Robust Stroke Extraction Method for Handwritten Korean Characters

Young-Kyoo Choi and Sang-Burm Rhee
Dept. of Electronics and Computer Engineering, Dankook University
San 8, Hannam-dong, Yongsan-gu, Seoul, Korea
Tel: +82-2-709-2860, Fax: +82-2-797-5857
young@dankook.ac.kr and sbrhee@dankook.ac.kr

Abstract: The merit of the stroke extraction algorithm is the ease of the feature abstraction from the skeleton of a character. But, extracting strokes from Korean characters has two major problems that must be dealt with. One is extracting primitive strokes and the other is merging or splitting the strokes using dynamic information of the strokes. In this paper, a method is proposed to extract strokes from an off-line handwritten Korean character. We have developed some stroke segmentation rules based on splitting, merging and directional analysis. Using these techniques, we can extract and trace the strokes in an off-line handwritten Korean character accurately and efficiently.

1. Introduction

Generally, a character recognition system carries out three processes: preprocessing, feature extraction and matching. The main task of preprocessing is to represent the input pattern with a meaningful description that is convenient for further processing.[1]

There are 51 graphemes in Korean characters, and a character is constructed by combining two or three graphemes in two dimensions. Therefore, Korean characters are much harder to recognize than English characters and numerals. Moreover, Korean characters have more cursive strokes than Chinese characters, which are composed of relatively simple straight strokes. Thus, a large stroke set is needed to represent Korean characters.[2]

In an on-line recognition system, the stroke structure can be obtained according to the sequences of writing via a pen-based input device such as a tablet. But in an off-line recognition system, the input characters are scanned optically and saved as raster images, so the stroke structure information is not available. The stroke extraction becomes more complicated and very important issue in off-line handwritten Korean character recognition. It is known that the advantage of on-line over off-line systems mostly comes from the dynamic information of strokes. We consider the stroke sequence of a character as most useful dynamic information. In this paper, a new method is proposed to extract strokes from an off-line handwritten Korean character. But, there are two major problems that must be deal with. One is to find the certain adjacent segmental strokes that should be merged into a complete stroke and the other is to divide the bend segment stroke into two or more individual strokes. We have developed some stroke segmentation rules based on splitting, merging and directional analysis. Using these techniques, stroke segmentation and stroke curve tracing can be carried out efficiently. A block diagram of the proposed stroke extraction method of handwritten Korean character is presented in Fig. 1. In Section 2, handwritten Korean characters are illustrated and a process of preprocessing is shown. Primitive strokes and the stroke extraction method are shown in Section 3. The experimental results are described in section 4. And, section 5 shows conclusions and further research directions.

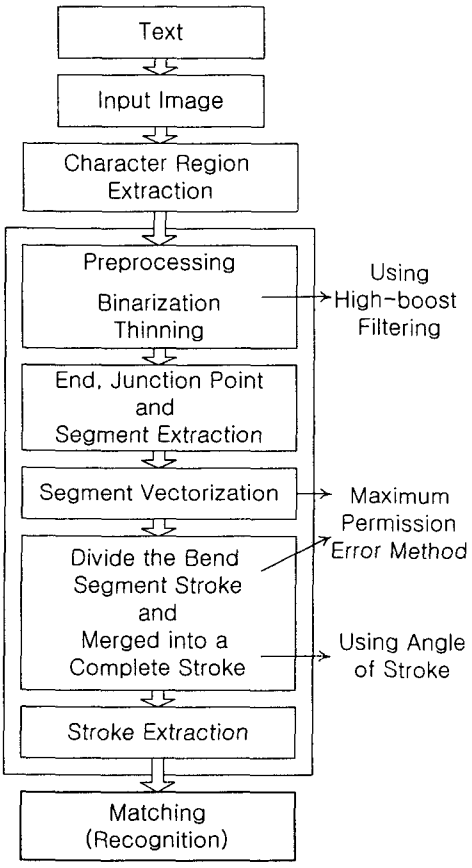


Fig. 1. A block diagram of the proposed method

2. Handwritten Korean Characters and Preprocessing

2.1 Handwritten Korean Characters

There are 24 simple graphemes in Korean characters. The graphemes can be further classified as consonants and vowels. Ten of the simple graphemes are vowels and the rest are consonants. Complex vowels and consonants are made by combining simple vowels and simple consonants, respectively. Table 1 shows graphemes of Korean characters. For every character, there must be one first consonant and at least one vowel, if it exists. The optional last consonant is placed below the first consonant and the vowel.[2]

Table 1. The graphemes of Korean characters

Vowel	Simple	ㅏ ㅑ ㅓ ㅕ ㅣ ㅗ ㅛ ㅜ ㅠ ㅡ
	Complex	ㅐ ㅒ ㅔ ㅖ ㅘ ㅙ ㅚ ㅝ ㅞ ㅟ ㅢ
Consonant	Simple	ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ
	Complex	ㄲ ㄸ ㅃ ㅆ ㅉ ㄳ ㄵ ㄶ ㄺ ㄻ ㄼ ㄽ ㄾ ㄿ ㅀ ㅄ

2.2 Preprocessing

Each character parts of the input image are extracted before the binarization of the parts are carried out. In the case of most of the handwritten character image, there comes some errors when extracting strokes from a character since the end part of a stroke tends to stick to the other strokes. Therefore, this kind of errors shown in Fig. 2 can be reduced by carrying out a binarization which is proper to handwritten character image. In this paper, by the use of the binarization method, a noise in a character or some errors by the background can be prevented using high-boost filter. Also, some sectional noise of a character image can be prevented using lowpass filter in high-boost filter. And, a skeleton to extract strokes can be extracted by thinning the result binary image using thinning algorithm. Fig. 4 shows a result of preprocessing.

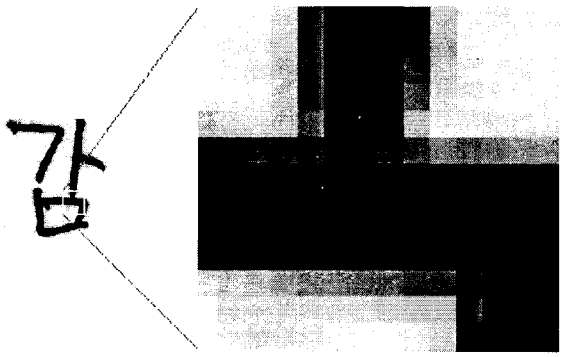


Fig. 2. The character with loose connection.

2.2.1 High-boost filter

A highpass filterd image may be computed as the difference between the original image and a lowpass filtered version of the image; that is

$$\text{Highpass} = \text{Original} - \text{Lowpass}$$

Multiplying the original image by an amplification factor, denoted by A, yield the definition of a high-boost or high-frequency-emphasis filter:

$$\begin{aligned}\text{High boost} &= (A)(\text{Original}) - \text{Lowpass} \\ &= (A-1)(\text{Original}) + \text{Original} - \text{Lowpass} \\ &= (A-1)(\text{Original}) + \text{Highpass}\end{aligned} \quad (1)$$

An A=1 value yields the standard highpass result. When A>1, part of the original is added back to the highpass result, which restores partially the low frequency component lost in the highpass filtering operation. The result is that the high-boost image looks more like the original image, with a relative degree of edge enhancement that depends on the value of A. The general process of subtracting a blurred image from an original, as given in the first line of Equation.(1), is called unsharp masking. This method is one of the basic tools for image processing applications in the printing and publishing industry. In terms of implementation, the preceding results can be combined by letting the center weight of the mask shown in Fig. 3 be

$$w = 9A-1 \quad (2)$$

with $A \geq 1$. The value of A determines the nature of the filter.[3] It is shown that 1.15-1.20 is experimentally suitable for the value of A in the basis of Korea national language information of KAIST(Korea Advanced Institute of Science and Technology).

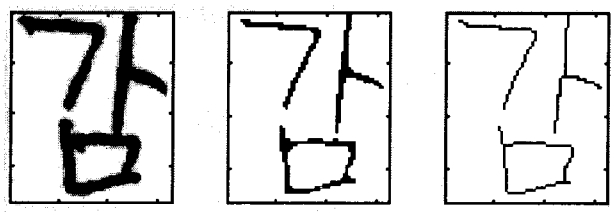
$$1/9 * \begin{bmatrix} -1 & -1 & -1 \\ -1 & w & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Fig. 3. The mask used for high-boost spatial filtering.

2.2.2 Thinning

Thinning basically consists of reducing a pattern to a thin-line representation. Thinning algorithms can be divided into two categories: contour-stripping and non-iterative. A non-iterative algorithm attempts to find the skeleton by finding which pixels are in the "middle" of a stroke, by tracing both sides of the contour. This presents difficulties that are known as parallel and sequential thinning algorithms. In sequential thinning algorithms, pixels are examined in a predetermined order, and changing the order implies changing the resulting skeleton. In parallel thinning algorithms, the pixel can be examined simultaneously, since only the results of the previous iteration influence the outcome of the current iteration, which proves advantageous for parallel implementations.[5]

Our experiments started with a version of the parallel thinning algorithm described at Zhang and Swen[6] which is later modified as proposed by Lu and Wang[7]. At a certain point a run length encoded version of the algorithm by Zang and Swen[6] was also used to process the digits. Instead of processing the digit map representation of the patterns, the algorithm directly treats the run length encoded image, which speeds the process as it easily skips white runs in the patterns. Later, more data was added and an improved sequential algorithm became available. On each iteration, it traces the contour, removing points marked for deletion. Under certain conditions, some points are marked to avoid deletion, primarily at sharp corners.



(a) Original image (b) Binary image (c) Skeleton

Fig. 4. The result of preprocessing.

3. Stroke Extraction

3.1 Feature point

A definition of the three types of points in a skeleton is presented in Table 2.

Table 2. The point types on skeletons

End point	Only one black neighbor
Skeleton point	Point with exactly two black neighbors
Junction point	Point with more than 2 black neighbors

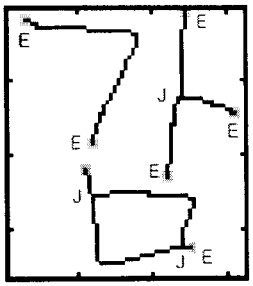


Fig. 5. The Feature point

From the thinned character image, feature points are found. To extract strokes, three types of feature points are used according to the numbers of neighbor, i.e. end point, skeleton point and junction point. An end point is a character point, which has only one neighbor pixel. Skeleton point is a character point that has two neighbor pixels. Junction point is a character point that has three or more neighbor pixels. Fig. 5 shows a feature point of handwritten Korean character. Marked "E" is end point, "J" is junction point and others are skeleton points.

3.2 Splitting and merging of primitive stroke

The two main tasks of the Korean character stroke extraction algorithms with thinning process are to find corner points and to handle junction points. A corner point divides a stroke into two strokes. The difficulty of finding the corner point is that the smooth transition corner and the abrupt transition corner exist in a Korean character. And the difficulty of handling the junction point is the violent distortion on the junction point produced by the thinning process, such as the split of one branch point into two or more branch points. The most natural way to extract primitive strokes is to transcribe them one by one. A primitive stroke can be transcribed from its one end point to the other end point or junction point. Also primitive strokes can be transcribed from its one junction point to the other junction point or an end point. If there is not any such point, then the transcription of this primitive stroke is finished. The transcription is repeated until all primitive strokes are extracted. The maximum permission error method[8] is used to vectorize the primitive strokes and to split the corner points before finding the angle of primitive vectors. To find the certain adjacent primitive strokes that shoud be merged into a complete stroke and calculate the angle of the coming and the forward primitive vector at the junction point, the forward primitive vector whose angle is near to the angle of the coming primitive vector is selected to be transcribed forward.

4. Experimental Results

In this paper, we use relatively simple character in the basis of Korea national language information of KAIST(Korea Advanced Institute of Science and Technology). Fig. 6 and Fig. 7 are the process and result of the stroke extraction for handwritten Korean characters.

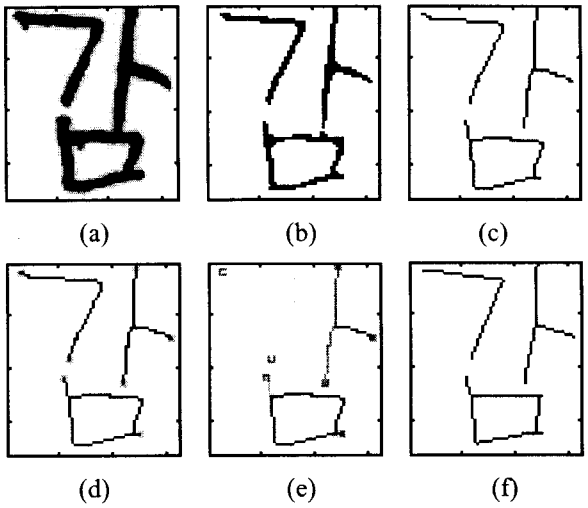


Fig. 6. The process of stroke extraction

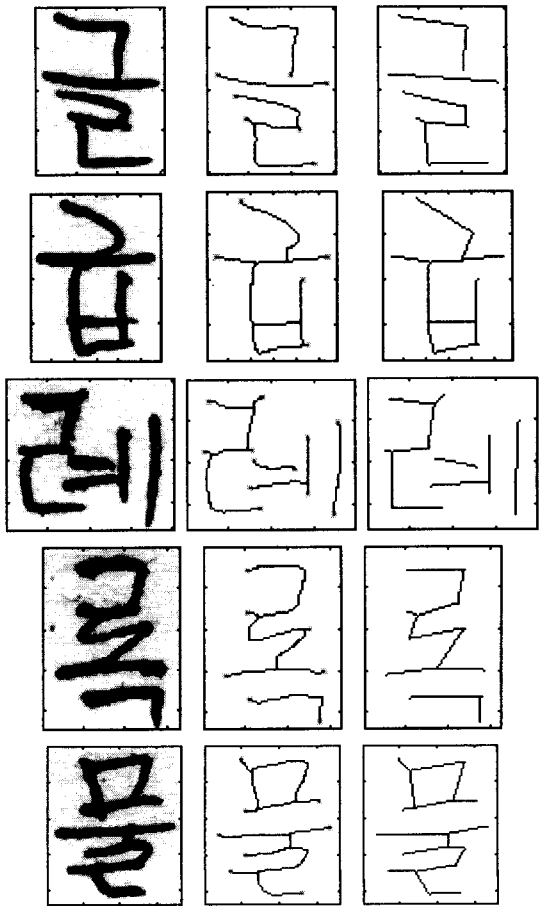


Fig. 7. The resut of stroke extraction

5. Conclusions

The two main purposes of the Korean character stroke extraction algorithms with thinning process are to find corner points and to handle junction points. The difficulty of finding the corner point is that the smooth transition corner and the abrupt transition corner exist in a Korean character. And the difficulty of handling the junction point is the violent distortion on the junction point produced by the thinning process. In this paper, we propose an intuitive and effective stroke extraction method for handwritten Korean characters. The proposed stroke extraction method consists of five stages: (1) binarization of the character using high-boost filtering and thinning. (2) searching and labeling the end points and junction points. (3) transcribing each primitive stroke from its one end point and recording the trace until reaching the other end point of the stroke or reaching the junction point that is judged to be the end of this stroke. (4) vectorization and splitting using maximum permission error method. And (5) merging the separated stroke using angle of vector. Based on a set of splitting, merging and writing direction rule, we can get a set of strokes from the given skeleton. The experimental results show that the proposed method is effective. This shows that the method used in this paper is useful and reliable.

References

[1] H. Chang and H. Yan, "Analysis of Stroke Structure of Handwritten Chinese Characters," IEEE Tranaction on Systems, Man .and Cybernetics-Part B: Cybernetics, Vol.29, No.1, February 1999.

[2] H. J. Kim and P. K. Kim, "Recognition of Off-line Handwritten Korean Characters," Pattern Recognition, Vol.29, No.2, pp.245-254, 1996.

[3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Co. Inc., 1992.

[4] J. R. Lin and C. F. Chen, "Stroke Extraction for Chinese Characters Using A Trend-Followed Transcribing Technique," Pattern Recognition, Vol.29, No.11, pp.1789-1805, 1996.

[5] P. Scattonlin, "Recognition of Handwritten Numerals using Elastic Matching," Master thesis, Univ. Concordia, Canada, 1995.

[6] T. Y. Zhang and C. Y. Suen, " A Fast Parallel Algorithm for Thinning Digital Patterns," Commun. ACM Vol.27, No.3, pp.236-239, Mar. 1984.

[7] H. E. Lu and P. S. P. Wang, " A Comment on 'A Fast Parallel Algorithm for Thinning Digital Patterns'," Commun. ACM Vol.29, No.3, pp.239-242, 1986.

[8] C. S. Bae and B. W. Min, "Automatic Drawing Input by Segmentation of Text Region and Recognition of Geometric Drawing Element," Journal of the KITE, Vol.31-B, No.6, pp.91-103, 1994

[9] S. W. Lu and H. Xu, "False Stroke Detection and Elimination for Character Recognition," Pattern Recognition Lett., Vol.13, pp.546-548, 1992

[10] C. M. Privitera and R. Plamondon, "A System for Scanning and Segmenting Cusively Handwritten Words into Basic Strokes," in Proc. 3rd ICDAR'95, pp.1047-1050 1995.