition method. During the abstraction of the individual characters, we classify one group that has not numerical information by the projection of the vertical center of the character. In case of another group that has the longer width than the height, we assort them by normalizing the width. The other group normalizes the height of the images. With the reverse application of the basic principle of the Japanese Puzzle to a normalized character image, the proposed system classifies and recognizes the printed numeral figures, symbols and characters, consequently we meet with good result.

Key words : Japanese Puzzle, image processing, character recognition, pattern recognition

1. Introduction

Men have used conventional computer input devices such as keyboard, mouse, tablet and touch screen conveniently. With the enhancement of computer capacity, high speed, enlargement of computer storage and precision, it is difficult for conventional input devices to process large amount of document because of increment of the error rate and the time required. To solve such problem, many automatic recognition systems have been developed. The representative algorithms include the circular pattern vector method [1], the structural method[2], the statistical method[3], the neural network method[4]. The fields of application are the recognition of a visiting card, real time English dictionary, the recognition of an automobile registration number plate, the recognition of postal matter, the recognition of a passport.

This paper proposes a character recognition algorithm that classifies the characteristics of the printed alphabet, numeral figures and symbols written on the keyboard through the use of the Japanese puzzle for developing the recognition of the printed alphabet[5] to the recognition of the printed English sentences.

2. Japanese Puzzle

Japanese Puzzle is a pictorial logic puzzle developed by Nishio Tetsuya in 1988, it is based on the squared grid that represents the figure with black and white. Figue 1 shows "Kingkong" an example of the Japanese puzzle.

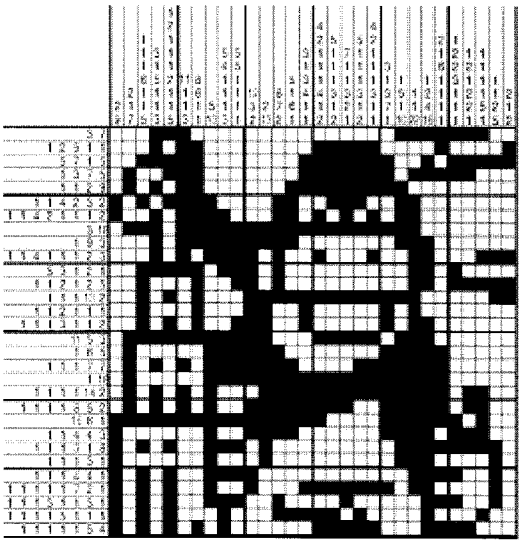


Figure 1. An example of the Japanese Puzzle

On each row and column is a series of numbers that represent the black squares of the respective line. Each number indicates the span of consecutive black squares. For example, the numerical information "3,2"signifies a set of 3 consecutive black squares and a set of 2 consecutive black squares with one or more white squares in between the two[5].

3. System algorithm

The image of the printed character is inputted then binarized, and the Histogram method is applied to separate the individual characters. The separated individual characters are normalized, and the proposed algorithm is applied to obtain the numerical information of the character image to be recognized. From the obtained numerical information, the characteristic patterns are

Manuscript received May. 15, 2008; revised Sep. 10, 2008.

extracted and compared with the standard patterns in order to recognize the character. Finally, the word editor outputs the recognized character.

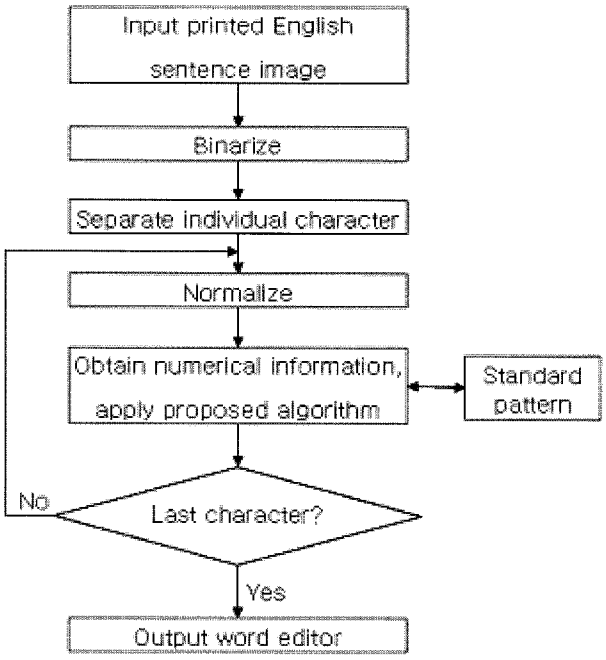


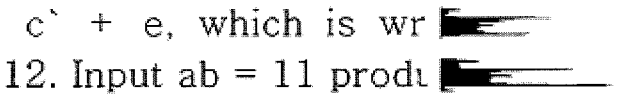
Figure 2. Whole system flow chart

3.1 Binarization of the Image and Character Region Selection

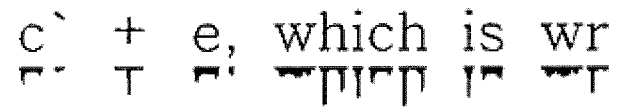
The text document is inputted by using the black-and-white CCD camera, and it is binarized. Histogram is projected horizontally for separating lines and vertically for separating individual characters. Figure 3 demonstrates an example of the Histogram method application on a binarized image.

The height of the separated individual characters (alphabet, numeral figure, symbol) image is the height of a line. To get the real height of each character image, only the character regions are obtained through applying the horizontal Histogram to each separated character. From the example of figure 4(a), the width of character images, 'c` + e, which', is separated in accordance with the region of individual character image, and the height of character images is applied to the height of a line region. Figure 4(b) shows the example of getting the height region of individual character image through applying the horizontal Histogram.

At the stage of separation of the individual character region as shown in figure 4(a), if the numerical information is not obtained by horizontally projecting the vertical center of individual character, some symbols are divided into 3 groups on the basis of the horizontally projected vertical center of individual character. One represents symbols which are located to the upper part(‘ ’ “ ” ` ^), another includes symbols which are located to the lower part(. ,), the other has symbols which are located to the both part(; :).



(a) Separation of lines



(b) Separation of individual characters

Figure 3. An example of the histogram

(a) The height of individual character image by separation of line

(b) The real height of individual character image through applying the horizontal Histogram to (a)

Figure 4. An example of the abstract of the character area

3.2 Size Normalization

Because the numerical information is acquired and compared on equal terms to compare inputted individual character image with standard pattern, all characters are normalized to a same size. Most characters(alphabet, numeral figures, symbols) are normalized by the vertical pixel number of the character at 48 pixels. But because the height of the symbols, i.e. equal(=), minus(-) and tilde(~), are shorter than the width of them, these symbols are normalized by the horizontal pixel number of the symbols at 48 pixels and this condition is used on the terms of classification.

For example when the minus(-) symbol with 22×3 (22 pixels wide, 3 pixels high) is normalized, the resulting size of the normalized image will be 48×6 pixels. If the capital W with 40×32(49 pixels wide, 32 pixels high) is normalized, the resulting size of the normalized image will be 60×48 pixels. Any pixels not selected during the mapping procedure are supplemented by the linear interpolation[7].

3.3 Character recognition method by numerical information of image and position using Japanese puzzle

The characters are recognized by using the numerical information obtained through the reverse application of the Japanese puzzle to the normalized character images. At each stage, the terms of classification is decided by the numerical information or the position information of the region which

represents a characteristic of the character image. For the recognition of the individual characters, the presence of the data of the characteristic regions determines the classification of the patterns. Figure 5 shows the 8 kinds of characteristic regions(R_1~R_5 for height, C_1~C_3 for width) which present useful information for the character recognition.

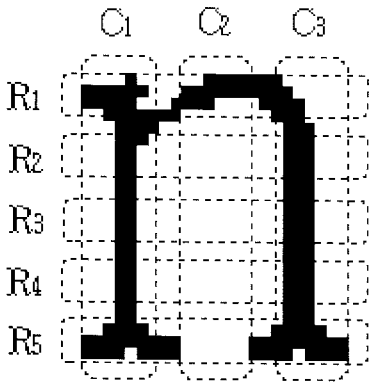


Figure 5. 8 characteristic area

The numerical information is obtained by projecting the pixel-based lines on the vertical and horizontal regions of the normalized printed character. From the obtained numerical information, the number of the consecutive black pixels that represent the image information of the character is expressed as *Pn*. The position information represents the position of the specific black pixel of a character as the coordinates (x, y). It is denoted as (*P1*) if the consecutive black pixels occur once, (*P1*, *P2*) for twice and so on depending on the number of such occurrence; the alphabet shows 4 patterns, (*P1*), (*P1*, *P2*), (*P1*, *P2*, *P3*), and (*P1*, *P2*, *P3*, *P4*), due to its structural properties.

For the recognition of the character, the individual characters are classified into 5 groups as shown in figure 6. One is horizontal 48 pixels group which is classified in the process of normalization, another includes 3 groups which are classified in early stage in the process of individual character region, the other is vertical 48 pixels group which is represented with general character image.

Firstly, the horizontal 48 pixel group which is classified in the process of normalization is divided minutely. The proposed system checks for the presence of the consecutive black pixels of 48 within all of the vertically projected region in order to separate the symbol character tilde(~) from the rest. The system checks for the presence of the consecutive black pixels of 3 or more within the vertically projected region C2 is (*P1*, *P2*) in order to separate the symbol character equal(=) from the rest. The other is the symbol character minus(-).

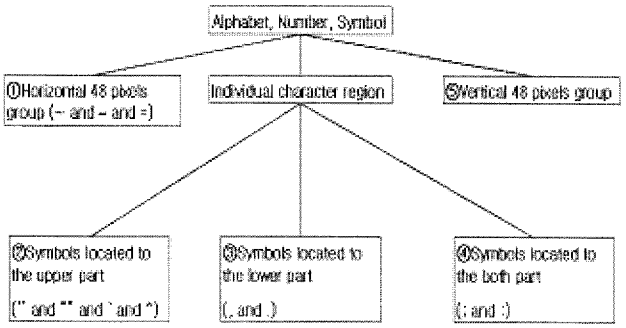


Figure 6. First 5 separations of the characters

Secondly, the three groups which are divided at the stage of separating the region of individual character is classified minutely. In case of symbols which are located to the both part(: and ;), The system checks for the presence of the consecutive black pixels of 5 or more and 5 or more and 3 or more within the vertically projected region C2 is (*P1*, *P2*, *P3*) in order to separate the symbol character semi-colon(;) from the rest. The other is the symbol character colon(:). In case of symbols which are located to the upper part(‘ ’ and “ ” and ` and ^), the system checks for the presence of the consecutive black pixels of 10 or more and 3 or more, or 3 or more and 10 or more within the vertically projected region C2 is (*P1*, *P2*) in order to separate the symbol character a single quotation marks(‘ ’) from the rest. The system checks for the presence of the consecutive black pixels of 15 or more within the horizontally projected region R2 is (*P1*) in order to separate the right side mark of the single quotation marks from the left side mark. Since the double quotation marks is not divided as one character region at the stage of separating the region of individual character, if the system recognizes the single quotation marks two times continuously within the vertically projected region C2, the system recognizes it as the double quotation marks. From the other symbols(` and ^), the system checks for the presence of the consecutive black pixels of 3 or more within the horizontally projected region R3 is (*P1*, *P2*) in order to separate the symbol character circumflex(^) from the rest. The other is the symbol character grave(`). In case of symbols which are located to the lower part(, and .), the system checks for the presence of the consecutive black pixels of 10 or more and 3 or more within the vertically projected region C2 is (*P1*, *P2*) in order to separate the symbol character comma(,) from the rest. The other is the symbol character period(.).

For the example of the character recognition of the vertical 48 pixel group, recognition process of an alphabet ‘Z’ is explained. Table 1 provide the numerical informations of an alphabet ‘Z’ which are obtained by projection of all the vertical and horizontal region of an alphabet ‘Z’.

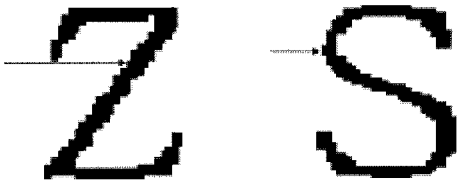
Table 1. The numerical information of the character 'Z'

Horizontal projection		Vertical projection	
[1]-(33)	[25]-(6)	[1]-(3)	[25]-(4, 8, 4)
[2]-(33)	[26]-(5)	[2]-(3)	[26]-(4, 7, 3)
[3]-(8, 7)	[27]-(5)	[3]-(4, 4)	[27]-(5, 7, 3)
[4]-(8, 7)	[28]-(5)	[4]-(4, 4)	[28]-(5, 8, 3)
[5]-(6, 5)	[29]-(5)	[5]-(6, 7)	[29]-(5, 7, 3)
[6]-(3, 5)	[30]-(5)	[6]-(8, 9)	[30]-(3, 7, 3)
[7]-(3, 5)	[31]-(7)	[7]-(8, 9)	[31]-(3, 6, 4)
[8]-(5, 5)	[32]-(5)	[8]-(5, 6, 4)	[32]-(10, 6)
[9]-(3, 6)	[33]-(5)	[9]-(5, 7, 4)	[33]-(10, 6)
[10]-(3, 6)	[34]-(4)	[10]-(5, 7, 4)	[34]-(7, 6)
[11]-(2, 4)	[35]-(4)	[11]-(4, 6, 4)	[35]-(7, 6)
[12]-(5)	[36]-(6)	[12]-(4, 8, 4)	[36]-(5, 4)
[13]-(5)	[37]-(5)	[13]-(4, 8, 4)	[37]-(4, 3)
[14]-(6)	[38]-(5)	[14]-(4, 8, 4)	[38]-(4, 1)
[15]-(4)	[39]-(4, 3)	[15]-(5, 6, 4)	
[16]-(4)	[40]-(5, 3)	[16]-(4, 6, 3)	
[17]-(5)	[41]-(5, 3)	[17]-(4, 7, 3)	
[18]-(5)	[42]-(6, 3)	[18]-(5, 7, 4)	
[19]-(4)	[43]-(4, 4)	[19]-(5, 6, 3)	
[20]-(6)	[44]-(4, 4)	[20]-(5, 8, 4)	
[21]-(6)	[45]-(5, 5)	[21]-(5, 8, 4)	
[22]-(5)	[46]-(7, 8)	[22]-(5, 7, 5)	
[23]-(4)	[47]-(7, 8)	[23]-(4, 6, 4)	
[24]-(4)	[48]-(33)	[24]-(4, 6, 4)	

Figure 7 shows the example of the position information to separate the alphabet 'Z' between 'Z' and 'S' which has similar numerical informations. In this example, the system projects horizontally from the top of each character in order to check for the presence of the consecutive black pixels of more or less of 7 is (*P1*) for the first time. The system compares the positive information (x, y) for each character and recognizes the alphabet 'Z'. From figure 7, the value of coordinate (x, y) pointed by an arrow is the position information.

Table 2, 3 provide the 74 separation conditions for the vertically normalized characters. Figure 8 shows the separation condition, X1, X3, X6, X13, X26, X42, X55, X65 to recognize an alphabet 'Z'.

The separation condition X1 from figure 8 checks for the presence of the consecutive black pixel value greater than 40 out of the total 48 within the vertically projected region C1, C2, C3 from figure 5 in order to separate the character with long vertical lines from the rest. The separation condition X3 checks for the presence of the consecutive black pixel of 3 or more and 20 or less within the horizontally projected region R3 is (*P1*) in order to separate the character with 1 set of horizontal consecutive black pixel and long horizontal lines from the rest. The separation condition X6 checks for the presence of the consecutive black pixel of 0 within the horizontally projected region R1, R5 is (*P1*). The separation condition X13 checks for the number of the presence of the consecutive black pixel of 3 or more within the vertically projected region C2 is (*P1*, *P2*, *P3*) and separates the character with the condition of 8 times detection.



Z : (24, 15) S : (1, 13)

Figure 7. An example of the position information for the character 'Z'and 'S'

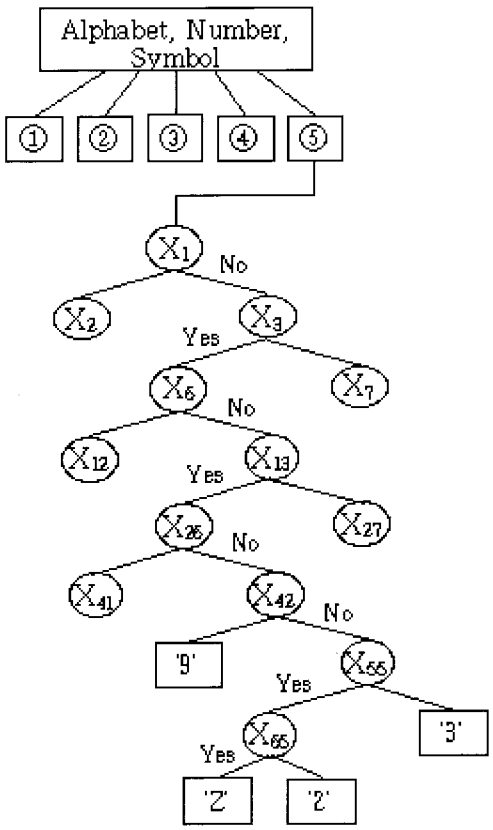


Figure 8. The separation diagram for the recognition of a character 'Z'

Among the character group 'S, Z, e, s, z, 2, 3, 5, 6, 8, 9' which apply the separation condition X26, character of font type Batang 'S, Z, 2' have similar numerical information, so the system separates characters by using the position information of the specific black pixel. The separation condition X26 checks for the presence of the consecutive black pixel of 3 or more and 12 or less within the horizontally projected region R3 is (*P1*) and checks for the presence of the x position value of (x, y) coordinates is left or not.

The separation condition X42 checks for the presence of the consecutive black pixel of 3 or more within the horizontally projected region R3 is (*P1*, *P2*). The separation condition X55

checks for the presence of the consecutive black pixel of 3 or more within the vertically projected region C1 is (*P1*, *P2*, *P3*). The separation condition X65 checks for the presence of the consecutive black pixel of 25 or more within the horizontally projected region R1 is (*P1*). From the result of separation condition X65, if the result is 'Yes', then the character 'Z' is recognized.

The system separates the same formed and different sized uppercase/lowercase alphabet just as 'Z, z' by applying the proportion of the height of the sentence to the height of the each character.

Table 2. The separation condition X1~X30 for the normalized characters on vertical area

Node	Regions	Conditions
X_1	C_1,C_2,C_3	If consecutive pixels of 40 or more and (P_1)
X_2	R_2	If consecutive pixels of 3 or more and (P_1)
X_3	R_3	If consecutive pixels of 3 or more and 20 or less and (P_1)
X_4	R_4	If consecutive pixels of 3 or more and (P_1)
X_5	C_2	If consecutive pixels of 3 or more and (P_1)
X_6	R_2, R_4	If consecutive pixels of 0 and (P_1)
X_7	*C_2, **R_3	*If consecutive pixels of 3 or more and (P_1), **If consecutive pixels of 40 or more and (P_1) is not detected
X_8	R_2	If consecutive pixels of 15 or more and (P_1)
X_9	C_1	If consecutive pixels of 40 or more and (P_1)
X_{10}	R_3	If consecutive pixels of 3 or more and (P_1)
X_{11}	R_3	If consecutive pixels of 15 or more and (P_1)
X_{12}	R_2	If consecutive pixels of 0 and (P_1)
X_{13}	C_2	If consecutive pixels of 3 or more and (P_1, P_2, P_3) :8 times detection
X_{14}	R_3	If consecutive pixels of 3 or more and (P_1, P_2)
X_{15}	R_3, R_4	If consecutive pixels of 30 or more and (P_1)
X_{16}	C_2	If consecutive pixels of 40 or more and (P_1)
X_{17}	R_1, R_5	If consecutive pixels of 25 or more and (P_1)
X_{18}	C_2	If consecutive pixels of 3 or more and (P_1)
X_{19}	C_2	If consecutive pixels of 3 or more and (P_1, P_2, P_3)
X_{20}	R_3	If consecutive pixels of 25 or more and (P_1)
X_{21}	R_3	If consecutive pixels of 3 or more and (P_1, P_2, P_3)
X_{22}	C_2	If consecutive pixels of 3 or more and(P_1, P_2)
X_{23}	C_1	If consecutive pixels of 40 or more and (P_1)
X_{24}	*C_1, **R_5	*If consecutive pixels of 2 or more and (P_1):3 times detection, **If consecutive pixels of 2 or more and (P_1, P_2)
X_{25}	R_2	If consecutive pixels of 3 or more and(P_1, P_2)
X_{26}	R_3	If consecutive pixels of 3 or more and 20 or less and (P_1) and checks for the presence of the x position value of (x, y) coordinates is left
X_{27}	C_1,C_2,C_3	If consecutive pixels of 24 or less
X_{28}	R_3	If consecutive pixels of 30 or more and (P_1)
X_{29}	R_3	If consecutive pixels of 3 or more and (P_1,P_2,P_3,P_4)
X_{30}	R_3	If consecutive pixels of 40 or more and (P_1)

Table 3. The separation condition X31~X74 for the normalized characters on vertical area

Node	Regions	Conditions
X_{31}	C_2	If consecutive pixels of 3 or more and(P_1, P_2, P_3)
X_{32}	C_2	If consecutive pixels of 40 or more and (P_1)
X_{33}	R_5	If consecutive pixels of 3 or more and(P_1, P_2)
X_{34}	C_3	If consecutive pixels of 25 or more and (P_1)
X_{35}	C_1	If consecutive pixels of 40 or more and (P_1)
X_{36}	C_3	If consecutive pixels of 35 or more and (P_1)
X_{37}	C_2	If consecutive pixels of 40 or more and (P_1)
X_{38}	R_4	If consecutive pixels of 3 or more and (P_1)
X_{39}	R_4	If consecutive pixels of 3 or more and (P_1)
X_{40}	C_2	If consecutive pixels of 3 or more and(P_1, P_2)
X_{41}	R_3	If consecutive pixels of 30 or more and (P_1)
X_{42}	R_3	If consecutive pixels of 3 or more and(P_1, P_2)
X_{43}	C_1, C_3	If consecutive pixels of 3 or more and (P_1) :3 times detection
X_{44}	C_2	If consecutive pixels of 8 or more and (P_1)
X_{45}	R_3	If consecutive pixels of 30 or more and (P_1)
X_{46}	C_1,C_2,C_3	If consecutive pixels of 35 or more
X_{47}	R_3	If consecutive pixels of 3 or more and(P_1,P_2,P_3,P_4)
X_{48}	R_5	If consecutive pixels of 8 or more and (P_1)
X_{49}	C_1	If consecutive pixels of 40 or more and (P_1)
X_{50}	C_3	If consecutive pixels of 2 or more and (P_1, P_2), :4 times detection
X_{51}	R_1	If consecutive pixels of 3 or more and 18 or more and (P_1, P_2)
X_{52}	C_2	If consecutive pixels of 40 or more and (P_1)
X_{53}	C_3	If consecutive pixels of 40 or more and (P_1)
X_{54}	R_4	If consecutive pixels of 3 or more and(P_1, P_2), :6 times detection
X_{55}	C_1	If consecutive pixels of 3 or more and (P_1,P_2,P_3)
X_{56}	C_2	If consecutive pixels of 10 or more and(P_1, P_2)
X_{57}	C_3	If consecutive pixels of 20 or more and (P_1)
X_{58}	C_1	If consecutive pixels of 3 or more and(P_1, P_2)
X_{59}	C_2	If consecutive pixels of 3 or more and(P_1,P_2,P_3,P_4)
X_{60}	R_3	If consecutive pixels of 3 or more and(P_1, P_2)
X_{61}	C_1	If consecutive pixels of 40 or more and (P_1)
X_{62}	*C_1, **C_3	*If consecutive pixels of 3 or more and(P_1, P_2), :5 times detection, or **If consecutive pixels of 3 or more and(P_1),:5 times detection
X_{63}	R_2	If consecutive pixels of 3 or more and (P_1)
X_{64}	R_2	If consecutive pixels of 3 or more and(P_1, P_2)
X_{65}	R_1	If consecutive pixels of 25 or more and (P_1)
X_{66}	C_1	If consecutive pixels of 3 or more and(P_1), :3 times detection
X_{67}	C_3	If consecutive pixels of 3 or more and (P_1)
X_{68}	R_3	If consecutive pixels of 3 or more and (P_1,P_2,P_3)
X_{69}	R_4	If consecutive pixels of 3 or more and 10 or less and (P_1, P_2)
X_{70}	C_2	If consecutive pixels of 40 or more and (P_1)
X_{71}	R_3	If consecutive pixels of 3 or more and 10 or more and (P_1, P_2)
X_{72}	R_1	If consecutive pixels of 25 or more and (P_1)
X_{73}	R_2	If consecutive pixels of 3 or more and(P_1, P_2, P_3)
X_{74}	C_1	If consecutive pixels of 2 or more and(P_1), :5 times detection

4. Experiment

In this paper, 2156 characters of font type Batang and 2156 characters of font type Dodume that include the entire uppercase and lowercase alphabet, 200 number of font type Batang and 200 number of font type Dodume, and 560 symbols of font type Batang and 560 symbols of font type Dodume were tested by implementing the proposed algorithm which maintains the height of the character image at 30 pixels or higher for a stable recognition.

Figure 9 shows an example of the input image, and figure 10 shows an example of the output of the characters, recognized by applying the proposed algorithm, printed on the word editor.

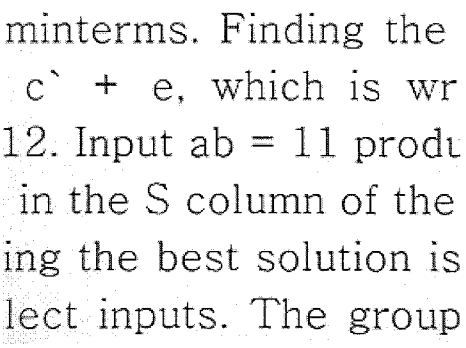


Figure 9. An example of the input image

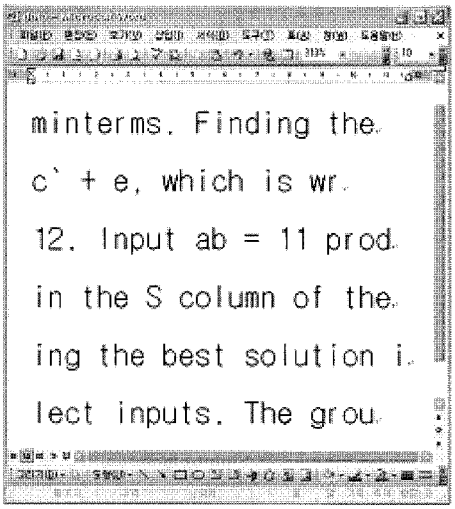


Figure 10. The recognition result of the input image

5. Results and Future Improvement

This paper embodied a character(an alphabet, a number and a symbol) recognition system by applying the proposed algorithm to the image of the English sentence inputted by a black-and-white CCD camera. The test result yielded a 100 % recognition rate under the optimal condition. The proposed algorithm does not include a linearization process that is generally used in the character recognition. From the result of high recognition rate earned in this paper, we can see that this algorithm is useful to the recognition of public signs that requires safety and accurate interpretation since the algorithm can be applied to the recognition of a figure image that does not use linearization process.

References

[1] S. B. Baek and Y. S. Sohn, Documentation of Printed hangul Images of the Selected Area by Finger Movement, *Journal of Fuzzy Logic and Intelligent Systems*, Vol. 12, No. 4, pp. 306-310, 2002.

[2] Z. Y. Lin, and P. Liu, Structural Attribute Feature Code Representation and Recognition of Multi - font Printed Chinese Characters, *International journal of pattern recognition and artificial intelligence*, Vol. 15, No. 2, pp. 218-310, 2001.

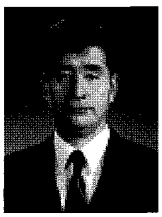
[3] K. C. Jung and H. J. Kim, Korean character recognition using a TDNN and an HMM, *Pattern recognition letters*, Vol. 20, No. 6, pp. 551-563, 1999.

[4] J. H. Kim, K. K. Kim and S. I. Chien, Korean and English Character Recognition System Using Hierarchical Classification Neural Network, *IEEE International Conference on System, Man and Cybernetics*, Vol. 1, pp. 759-764, 1995.

[5] Y. S. Sohn, K. S. Oh, B. S. Kim, A Recognition method of the Printed Alphabet by using Nonogram Puzzle, *Proceedings of The 8th International Symposium on Advanced Intelligent Systems*, pp. 232-236, 2007.

[6] Salcedo-Sanz. S, Ortiz-Garcia. E. G, Perez-Bellido. A. M, Portilla-Figueras. A and Xin Yao, Solving Japanese Puzzles with Heuristics, *Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Games (CIG 2007)*, pp. 224-231, 2007.

[7] D.J.Kang and J.E.Ha, *Digital Image Processing by Using Visual C++*, SciTech media, pp. 184-194, 2003.



Young-Sun Sohn
received B.S., M.S. degree in electronics from Dong-A University in 1981, 1983. From 1990 to 1998, he has been senior researcher in ETRI. He received Ph.D. degree from Tsukuba University in Japan. From 1998 to present, he is an Associate Professor, Dept. of Information & Communication Engineering college of Engineering, Tongmyong University in Korea. His research interests include Human Interface, Fuzzy Measures and Fuzzy Integrals, and Evaluation.