statistical N-gram grammar instead of a dictionary to allow an unlimited vocabulary. Experiments are to be carried out in the future to test the system's performance on large and unlimited vocabularies.

REFERENCES

[1] C.C. Tappert, C.Y. Suen, and T. Wakahara, "The State of the Art in On-Line Handwriting Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 8, pp. 787-808, Aug. 1990.

[2] R. Nag, K.H. Wong, and F. Fallside, "Script Recognition Using Hidden Markov Models," *Proc. ICASSP '86*, vol. 3, pp. 2,071-2,074, Japan, Apr. 1986.

[3] A. Kundu and P. Bahl, "Recognition of Handwritten Script: A Hidden Markov Model Based Approach," *Proc. ICASSP '88*, vol. 2, pp. 928-931, New York, Apr. 1988.

[4] S. Bercu and G. Lorette, "On-Line Handwritten Word Recognition: An Approach Based on Hidden Markov Models," *Proc. Third IWFHR*, pp. 385-390, Buffalo, N.Y., May 1993.

[5] J. Makhoul, T. Starner, R. Schartz, and G. Lou, "On-Line Cursive Handwriting Recognition Using Speech Recognition Models," *Proc. ICASSP '94*, pp. v125-v128, Adelaide, Australia, Apr. 1994.

[6] K.S. Nathan, H.S. M. Beigi, J. Subrahmonia, G.J. Clary, and H. Maruyama, "Real-Time On-Line Unconstrained Handwriting Recognition Using Statistical Methods," *Proc. ICASSP '95*, pp. 2,619,-2,622, Detroit, Mich., June 1995.

[7] M.Y. chen, A. Kundu, and J. zhou, "Off-Line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 481-496, May 1994.

[8] M.K. Brown and S.C. Glinski, "Stochastic Context-Free Language Modeling with Evolutional Grammars," *Proc. ICSLP '94*, vol. 2, pp. 779-782, Yokohama, Japan, Sept. 1994.

[9] G.L. Miller and R. Boie, "Capacitive Proximity Sensors," U.S. Patent #5,337,353,9, Aug. 1994.

[10] J. Hu, M.K. Brown, and W. Turin, "Invariant Features for HMM Based Handwriting Recognition," *Proc. ICIAP '95*, pp. 588-593, Sanremo, Italy, Sept. 1995.

[11] L.R. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[12] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, Feb. 1989.

[13] S.A. Guberman and V.V. Rozentsveig, "Algorithm for the Recognition of Handwritten Text," *Automation and Remote Control*, vol. 37, pp. 751-757, May 1976. (Translated from *Automatika i Telemekhanika*, vol. 37, no. 5, pp. 122-129, May 1976.)

[14] B.T. Lowerre and D.R. Reddy, "The HARPY Speech Understanding System," *Trends in Speech Recognition*, W.A. Lean, ed., chapter 15, pp. 340-360. Prentice Hall, 1980.

[15] M.K. Brown and J.G. Wilpon, "A Grammar Compiler for Connected Speech Recognition," *IEEE Trans. Signal Processing*, vol. 39, no. 1, pp. 17-28, Jan. 1991.

[16] A.M. Bruckstein, R.J. Holt, A.N. Netravali, and T.J. Richardson, "Invariant Signatures for Planar Shape Recognition Under Partial Occlusion," *CVGIP: Image Understanding*, vol. 58, pp. 49-65, July 1993.

[17] M. Schenkel, I. Guyon, and D. Henderson, "On-Line Cursive Script Recognition Using Time Delay Neural Networks and Hidden markov Models," *Special Issue of Machine Vision and Application on Cursive Script Recognition*, R. Plamondon, ed. Springer Verlag, 1995.

[18] F. Sinden and G. Wilfong, "Method of Recognizing Handwritten Symbols," U.S. Patent #5,333,209, July 26, 1994.

A New Methodology for Gray-Scale Character Segmentation and Recognition

Seong-Whan Lee, *Member, IEEE Computer Society*,
Dong-June Lee, *Member, IEEE*, and
Hee-Seon Park, *Member, IEEE*

Abstract—Generally speaking, through the binarization of gray-scale images, useful information for the segmentation of touched or overlapped characters may be lost in many cases. If we analyze gray-scale images, however, specific topographic features and the variation of intensities can be observed in the character boundaries. We believe that such kinds of clues obtained from gray-scale images may work for efficient character segmentation and recognition. In this paper, we propose a new methodology for character segmentation and recognition which makes the best use of the characteristics of gray-scale images. In the proposed methodology, the character segmentation regions are determined by using projection profiles and topographic features extracted from the gray-scale images. Then a nonlinear character segmentation path in each character segmentation region is found by using multi-stage graph search algorithm. Finally, in order to confirm the nonlinear character segmentation paths and recognition results, recognition-based segmentation method is adopted. Through the experiments with various kinds of printed documents, it is convinced that the proposed methodology is very effective for the segmentation and recognition of touched and overlapped characters.

Index Terms—Character segmentation and recognition, topographic feature, gray-scale character recognition, multistage graph search, recognition-based segmentation.

1 INTRODUCTION

IT is a challengeable issue to develop a practical system which can maintain a high recognition accuracy, independent of the quality of the input documents and the character fonts. Very often even in printed text, adjacent characters tend to be touched or overlapped. Therefore, it is essential to segment a given string correctly into its character components. Any failure or error in this segmentation step produces a negative effect on character recognition [1].

The complexity of character segmentation stems from the wide variety of fonts, rapidly expanding text styles, and image characteristics such as poor-quality printing and poor binary images. Touched, overlapped, separated, and broken characters are major factors for causing segmentation errors. Moreover, when a document is composed of multiple languages, (e.g., Hangul with alphanumeric characters), it is more difficult to segment characters due to differences in character sizes and touching types of each language.

Previous methods for character segmentation can be roughly classified into three categories: straight segmentation method, recognition-based segmentation method, and cut classification method.

- *S.-W. Lee is with the Dept. of Computer Science and Engineering, Korea University, Anam-dong, Seongbuk-ku, Seoul 136-701, Korea. E-mail: swlee@human.korea.ac.kr.*
- *D.-J. Lee is with the Telecommunication Network Research Lab., Korea Telecom, Woomyun-dong, Suhcho-ku, Seoul 137-792, Korea E-mail: djlee@tigermask.kotel.co.kr.*
- *H.-S. Park is with the Multimedia Lab, Samsung Electronics Co. Ltd., Suwon P.O. Box 105, Kyungki-do 440-600, Korea. E-mail: hspark@coda.info.samsung.co.kr.*

Manuscript received Mar. 13, 1995; revised Dec. 26, 1995. Recommended for acceptance by J.J. Hull.
For information on obtaining reprints of this article, please send e-mail to: transpami@computer.org, and reference IEEECS Log Number P96081.

In the first category, each word is segmented into several characters, and the character recognition techniques are applied to each segment [2]. In spite of the simplicity in implementing this method, its limit comes from the fact that it should depend on high accuracy of the segmentation points found. However, such accurate segmentation technique is not yet available yet. Consequently, word segmentation and character recognition are needed to be combined. In the second category, a number of potential segmentation points are found in the touched characters [3], [4], [5]. And the candidates are confirmed by using recognition results. This method is more reasonable than the first one, but it depends on the performance of the recognizer. The third category is cut classification method for segmentation [6]. This method is based on a classifier deciding whether it represents a cut hypothesis or not, for each column of the character image. In this method, the neural network can be adapted by a sample set containing images of merged patterns in the training phase, and the decision rules are created automatically rather than being man-made heuristics. But, it is difficult to train the neural network for every pair of touching characters when the number of characters to be recognized increases.

Most character segmentation methods have been developed for binary text images. However, through the binarization of gray-scale images, useful information for character segmentation and recognition may be lost as shown in Fig. 1. Furthermore, in order to segment overlapping characters in binary images, it is necessary to extract and merge the connected components. However, in cases that characters are touching and overlapping at the same time, the connected component analysis is ineffective strategy for character segmentation.

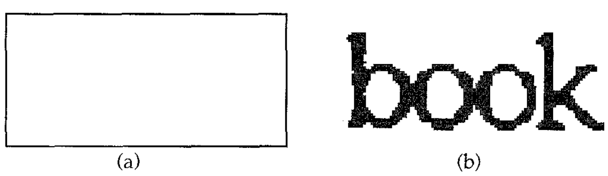


Fig. 1. Loss of information through binarization. a) Gray-scale image. b) Binary image.

If we analyze the gray-scale images, however, specific topographic features could be found in touching fields [7]. In addition to that, the variation of intensities can be observed in the character boundaries. We believe that such kinds of clues obtained from gray-scale images may work for efficient character segmentation and recognition.

In this paper, we propose a new methodology for character segmentation and recognition which makes the best use of topographic features and the variation of intensities in gray-scale images. The proposed methodology is composed of three steps; determination of character segmentation region, search for nonlinear character segmentation paths by multi-stage graph search algorithm, and confirmation of the nonlinear character segmentation paths and character recognition results.

Recently, an interesting approach for segmentation-free recognition on gray-scale images was proposed by Rocha and Pavlidis [8]. This approach is specially suited for the processing of touched and broken characters. The main difference between this method and the proposed method is that the former method searches for subgraphs homeomorphic from previously defined prototypes of characters, while the latter method searches for nonlinear character segmentation paths in multistage graph.

In order to verify the performance of the proposed methodology, various printed documents which are composed of Hangul and alphanumeric characters have been used for experiments. Also experiments have been carried out with word images having Gaussian noise and salt and pepper noise to verify the noise effects on the proposed methodology. Experimental results reveal that the proposed methodology is very effective for the segmentation and recognition of touched or overlapped characters.

2 Determination of Character Segmentation Regions

2.1 Projection Profiles in Gray-Scale Images

For the binary image, the projection profile can be obtained by counting the number of black pixels in a column or row. Since gray-scale images have an intensity for each pixel, however, the projection profiles cannot be obtained by simply counting the black pixels. We define the projection profiles in gray-scale images as follows:

Let $g(x, y)$ be the intensity of a pixel (x, y) in gray-scale images. Then $g(x, y)$ has the value of range as follows:

$$0 \le g(x, y) \le L - 1 \tag{1}$$

where L is the level of intensity. Let $H_x(g)$ and $H_y(g)$ be the histograms of column x and row y with intensity of g, respectively. The vertical projection profile, $P(x)$ can be defined as follows:

$$P(x) = \sum_{g=0}^{L-1} H_x(g) \cdot c(g), \qquad 0 \le c(g) \le 1 \tag{2}$$

where $c(g) = \sum_{y=0}^{h} \frac{g(x,y)}{L}$ is a ratio contributing to the projection with intensity of g and h is the height of the image.

In similar way, the horizontal projection profile, $P(y)$ can be defined by

$$P(y) = \sum_{g=0}^{L-1} H_y(g) \cdot c(g), \qquad 0 \le c(g) \le 1. \tag{3}$$

2.2 Topographic Features in Gray-Scale Images

Recent works on topographic feature extraction are aimed at preserving topological information of gray-scale images, and minimizing loss of information caused by the binarization. In this paper, Lee and Kim's method [10] has been used for topographic feature extraction. Fig. 2a shows a gray-scale image, and Fig. 2b shows the 3D topographic shape for the gray-scale image in Fig. 2a. Through the analysis of this topographic shape, we can extract the topographic features such as peak, ridge, hillside, saddle, ravine, flat, and pit. In this paper, the peak, ridge, and saddle points have been used for character segmentation. Examples of the extracted topographic features are shown in Fig. 2c. The peak, ridge, and saddle points were marked as P, R, and S, respectively.

2.3 Character Segmentation Region

2.3.1 Selection of Presegmentation Points

The input image is presegmented by using the projection profile $P(x)$ and the topographic features of gray-scale images. The columns satisfied by at least one of the following conditions are selected as the presegmentation points.

1) A column which has $P(x)$ less than a threshold.
2) A column which is composed of only saddle features.
3) A column which is composed of saddle and ridge features, and has $P(x)$ less than a threshold.
4) A column which is composed of ridge features, and has $P(x)$ less than a threshold.
5) A middle column of the right end for a connected component, and the left end for the next connected component of topographic features.

Here, the threshold has been determined to be proportional to the height of word image.

Fig. 3(a) shows an example of pre-segmentation points.

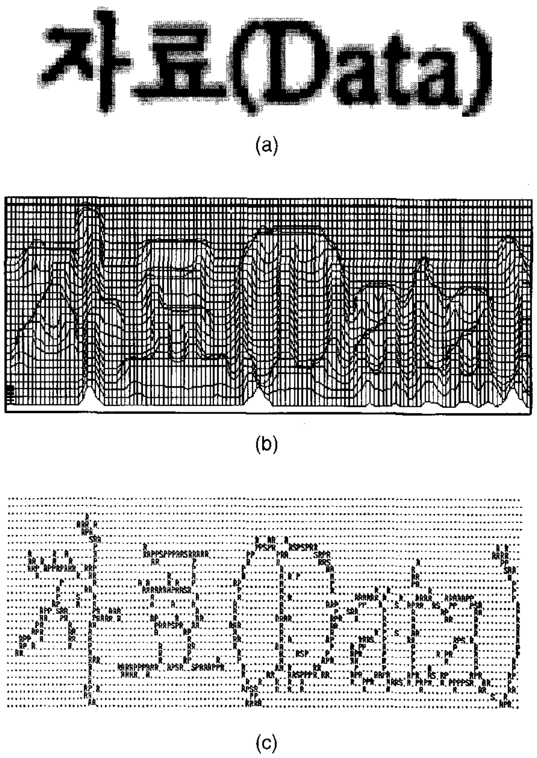


Fig. 2. Topographic shape and features of gray-scale image. a) Gray-scale image. b) 3D topographic shape. c) Topographic features extracted by using Lee and Kim's method.

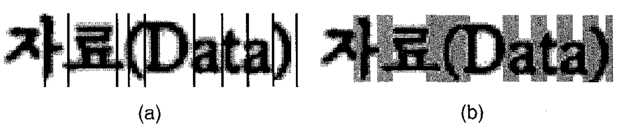


Fig. 3. An example of presegmentation and character segmentation regions. a) Presegmentation results. b) Character segmentation regions obtained from a).

2.3.2 Determination of Character Segmentation Region

A selected presegmentation point may be adjacent to optimal segmentation point. Moreover, when the characters are overlapped, the character boundaries must be defined nonlinearly. For each presegmentation point, the character segmentation region which is composed of the left and right neighbor columns of the presegmentation point is determined. And then the correct character boundaries are searched by using the nonlinear character segmentation path search algorithm which will be presented in the following section. Fig. 3b shows an example of character segmentation regions determined by pre-segmentation points in Fig. 3a. The gray rectangle corresponds to segmentation region, and the gray rectangle near "(" represents three character segmentation regions overlapped.

3 Search for Nonlinear Character Boundaries

3.1 Character Segmentation Problem in Gray-Scale Images

Because we assume that the intensity of a pixel in a noncharacter region is less than that in a character region, the accumulated intensity of the path along a character boundary is less than that through the character stroke. Therefore, the character segmentation problem can be defined as a problem of finding the shortest path which minimizes the accumulated intensity in the character segmentation region. For searching the shortest path, multistage graph search algorithm [11] is often used when a process can be partitioned into disjointed subprocesses. The problem of finding the minimum of accumulated intensity in the character segmentation region can be transformed into the problem of searching for a shortest path in the multistage graph.

Each row of pixels in character segmentation region corresponds to a stage in the multistage graph, each pixel in the character segmentation region a vertex, and the intensity of each pixel the distance between a vertex in a current stage and a vertex in the next stage.

3.2 Search for Nonlinear Character Segmentation Paths

The character segmentation region can be represented by a multistage graph as shown in Fig. 4. In Fig. 4, ∘ represents a pixel of a gray-scale image, and the line between pixels represents the linkage from the previous stage to the pixel. Each pixel has an intensity which is regarded as a distance in multistage graph.

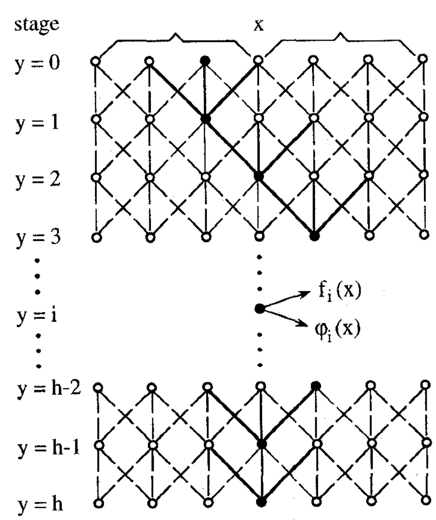


Fig. 4. Multistage graph representation of character segmentation region.

Let $f_y(x)$ be the minimum of accumulated distance in stage y, and $\psi_y(x)$ be a shortest path from vertices of $y-1$ stage to a vertex of y stage. We constrain the possible vertices linked into (x, y) to $(x-1, y-1)$, $(x, y-1)$, and $(x+1, y-1)$. We define the path with the minimum accumulated intensity as the nonlinear character segmentation path. The path can be searched by the following algorithm.

- **Nonlinear character segmentation path search algorithm**

  w: the width of the character segmentation region

  - Initialization: for $0 \le x \le w$

    $f_0(x) = g(x, 0)$,

    $\psi_0(x) = x$.

  - Recursion: for $1 \le y \le h$

    $f_y(x) = \min_{0 \le x \le w} \{g(x,y) + f_{y-1}(x)\}$,

    $\psi_y(x) = arg \min_{0 \le x \le w} \{g(x,y) + f_{y-1}(x)\}$.

- Termination:

$$f^* = \min_{0 \le x \le w} \{f_h(x)\},$$

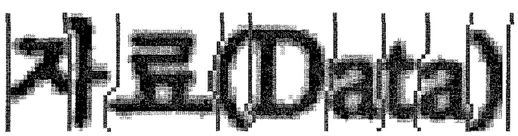
$$m_h^* = arg \min_{0 \le x \le w} \{f_h(x)\}$$

- Backtracking: for $y = h - 1, h - 2, \ldots, 1$

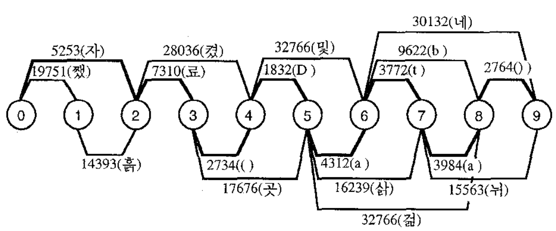
$$m_y^* = \psi_{y+1}(m_{y+1}^*).$$

In the first step, $f_0(x)$ and $\psi_0(x)$ are initialized with the intensity of pixel $(x, 0)$ and the columns in the first row, respectively. Then, the accumulated distance $f_y(x)$ can be recursively evaluated at each stage. In the final stage $y = h$, we have $w + 1$ accumulated distances, $f_h(x)$, $x = 0, 1, \cdots, w$. The minimum accumulated distance f^* of these distances is the candidate for the shortest path. The final task, now, is to backtrack from m_h^* to the initial vertex following ψ_y. It is not difficult to see that the complexity of this algorithm is $\theta(e + h)$ [11], where e is the number of vertices, and h is the number of stages in a graph.

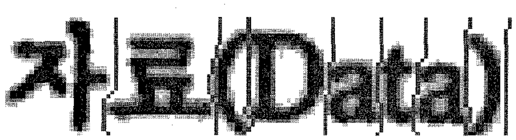
Fig. 5a shows candidate nonlinear character segmentation paths.



(a)



(b)



(c)

Fig. 5. Optimal character segmentation. a) Candidate nonlinear character segmentation paths. b) Segmentation graph. c) Optimal nonlinear character segmentation paths.

4 Character Segmentation Based on Recognition Results

Candidate nonlinear character segmentation paths which are found by the proposed nonlinear character segmentation path search algorithm are not always correct character boundaries. In order to confirm the nonlinear character segmentation paths and recognition results, the recognition-based segmentation scheme has been adopted. The candidate nonlinear character segmentation paths and the recognition results can be represented as nodes and distances in a graph, respectively. In order to distinguish this graph from multi-stage graph, it is called as segmentation graph. The optimal nonlinear character segmentation paths can be confirmed by searching a shortest path in the segmentation graph.

4.1 The Graph Representation of Nonlinear Character Segmentation Paths and Character Recognition Results

The candidate nonlinear character segmentation paths can be represented as nodes in the segmentation graph. An example of a segmentation graph is shown in Fig. 5b. In the segmentation graph, the nodes, 0, 1, ..., 9 represent candidate nonlinear character segmentation paths. Candidate characters are generated by combining the candidate nonlinear character segmentation paths which have the widths less than a threshold. The distances between each candidate character and all reference models are calculated, and then the reference model with the minimum distance is selected. The minimum distance is represented as the distance between nodes in the segmentation graph as shown in Fig. 5b. Then, the optimal nonlinear character segmentation paths and recognition results can be confirmed by searching the shortest path in the segmentation graph.

4.2 Optimal Character Segmentation and Recognition

The set of possible paths in the segmentation graph can be represented as follows:

$$P = \{p_k \mid k = 1, \ldots, M\}, \tag{4}$$

where p_k is a path and M is the number of possible paths in the segmentation graph. Let d_{ij} be the distance from node i to node j, and $E(p_k)$ be a set of edges on the path p_k. Then the total distance of a path p_k can be determined as follows:

$$D(p_k) = \sum_{<i,j> \in E(p_k)} d_{ij} \tag{5}$$

where $<i, j>$ is an edge in $E(p_k)$.

The optimal nonlinear character segmentation paths and recognition results can be confirmed by searching the shortest path, p_s which minimizes $D(p_k)$ in the segmentation graph.

$$p_s = arg \min_{p_k} D(p_k) \tag{6}$$

The shortest path is drawn with thick gray lines in Fig. 5b. Each node on the shortest path represents a character boundary between characters, and the recognition results on the edge of the shortest path become the final recognition results. Fig. 5c shows an example of optimal nonlinear character segmentation paths.

5 Experimental Results and Analysis

Experiments for character segmentation and recognition were carried out with images which are scanned from the photocopy of five kinds of the real-life documents, such as technical journals, magazines, and some printed materials. All images were 300 dpi, gray-scale. Those documents contained Hangul and alphanumeric characters, and had many touched and overlapped characters. A hierarchical neural network classifier [12] has been used for character recognition. In order to classify 2,432 classes including 2,350 Hangul characters, 10 numerals, 52 English alphabets, and 20 special characters, the neural network has been trained with 729,600 sample characters which are different from the testing set. They had three most frequently used font types (Myungjo, Gothic, and Goongseo for Hangul characters and Times, Arial, and Courier for alphanumeric characters), and five different font sizes (8, 10, 12, 14, and 16 points).

The correct character segmentation accuracies for five kinds of documents are shown in Table 1.

In order to compare the performance of the proposed method to that of character segmentation methods in binary images, a projection profile analysis method (Method 1) and authors' previous methods (Method 2) [13] have been applied. The documents in which the letterspaces are varied with 0%, –4%, and –7% of current character width using Hangul wordprocessor [14] have been generated. In Method 1, the cutting points have been obtained by finding the positions where the vertical projection profile is less than a threshold. If the width of candidate character is larger than the threshold, the candidate character is separated in the position where the vertical projection profile is minimized. Method 2 is a kind of recognition-based segmentation method. In this method, pre-segmentation points are found, and candidate characters are recognized in binary images. And then the pre-segmentation points and recognition results are represented as nodes and distances in a graph. Finally, segmentation points are confirmed by searching a shortest path in the graph. Fig. 6 shows examples of the segmentation results of three methods. Table 2 shows the correct character segmentation and recognition accuracies for documents with various letterspaces. As shown in Table 2, the proposed methodology is more effective for touched or overlapped characters than the previous two methods.

TABLE 1
CHARACTER SEGMENTATION ACCURACIES FOR THE TEST SET WITH FIVE KINDS OF DOCUMENTS

	Kinds of documents				
	#1	#2	#3	#4	#5
Segmentation accuracy	98.8%	98.4%	96.7%	97.8%	98.4%

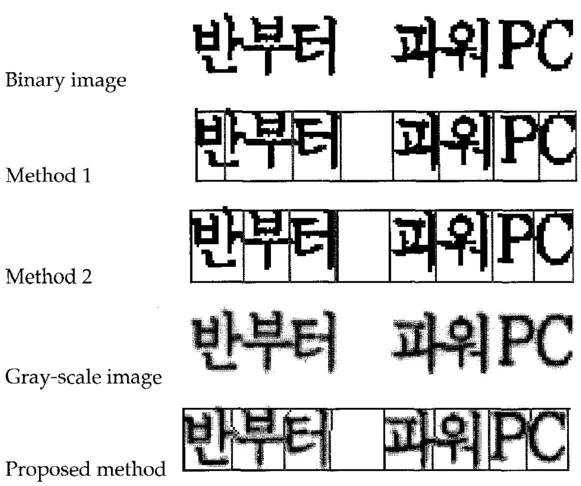


Fig. 6. Examples of character segmentation results.

TABLE 2
CHARACTER SEGMENTATION AND RECOGNITION ACCURACIES WITH VARIOUS LETTERSPACES

	Method	Letterspace		
		0%	–4%	–7%
Segmentation	Method 1	91.0%	87.8%	85.0%
	Method 2	98.0%	96.0%	92.3%
	Proposed method	98.6%	97.6%	96.2%
Recognition	Method 1	85.5%	82.7%	79.0%
	Method 2	92.1%	90.2%	86.8%
	Proposed method	92.7%	91.7%	90.4%

In order to verify the noise effects on the proposed methodology, 50 word images composed of Hangul and alphanumeric characters were generated, and the generated images have been corrupted by additive Gaussian noise with $N(0, \sigma^2)$.

The signal to noise ratio (SNR) has been defined as

$$SNR = \frac{|q_2 - q_1|}{\sigma}, \tag{7}$$

where q_1 and q_2 are the two gray-levels, and σ is the standard deviation of the noise. In this experiments, σ was varied with 10, 20, and 30. In Fig. 7, an original gray-scale image, images corrupted by additive Gaussian noise, and character segmentation results for three SNR's (SNR = 25.6, 12.8, and 8.2) have been presented.

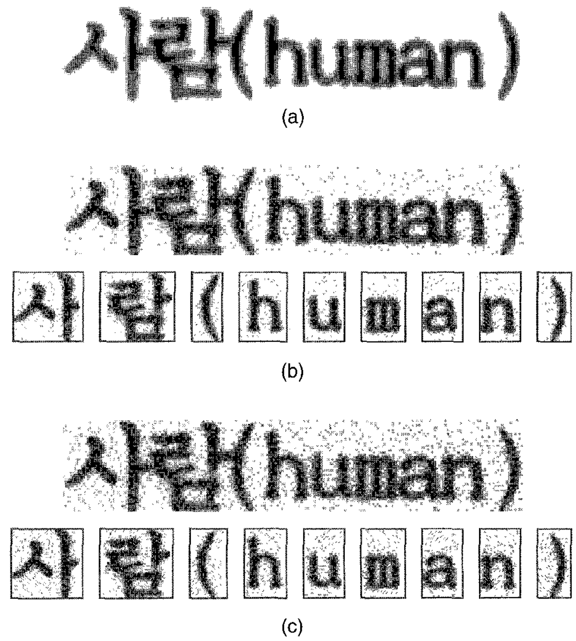


Fig. 7. Example of character segmentation for Gaussian noise-added image. a) Original gray-scale image. b) Noise-added image with SNR = 12.8 (σ = 20) and segmentation result. c) Noise-added image with SNR = 8.2 (σ = 30) and segmentation result.

Salt and pepper noise is added to the original image with a probability of 5%, 10%, and 15% for each noise with 256 gray-level values. Fig. 8 shows the salt and pepper noise-added images and character segmentation results. The segmentation error rates for the noise-added word images are shown in Table 3.

The segmentation errors in this experiment are classified into two categories: error in nonlinear character segmentation path search step (Type 1) and error in recognition-based character segmentation step (Type 2). The former occurs when the accumulated gray value of correct nonlinear character segmentation path is greater than searched segmentation path. The latter occurs when a wrong candidate character has shorter distance than correct one. The reason why the wrong candidate character has shorter distance than the correct one may be attributed to the fact the reference models were not fully trained with noise-added character images.

As shown in Table 3, the total correct segmentation accuracies decrease as the noise increases. However, the recognition error rates can be reduced by adopting noise invariant character recognizer.

6 CONCLUDING REMARKS

A binarization process can affect the segmentation and recognition results in a negative way. We have found that many of the segmentation errors are actually induced by binarization process in many cases. The broken and touched characters in binary images are well-formed characters in original gray-scale images. Moreover, we can extract specific topographic features and observe the variation of intensities in the character boundaries.

Fig. 8. Example of character segmentation for salt and pepper noise-added image. a) Original gray-scale image. b) Noise-added image with probability of 10% and segmentation result. c) Noise-added image with probability of 15% and segmentation result.

TABLE 3
CHARACTER SEGMENTATION ERROR RATES

Segmentation error	Gaussian noise (*SNR*)			Salt and pepper noise (*Noise ratio*)		
	25.6	12.8	8.2	5%	10%	15%
Type 1	0.99%	1.98%	3.47%	1.98%	4.46%	1.98%
Type 2	3.46%	3.47%	9.03%	2.97%	3.34%	7.92%
Total	4.45%	5.45%	12.5%	4.95%	7.8%	9.9%

In this paper, we proposed a new methodology for character segmentation and recognition which makes the best use of characteristics of gray-scale images. In the proposed methodology, the candidate segmentation points could be found efficiently by using topographic features and projection profiles of gray-scale images. And, the character boundaries could be found by observing the variation of intensities in gray-scale images even though the character boundarieswere defined nonlinearly. Finally, optimal nonlinear character segmentation paths and character recognition results could be found by adopting the recognition-based segmentation scheme. Through the experiments with various kinds of printed documents, it was convinced that the proposed methodology is very effective for the segmentation and recognition of touched and overlapped characters.

Segmentation and recognition may be improved by using contextual information. The proposed method used only the character recognition results in the recognition-based character segmentation step. However, the contextual knowledge could be used to reject the mis-recognized characters and find another path which has correct segmentation and recognition results.

ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their helpful comments in improving the earlier draft of this paper. This work was supported by the 1995 Directed Basic Research Fund of the Korea Science and Engineering Foundation.

REFERENCES

[1] Y. Lu, "Machine Printed Character Segmentation—An Overview," *Pattern Recognition*, vol. 28, no. 1, pp. 67-80, Jan. 1995.
[2] S. Kahan, T. Pavlidis, and H.S. Baird, "On the Recognition of Printed Characters of Any Fonts and Sizes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 2, pp. 274-288, Mar. 1987.
[3] A. Ariyoshi, "A Character Segmentation Method for Japanese Documents Coping with Touching Character Problems," *Proc. 11th Int'l Conf. Pattern Recognition*, The Hague, Netherlands, pp. 313-316, Aug. 1992.
[4] S. Liang, M. Shridhar, and M. Ahmadi, "Segmentation of Touching Characters in Printed Document Recognition," *Pattern Recognition*, vol. 27, no. 6, pp. 825-840, June 1994.
[5] S. Tsujimoto and H. Asada, "Major Components of a Complete Text Reading System," *Proceedings IEEE*, vol. 80, no. 7, pp. 1,133-1,149, July 1992.
[6] T. Bayer and U. Kresel, "Cut Classification for Segmentation," *Proc. Second Int'l Conf. Document Analysis and Recognition*, Tsukuba, Japan, pp. 565-568, Oct. 1993.
[7] L. Wang and T. Pavlidis, "Direct Gray Scale Extraction of Features for Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1,053-1,067, Oct. 1993.
[8] J. Rocha and T. Pavlidis, "A Solution to the Problem of Touching and Broken Characters," *Proc. Second Int'l Conf. Document Analysis and Recognition*, Tsukuba, Japan, pp. 602-605, Oct. 1993.
[9] J. Wang and J. Jean, "Segmentation of Merged Characters by Neural Networks and Shortest Path," *Pattern Recognition*, vol. 27, no. 25, pp. 649-658, May 1994.
[10] S.-W. Lee and Y.-J. Kim, "Direct Extraction of Topographic Features for Gray Scale Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 724-729, July 1995.
[11] E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*. Rockville, Ill.: Computer Science Press, 1989.
[12] S.-W. Lee and J.-S. Kim, "Multilingual, Multifont, and Multisize Large-set Character Recognition Using Self-Organizing Neural Network," *Proc.Third Int'l Conf. Document Analysis and Recognition*, Montreal, pp. 28-33, Aug. 1995.
[13] D.-J. Lee and S.-W. Lee, "Character Segmentation and Recognition in Korean Document Mixed with Alphanumeric Characters," *Proc. Fall National Conf. Korea Information Science Soc.*, Seoul, Korea, pp. 403-406, Oct. 1994, (in Korean).
[14] Hangul and Computer Co. Ltd., *Hangul Wordprocessor: Reference Manual*, Version 3.0b, Seoul, Korea, 1995.