Stroke-based Online Hangul/Korean Character Recognition

Jinsu Jo[1], Jihyun Lee[1], Yillbyung Lee[1]

1. Department of Computer Science, Yonsei University, Seoul, Korea
E-mail: {hamster,jeeii}@csai.yonsei.ac.kr,
yillbyunglee@yonsei.ac.kr

Abstract: In this paper we propose a stroke recognition method for online handwritten Hangul recognition system. The proposed system extracts a distance-dependent curvature from two-dimensional original stroke data and achieves elastic matching between distance-dependent curvatures of reference and test characters. Elastic curvature matching has lower computational requirement than existing 2D-to-2D elastic matching. Each recognized stroke from the elastic curvature matching is converted into a Hangul syllable and additional position information is added to improve performance of the recognizer in this process.

Key Words: Stroke-based, Korean character, online character recognition

INTRODUCTION

As computers have been widely used in everyday life, character recognition technology is becoming more important as an intuitive input subsystem of computers. Besides, as electronic equipments like tablet PC, PDA, touch screen and PC cameras are becoming ever cheaper and being commonly used, researches on developing user-friendly input system with pen or hand gestures instead of keyboard become increasingly more valuable.

Online character recognition have several advantages: First, unlike offline character recognition, it provides both temporal and spatial information of a handwriting character, and gives additional critical information for the recognition, such as number, sequence, direction and velocity of strokes, pen pressure and so on. Second, it can exploit user feedback for enabling for users to correct or edit misrecognized characters, or restricting them to write inside the pre-defined area. Thus, there are far more online character recognition softwares than offline.

The recognition technologies, which are recognizing handwritten characters from such as the touch-screen in laptop computer and the electic-pen for PDA, are applicable to various uses: for instance, writing characters instead of typing keyboards, searching for the electric dictionary and verifying online signatures. Currently, there are a lot of active researches on character recognition, such as character input techniques without additional devices, like electric-pen or touch-screen, introducing motion recognition technologies using embedded cell-phone sensors.

On this ground online character recognition for tablet PC has been actively studied, and recognition performance in English characters and digit reached considerable level resulting in practical usages. For Korean characters much effort has been made as well, and commercial tablet PC has a built-in online Hangul/Korean character recognizer. However, compared to English characters or digits, its performance is not yet satisfactory. Hereupon, in order to provide Hangul users with more convenient and efficient input subsystem, improving its recognition performance of Hangul characters is required.

The main problems that we try to settle out are as follows: Stroke recognition is the first. Hangul characters are combinatorial in a way that some strokes constitute a consonant (or a vowel) and some consonants and vowels among 19 initial consonants, 21 vowels, and 21 final consonants constitute a character. The total number of such possible characters amounts to 11,172 not including some ancient characters occasionally used today.

The number of officially recognized Hangul characters in daily use is 2,350. But, they are all composed of two or three graphemes (individual vowels and consonants) and which are also composed from yet smaller number of basic strokes. Thus, reliable recognition of basic strokes (sometimes followed by graphemes) should precede recognizing such numerous characters.

Overcoming geometrical variations in cursive letters is the second. The most frequent problem is that it is difficult to extract invariant and stable features from all of the character variations due to differences among individual writers. Several techniques like statistical learning model, neural or fuzzy model, and elastic matching have been suggested to solve them out in mainly desktop computer or workstation. These methods on the system are massive, and are not suitable to mobile system like tablet PC that requires efficiencies of computations.

Following above two requirements, we extract distance-dependent curvatures from strokes and apply elastic matching of them into Hangul character recognition. While raw character data are given in two dimensions, the distance-dependent curvature is one-dimensional vector.

Since it is crucial to distinguish each stroke precisely for the stroke-based recognizer, the proposed recognizer uses curvature information from each stroke as the stroke feature.

978-1-4244-4199-0/09/$25.00 ©2009 IEEE

Each curvature from four basic strokes, ㄱ, ㄴ, ㅇ and ㅣ, has a very distinguished curve shape as shown in Figure 1, so it is suitable for the stroke recognition.

The proposed recognizer aims to recognize Hangul graphemes. For basic strokes, ㄱ, ㄴ, ㅇ and ㅣ, forming a Hangul character are recognized using curvature information of stroke, and each stroke recognized as ㅣ type then are classified as one of —, ㅣ, / and ＼ along with global slant (or angle). A basic stroke combination for a Hangul syllable, '말' is presented in Figure 2., An example of classifying a stroke recognized as ㅣ in Hangul syllable, '사' as one of —, ㅣ, / and ＼ is shown in Figure 3.

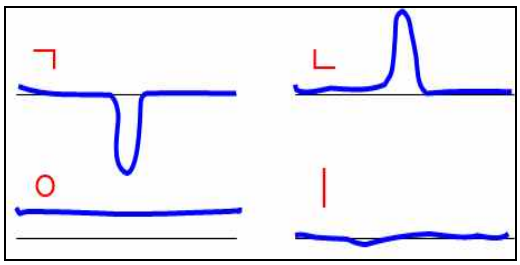


Fig. 1: Curvatures for four Hangul basic strokes

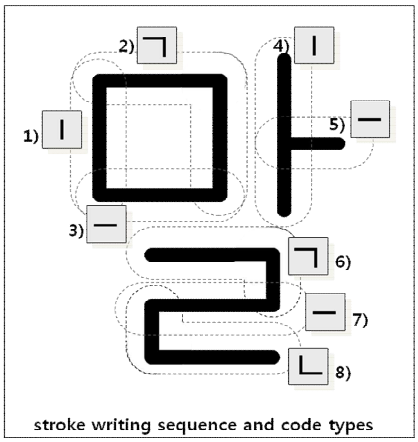


Fig. 2: Stroke combination for a Hangul syllable, '말'

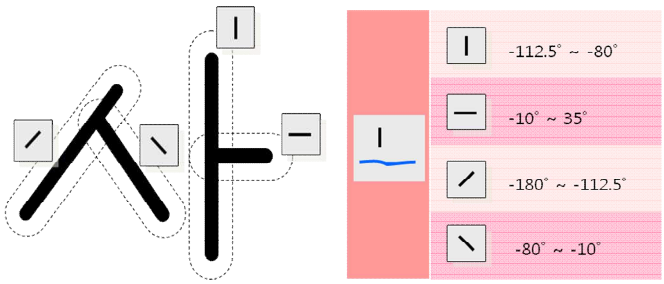


Fig. 3: Stroke combination of a Hangul syllable, '사' and the classification of 'ㅣ' along with global slant

Raw data is less time-consuming while the effectiveness of the elastic matching still holds. Besides, by the distance dependence, it provides more stable feature extraction stage irrespective of differences among individuals in pen tip speed, character size, and etc.

In post-processing unit, we suggest a cursive letter recognition module and predict probable next syllables based on n-gram statistics. This will provide alternative method for input characters faster and more accurately on your tablet pc or PDA. Because it reduces the number of strokes required to type any word. Here, we present a method of word prediction in combination with the current online-character recognition unit based on linguistic corpus and probability of bi-gram that is statistical frequency extracted by Sejong Database, KNC (Korea National Corpus).

PROPOSED MODEL

1. System Overview

Our recognition system consists of three modules as follows:

Stroke-based matching: When a Hangul character is purely divided into single basic strokes like 'ㄱ', 'ㄴ', 'ㅇ', 'ㅣ' and '—' without linking of adjacent strokes, this module is applied.

Grapheme-based matching: When the first module is failed to match patterns, this grapheme-based matching is attempted.

Bi-gram matching: After the matching process, a recommended pair of the next grapheme and syllable is presented.

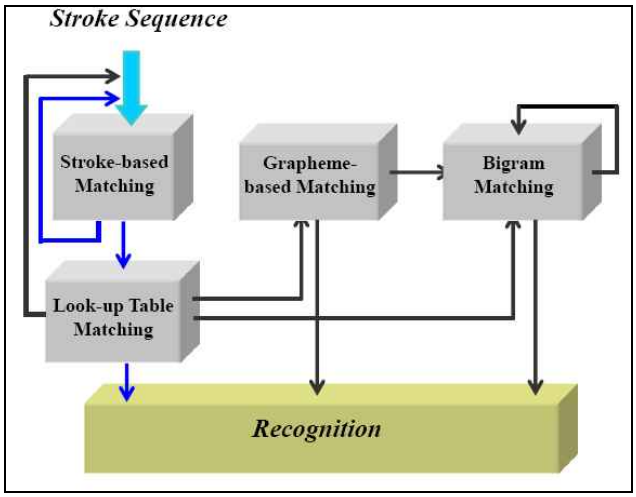


Fig. 4: System diagram

2. Collecting References

Our stroke-based recognition system needs various collections of Hangul basic strokes like 'ㄱ', 'ㄴ', 'ㅇ', 'ㅣ' and '—'. In order to accumulate such basic strokes, we produced a simple software program as shown in Figure 2. By using this 'reference constructor', Figure 5 examples some of basic strokes and Figure 6 shows the position vectors of a 'ㄱ' stroke.

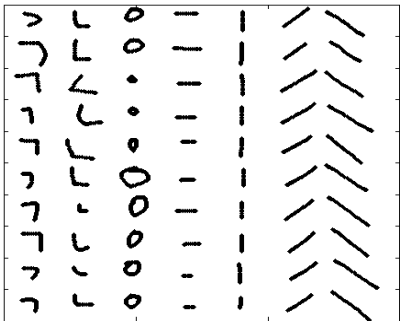
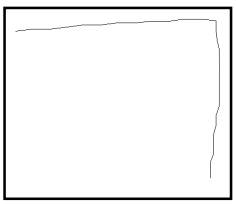


Fig. 5: Part of collected references



-0.0483660864705474	-0.0221003573354737
-0.0431301306396666	-0.0232120279869304
-0.0376695814868558	-0.0232120279869304
-0.0325308280126678	-0.0243236986383872
-0.0276082342474012	-0.0254353692898439
-0.0221476850945903	-0.0254353692898439
-0.017225091329237	-0.0265470399413006
-0.0117645421765129	-0.0265470399413007
-0.00662578870232492	-0.0276587105927574
-0.00116523954951413	-0.0276587105927574
0.00318363895183993	-0.0287703812442141
0.00864418810465075	-0.0287703812442141
0.0132088100882641	-0.0287744540750166
0.0132088100882641	-0.0224139049222058
0.0134664522001181	-0.0170024703160503
0.0140884514676757	-0.0116604934409863
0.0140884514676757	-0.00619994428817550
0.0142368911788057	-0.000791020246773377
0.0140884514676757	0.00353558652253549
0.0140884514676757	0.00899613567534630
0.0140884514676756	0.0144566848281571
0.0140884514676757	0.0199172339809679
0.0132088100882641	0.0252100963093767
0.0132088100882641	0.0306706454621875
0.0123291687088525	0.0359635077905962
0.0123291687088525	0.0414240569434070
0.0123291687088525	0.0468846060962178
0.0114495273294409	0.0514655138696170
0.0114495273294409	0.0569260630224278
0.0114495273294409	0.0601632708723251

Fig. 6: Shape and coordinates of a collected basic reference stroke, 'ㄱ'

3. Sequential Recognition of Basic Strokes

As shown in Figure 7, Hangul character recognition consists of preprocessing, curvature extraction, elastic curvature matching, look-up table searching.

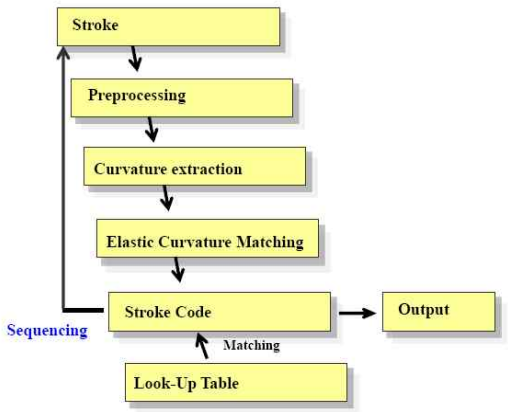


Fig. 7: Stroke-based matching and lookup table matching process

4. Pre-processing

Online stroke signals are given as sequences of two-dimensional points, and their equidistantly re-sampled, centered and normalized points are then provided for a stable curvature extraction method.

5. Curvature Extraction

We adapt distance-based curvature definition, which is slightly different from conventional curvature definition due to scale invariance, to the curvature extraction procedure.

Curvature is a geometrical quantity representing the state of being bent in curves or surfaces. As a point **P** moves along the curve in a constant speed, the arc distance **s** increases. The differential of the tangential angle with regard to **s** is the curvature.

We start our model with continuous time-dependent signals $x(t)$ and $y(t)$. The time t varies from 0 to T. The raw signals are transformed into the tangential angle and velocity:

$$\theta(t) = \tan^{-1}(\frac{dy}{dx}) \qquad \text{tangential angle}$$

$$v(t) = \sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2} \qquad \text{velocity}$$

Because we basically consider a single stroke and the raw signals are continuous, the tangential angle is not limited between 0 and 2π. For example, suppose that we get a circle by $x(t) = \cos(t + \pi/2)$ and $y(t) = \sin(t + \pi/2)$ from $t = 0$ to $t = 2\pi$, then the tangential angle can linearly increase like from $3\pi/2$ to $7\pi/2$.

This time-dependent angle is invariant to translation, but it depends on the tangential velocity. It could cause unwanted variations of curvature value even for the same shapes due to different velocities. Thus, we need to degenerate the variants of a stroke into a single unity. Let us consider the distance-dependent tangential angle:

$$\alpha(s) = \theta(t \mid s = \frac{\int_0^t v(t')dt'}{\int_0^T v(t')dt'})$$

Where s is the distance from the starting point on the curve of a stroke and is normalized to be a value between 0 and 1. This angle is obtained by solving inverse integral equation. Since it depends on the single variable, the solution is numerically given without iteration. Then the distance-dependent velocity $v^*(s)$ transformed from $v(t)$ becomes a constant. This means that the geometric information of the raw signals is transmitted into the distance-dependent tangential angle $\alpha(s)$ without loss of information. However, it still depends on rotation, and we differentiate the distance-dependent tangential angle $\alpha(s)$ into the curvature as follows:

$$k(s) = \frac{d\alpha(s)}{ds}$$

This curvature is invariant under translation, symmetric scaling, rotation and velocity of a pen tip. This gives more stable results than the time-dependent curvature $d\theta(t)/dt$.

The distance-dependent curvature from the number '6' is shown in the Figure 8 below.

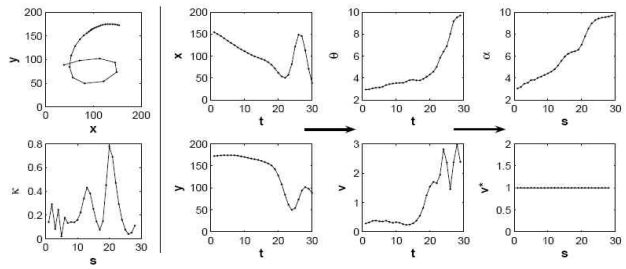


Fig. 8: Distance-dependent curvature: the curvature is invariant to translation, symmetric scaling, rotation, and tangential velocity of a pen tip. The velocity v*(s) is constant, and the distance-dependent curvature k(s) focuses on geometric shape of a stroke without loss of information

6. Elastic Curvature Matching

We suggest a new elastic matching technique using curvature vectors in this paper. One of the main problems of handwritten character recognition is how to deal with geometric deformations of characters. Elastic matching is a general group of matching techniques for solving the deformation problems.

Conventional application of elastic matching lies in working out an optimization problem by corresponding pixels to pixels between two two-dimensional domains. The traditional way of elastic matching is executed on the two-dimensional domain. However, although the elastic matching based recognition methods have shown good performance, they often suffer from misrecognitions due to overfitting, which is the phenomenon that the input pattern is closely fitted to the reference pattern of an incorrect category. Besides, it is rather time- and energy consuming to develop an online recognition system with practical use. Here, using the distance-dependent curvature, applying this one-dimensional elastic matching technique of the curvature vectors into Hangul strokes is the first process of our solution.

Consider two N-sampled curvatures k_c and k, where k_i^c and k_j are reference and test curvature components at time index i and j.

$$k_c = \{k_i^c \mid i = 1, \cdots, M\}$$
$$k = \{k_j \mid j = 1, \cdots, N\}$$

The elastic matching between reference curvatures and test curvatures is defined as an optimization problem.

The classification problem with this elastic matching between two curvatures components k_i^c and k_j is defined as the following constrained optimization problem. Constraints mean elasticity which set standard value 1.

$$G : k_c \rightarrow \arg\min_{c} \delta_c \quad \text{optimization problem}$$

$$\delta_c = \frac{1}{N}\sum_{i=1}^{N}(k_i^c - k_{j(i)})^2 \quad \text{error function}$$

$$j(i) \in \{i-1, i, i+1\} \quad \text{constraints}$$

Fig. 9: shows the schematic procedure in which 1D-1D elastic matching between curvatures is achieved.

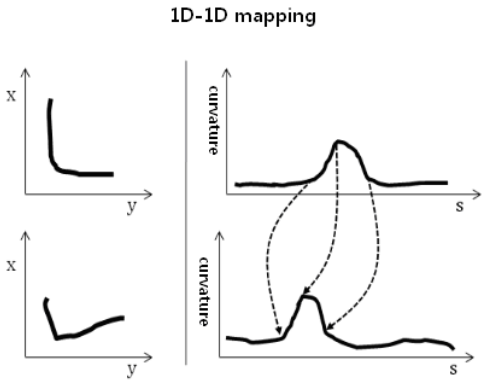


Fig. 10: Elastic curvature matching

7. Look-up Table Matching

In this stage, a Hangul character is finally recognized by matching stroke codes with a prepared table. The Hangul basic strokes like 'ㄱ', 'ㄴ', 'ㅇ', 'ㅣ', '—', '/' and '\' are allocated to 1, 2, 3, 4, 5, 6 and 7 respectively. And each of 11,000 syllables has a unique or non-unique code sequence. The following figures show the basic code allocation and some of code sequences.

Stroke	ㄱ	ㄴ	ㅇ	—	ㅣ	/	\
Code	1	2	3	4	5	6	7

Fig. 10: Single codes assigned to basic strokes

가	154	갸	1544	거	145
고	154	…	…	나	254
…	…	…	…	…	…
…	…	현	74..52	…	…
…	…	…	…	…	…

Fig. 11: Number sequences assigned to each grapheme

One syllable provided by the writer is transformed into code sequence after stroke recognition stage, and this code sequence is finally recognized by searching the same code sequence in the look-up table. Lookup table matching is not very theoretical, but even if one or two codes are misrecognized, Most of code sequences can be correctly recognized. This means that it is error-tolerant. However, the prepared table has a small portion of non-unique code sequences. For example, '가' and '고' has the same code sequence 154. To work out this trouble, we have to consider position information of strokes.

From the additional code, we can detect the medial parts of input Hangul character and their positional coordinates. Then the matching errors between the medial parts of input character and the medial parts of references of several candidates are calculated. According to the matching errors, we can finally sort the candidates that were given the same code. Figure 10 shows some of medial parts included for book references. Examples of both journal paper and book references are shown at the end of this template.

8. Post processing (Bi-gram)

Bi-gram, it would be advantageous to make the amount of storage and the computation time required independent of the dictionary. On the other hand, once storage is fixed and the dictionary grows sufficiently large, the performance of any error detection and correction algorithm will begin to suffer since all of the information in the dictionary is not retained. The evaluation of the method then becomes an evaluation of the tradeoff between the savings of reduced computation or storage with the increase in the error rate.

Most of the contextual algorithms that have performed well in the past did so at the expense of long computation times or a large amount of storage. They made use of a complete dictionary or else extracted the information in a probabilistic form in terms of the probability of trigrams and quad-grams. However, the following technique, developed bi-gram of extracting large amounts of the information from the KNC (Korea National Corpus) in a readily retrievable form at a relatively modest cost of storage. This data was verified by the national institute of the Korean language which is the national academy of the Korean language. So this system will enhance the Korean character recognition.

CONCLUSIONS AND FUTUWORKS

Elastic curvature matching (ECM), the modification that we actually applied into our system, is also attempting to find a reference curvature that best matches a given test curvature. However, recognition performance crucially depends on choice of references, and we can insert a learning component like 'clustering' by which efficient references can be selected from a training data set. Furthermore, it is possible to apply elastic matching technique into existing learning model (i. e. Eigen deformation: application of EM into PCA), which will be our future work.

ACKNOWLEDGEMENT

This project was supported by Microsoft Research Asia.

REFERENCES

[1] J.-H. Ahn, J. Lee, J. Jo, Y.-H. Choi, and Y. Lee, Online Character Recognition using Elastic Curvature Matching, IEEE proceedings of ICAPR 2009.

[2] J. Lee , J.-H. Ahn, Y. Lee, Elastic curvature matching for online handwritten Hangul recognition, Proceedings of Korean Information Sciences Society, v25, 2008, p238-239.

[3] A. Jain. Representation and recognition of handwritten digits using deformable templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(12):1386–1391, 1997.

[4] H. Mitoma, S. Uchida, and H. Sakoe. Online character recognition using eigen-deformations. IEEE Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition, 2004.

[5] K. T. Miura, R. Sato, and S. Mori. A method of extracting curvature features and its application to handwritten character recognition. Fourth International Conference Document Analysis and Recognition, pages 450–455, 1997.

[6] M. Shia, Y. Fujisawab, T. Wakabayashia, and F. Kimura. Handwritten numeral recognition using gradient and curvature of gray scale image. Pattern Recognition, 35:2051–2059, 2002.

[7] S. Uchida and H. Sakoe. Eigen-deformations for elastic matching base handwritten character recognition. Pattern Recognition, 36:2031–2040, 2003.

[8] S. Uchida and H. Sakoe. Survey of elastic matching techniques for handwritten character recognition. IEICE Transactions on Information and Systems, E88-D:1781–1790, 2005.

[9] Peter F.Brown, Peter V. deSouza, Robert L. Mercer,Vincent J. Della Pietra, Jenifer C. Lai, "Class-based n-gram models of natural language", Computational Linguistics, Volume 18, pp. 467-479, 1992.

[10] G. Lesher, B. Moulton, and J. Higgonbotham, "Effects of ngram order and training text size on word prediction.", In Proceedings of the RESNA '99 Annual Conference, 1999.

[11] William B. Frakes , Ricardo Baeza-Yates, "Information retrieval: data structures and algorithms", Prentice-Hall, Inc., Upper Saddle River, NJ, 1992

[12] Hangul database http://csai.yonsei.ac.kr/research.htm.