International Journal of Evaluation and Research in Education (IJERE)
Vol.1, No.1, July 2012, pp. 17~24
ISSN: 2252-8822

Learner-Friendly Kanji Learning System with Radical Analysis

Jungpil Shin, Yusuke Shimizu
Graduate School of Computer Science and Engineering
The University of Aizu, Fukushima, Japan

Article Info

Article history:

Received Apr 11, 2012
Revised May 23, 2012
Accepted June 12, 2012

Keyword:

Kanji learning system
Radical analysis
Stroke correspondence

ABSTRACT

This paper presents a novel friendly Kanji learning system using Radical Analysis to enable foreign people and elementary school students to learn Kanji by an interesting and efficient way. This way is for characters to analyze for each radical, to divide into some parts, and to correct strokes for each divided part. Here, the Radical Analysis Database (RAD) is used for dividing characters. RAD is a database to analyze characters for each radical and divide into some parts. On the other hand, characters are corrected by using a threshold. The threshold is a judgment value in the correction and learners can set it freely by handling threshold bars put on the interface. Then, the novel system is improved so that learners can set thresholds for each divided part. Since each bar corresponds to each part, the system judges whether each part is corrected or not according to set thresholds. Hence, since learners can freely determine radicals or parts in which they want to be instructed intensively, they can practice only their radicals not good or part of the character and easily master difficult characters, too. In addition, an animation helps learners understand the order of strokes virtually. Since each stroke used in this animation is displayed with different colors, learners can also understand virtually where the same strokes are from and to at once.

Corresponding Author:

Jungpil Shin

Graduate School of Computer Science and Engineering,
The University of Aizu, Fukushima, Japan
E-mail: jpshin@u-aizu.ac.jp

1. INTRODUCTION

Since a lot of Kanji are used in Japan, it is essential to master them in primary education. But as it stands, it is difficult to have an enough education in the limited time, such as classes. On the other hand, it is essential for foreign people in Japan to read and write Kanji, too. However, it is not easy for them to master Kanji in a short time. Hence, software that can aid learners in mastering them efficiently and in a short period is needed.

For this, research and development of Kanji learning systems was carried out by various styles. Tatsuoka and Yoshimura have developed the Kanji education support system intended for foreigners and the elementary school children [9]. Since it has a graphical interface, it is easy to use this system. Takesue develops a system enabling teachers to designate evaluation points [7]. This system has functions of not only evaluating stroke orders and character shapes but also enabling teachers to designate evaluation points and replace the model of character patterns easily and freely. So it is considered in not only learners but also teachers. Other, systems considered that enable learners to master via games have also been developed [3] [4].

The model of Kanji learning system I am studying is to correct strokes by comparing reference characters with input characters using inter-stroke information. Moreover, Radical Analysis is introduced to this system. This system analyzes characters for each radical and divides them logically into some parts. The Radical Analysis Database (RAD) is used for Radical Analysis. RAD is a database which analyzes characters for each radical and divides into some parts. This method enables learners not only to correct for each part but also to practice only parts they want to, and thus flexible and wide learning is realized.

2. RELATED STUDIES

There are some programs using real-time search of relation with correct stroke by Cube search, modification of a letter-shape using inter-stroke information, and instruction of the order of writing using animation [8], and setting a threshold bar [5]. Here, the threshold means a judgment value in character correcting.

2.1 Stroke Correspondence Search

An input character is expressed as an ordered series of strokes as follows:

$$A = A_1 A_2 A_3 \cdots A_k \cdots A_N$$

where the k-th stroke A_k is the time sequence representation of local feature a_{ik} of a character, e.g., x-y coordinates or stroke direction are expressed as

$$A_k = a_{1k} a_{2k} \cdots a_{ik} \cdots a_{Ik}, \quad I = I(k)$$

The reference pattern is similarly expressed as

$$B = B_1 B_2 B_3 \cdots B_k \cdots B_N$$

and

$$B_k = b_{1l} b_{2l} \cdots b_{jl} \cdots b_{Jl}, \quad J = J(l)$$

where N is the number of strokes of a character.

One-to-one stroke correspondence is defined by bijection l(k) to stroke number k of the input pattern. The measure of dissimilarity between the input pattern stroke A_k and the reference pattern stroke B_l is calculated using stroke information on the shape and position. The measure of dissimilarity is defined as δ(k,l) and called the stroke distance.

The sum of the distance δ(k,l) between strokes is used as a valuation basis of the optimal correspondence. The minimum value D(A,B) is calculated by the following formula, and the stroke correspondence l(k) is obtained as the following result:

$$D(A,B) = \min_{\{l(k)\}} \left[\sum_{k=1}^{N} \delta(k, l(k)) \right]$$

These processes for all stroke distances require an amount of calculation of the factorial order of the number of strokes and are unreal. Then, the 3rd Markov cube is used by the result that 3rd is enough from preliminary experiment [6]. Furthermore, the amount of calculation is decreased for the index order from the factorial order using a beam search. The stroke correspondence search in real time is possible, and the wrong order of a stroke can be pointed out precisely.

2.2 Inter-Stroke Information

As a geometric feature of a character, a relative arrangement relation exists between strokes. Moreover, the relative merits and demerits between strokes can become as important work for discernment between some characters (Figure 1). Such information is defined as inter-stroke information.

The vector connects the i-th representation point of stroke k-th of an input pattern and the j-th representation point of stroke p-th as

$$d_{ij}(A_k, A_p)$$

The vector connects the i-th representation point of stroke l-th of a reference pattern and the j-th representation point of stroke q-th as

$$d_{ij}(B_l, B_q)$$

The Inter-stroke information ρ is expressed following formula,

$$\rho(k,l;p,q) = \sum_{i=1}^{m} \sum_{j=1}^{m} R(d_{ij}(A_k, A_p), d_{ij}(B_{l(l)}, B_{l(q)}))$$

where m is the number of the representation point, and R (・, ・) are the weighted sums of the angle difference (θ) of vector and the difference (e) of the length of a vector.

The inter-stroke information is considered as a measure of dissimilarity of the position relation of each stroke of the input pattern and the position relation of each stroke of the reference pattern.

The relative position relation with the representation point of other strokes is the one of the evaluation values by m = 5, i.e., five points that are equally estimated from each stroke. Therefore, this system cannot catch the local features, such as “bounding” and “bending”.

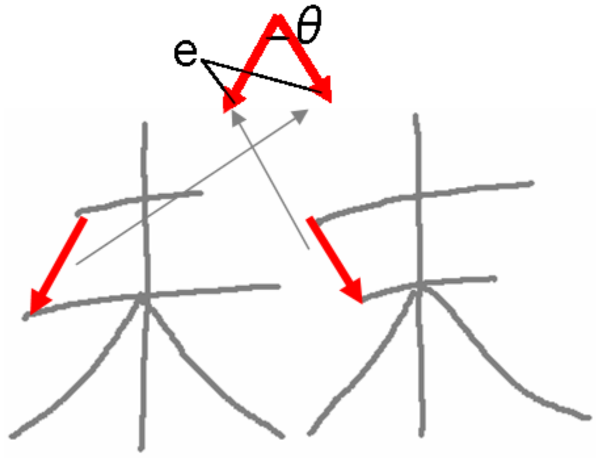

Figure 1. Inter-Stroke Information

2.3 Instruction of the Character Using Inter-Stroke Information

The inter-stroke information uses as a valuation basis, and the evaluation value ρ is considered which the stroke of a larger value than a threshold breaks down the character. Such a stroke is corrected by the instruction. k' is the drawing to which ρ becomes larger than the threshold, and k'-1 is the stroke in front of k'. The number of a representation point is set to 5, the vector

$$d_{11}(B_{k'-1}, B_{k'}), d_{22}(B_{k'-1}, B_{k'}), \cdots, d_{55}(B_{k'-1}, B_{k'})$$

is displayed in piles on an input pattern as correction directions.

For example, the characters “right” and “stone” shown in Figure 2 instruct to correct the relational position of the 1st stroke and the 2nd stroke. d_{11} is the vector which connects the 1st representation points of the stroke and d_{22} is the vector which connects the 1st representation points of the stroke.

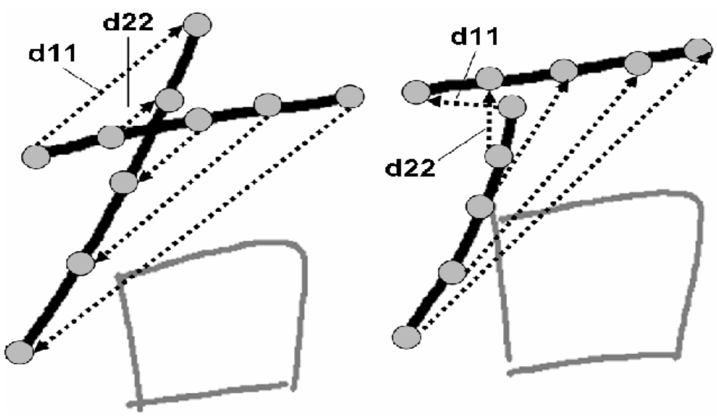

Figure 2. Instruction of the Character using Inter-Stroke Information

2.4 Threshold

When characters are corrected, the number of correction strokes is various according to number of strokes and Kanji. So the number of strokes which should be corrected is changed and character instruction is performed using a threshold. Here, the threshold is the judgment value when character is corrected.

Each characteristic point of reference and input strokes are subtracted and calculated as absolute value, and threshold d is calculated using the following formula.

$$d = \frac{s+m+e}{3}$$

$$s = \frac{|s(x)_{ref} - s(x)_{inp}| + |s(y)_{ref} - s(y)_{inp}|}{2}$$

$$m = \frac{|m(x)_{ref} - m(x)_{inp}| + |m(y)_{ref} - m(y)_{inp}|}{2}$$

$$e = \frac{|e(x)_{ref} - e(x)_{inp}| + |e(y)_{ref} - e(y)_{inp}|}{2}$$

where s, m, and e is characteristic points of each stroke (That is, starting point, middle point, and end point), x and y is x-y coordinates, ref and inp are reference and input strokes.

2.5 Radical Analsys Database (RAD)

According to the definition, Radicals are the part of characters and all characters have not less than a radical. On the other hand, parts except radicals also have characteristics like radicals. So Radical Analysis can divide characters logically. All characters are analyzed by the Database of Features of Characters, Character Information Service Environment (CHISE) project [1] and composed in a new database called Radical Analysis Database (RAD). Figure 3 shows examples of division types by RAD. From (a) to (e) are examples of dividing into two parts. (f) is an example of dividing into three parts. Finally, (g) is an example of no division.

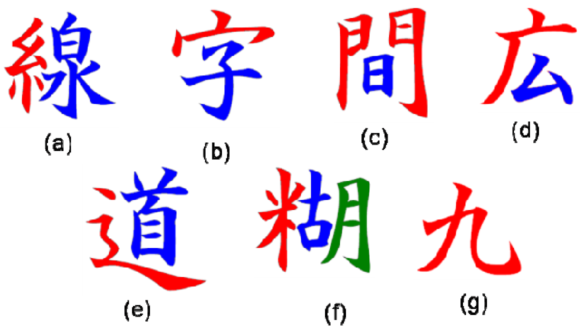

Figure 3. Examples of Division Types by RAD (a) two parts divided perpendicularly,(b) two parts divided horizontally, (c) kamae, (d) tare, (e) nyo, (f) three parts, (g) no division

3. KANJI LEARNING SYSTEM

The following functions are mentioned in this paper.

1. Correction with Radical Analysis
2. Animation Improvement

3.1 Interface

The interface of this system is shown in Figure 4.

(a) The study range is changed.
(b) The language of instruction to either Japanese or English is changed. Using English, it is easier for many foreigners to learn.
(c) This system is exited.
(d) A reference character is displayed.
(e) Learners write the character here with pen and tablet.
(f) Another character is displayed on the reference character column using the “Back”, “Next”, and “Random" buttons. A stroke correspondence table is expressed as a “Check” button, and animation is expressed as the “Anime” button.
(g) Animation is displayed.

(h) The incorrect writing order is displayed. The color of wrong strokes in the character input column turns red.
(i) A stroke correspondence relation is displayed.
(j) By inputting the character with the keyboard into this box, learners can quickly refer to any character they want to learn.
(k) Within the range list, learners can quickly refer to any character they want to learn.
(l) By threshold bars, learners can change the level of character instruction. When set to "STR", the judgment of character instruction becomes strict and most characters are corrected. When set to "LSE", the judgment of character instruction becomes loose and only the strokes which should be corrected are done according to desired thresholds. There are often no corrected strokes according to stroke numbers and strokes.
(m) The meaning of learning character is displayed. The Japanese and English definitions are prepared corresponding to both languages, respectively.

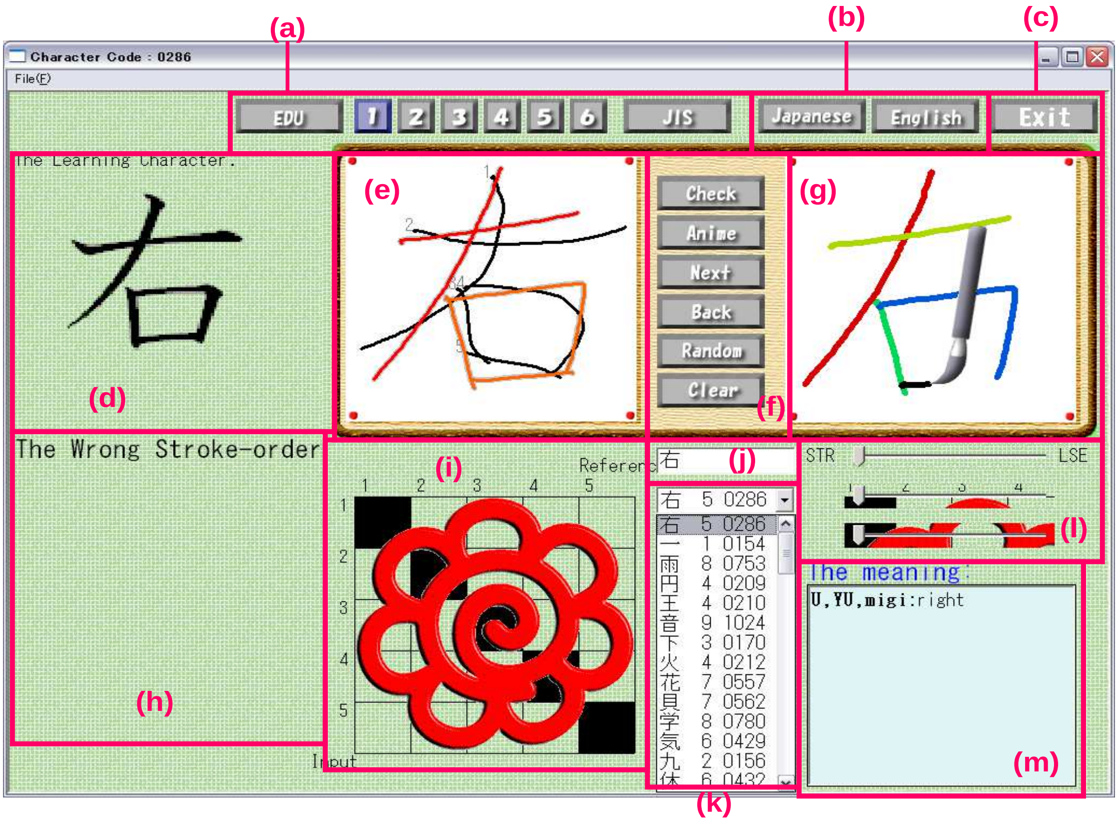

Figure 4. Interface of Kanji Learning System
(a) Radical practice mode button, (b) Study range selection button, (c) Language change button, (d) Exit button, (e) Reference pattern, (f) Character input column, (g) Operation button, (h) Animation display column, (i) Wrong stroke order display column, (j) Stroke correspondence table, (k) Character list, (l) Threshold bar, (m) The meaning display character column

3.2 Flow of Study

1. Click the study range selection button.
2. Select a character that you want to study from list.
3. Set thresholds freely with threshold bars.
4. Write the character checking the reference character. If you do not know its stroke order, click the "Anime" button, and you can check it.
5. The stroke correspondence is examined by the "Check" button of the operation button. If the number of strokes is wrong, the message window is displayed. And if the order of strokes is wrong, the color of the stroke turns red and the correct order is shown. Then, push the "clear" button, and the system returns to step 3.

6. Without any mistake at the number of strokes and stroke orders, the character is corrected and the system returns to step 1.

3.3 Correction with Radical Analysis

This system has 2965 characters with total from 1 to 25 strokes. Then, in order to examine how many characters are dividable, the number of dividable and undividable characters was examined. Here, this paper defines characters which can be divided as "dividable characters" with RAD and characters which cannot be divided as "undividable characters". As a result, totally 87% characters are dividable as shown in Figure 5. On the other hand, the other 13% characters are undividable. These undividable characters are corrected without Radical Analysis.

There are three thresholds, d1, d2, and d3, and three threshold bars to set each threshold individually. Since RAD divides into at most three parts, there are also three thresholds. Each threshold is for the threshold of, a red part, a blue part, and green part in order (Figure 6). Learners can set them by moving each threshold bar. The system judges whether each part is corrected or not according to set thresholds. Here, d3 is not used in two divisions and d2 and d3 are not used in undividable to the correction judgment. Therefore, if learners set unused thresholds, there is no influence to the correction judgment. For this, learners can freely set parts they want to practice intensively.

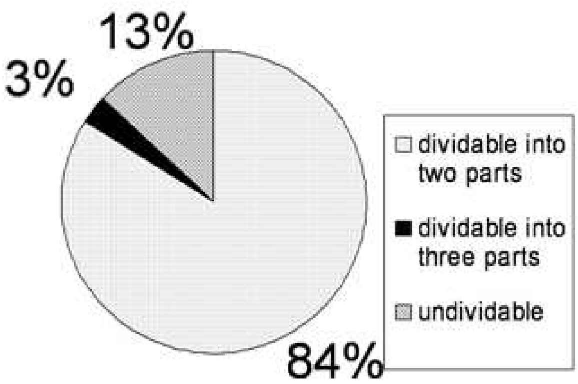

Figure 5. The Number of Dividable and Undividable Characters

Figure 6. Each threshold for Each Part

3.4 Animation Improvement

Learners can understand the order of strokes virtually by animation. In the old system, only one color is used for stroke colors in animation. However, it is not explicit where the same strokes are from and to. Therefore, the novel version is improved that each stroke is displayed with different colors (Figure 7). HSV model is used for color change and the color of the *k*-th stroke is calculated by the following formula:

$$H = \frac{360k}{N} \quad (N \le 12)$$

$$H = \frac{360k}{N} \% 360 \quad (N > 12)$$

$$S = 1.0, \quad V = 0.85$$

where *H* is hue, *S* is saturation, *V* is value, *N* is the number of all strokes. By setting low values it makes it easier to see light colors such as yellow. And characters with more than 13 strokes are limited to 12 colors because of the small difference in colors. For this, learners can understand virtually not only the order of strokes by animation but also where the same strokes are from and to.

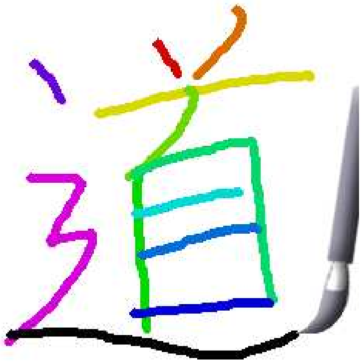

Figure 7. Stroke Colors in Animation

4. EXPERIMENTAL RESULT

Figure 8 shows an experimental result when two thresholds, d_1 and d_2, are changed. The number of the first part of corrected strokes is decreasing as d_1 is loosing and d_2 does not change. The number of the second part of corrected strokes is decreasing as d_2 is loosing and d_1 does not change vice versa. Therefore, it is clear that the number of corrected strokes is changing by changing thresholds when the same character is written.

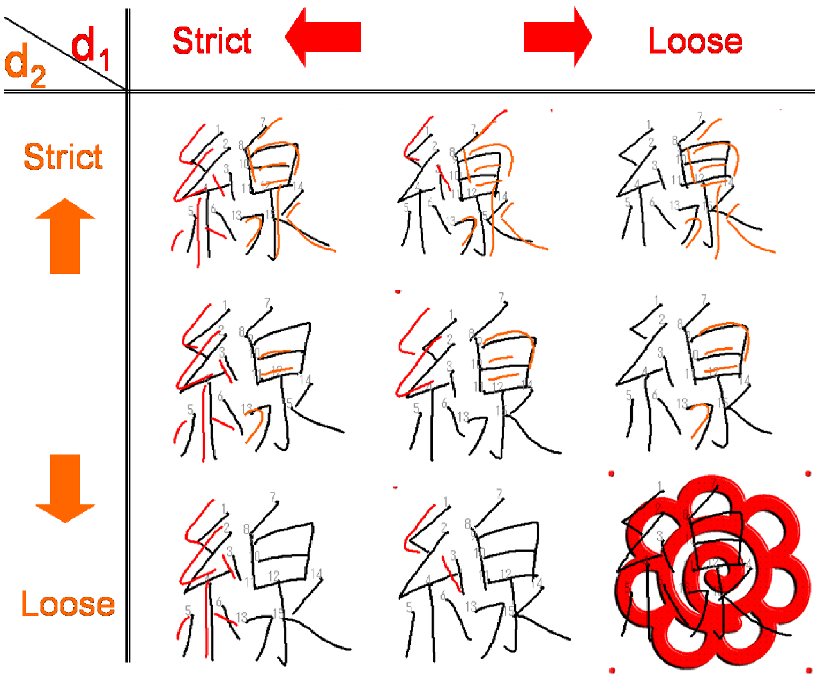

Figure 8. Corrected characters according to the instruction level

5. DISCUSSION

Though it is very important in Kanji learning to direct tightly the local features such as tome, hane, harai, and bending, this system cannot catch them. Therefore, when the character shape is correct but local features are wrong, this system sometimes does not correct them. So a novel algorithm to abstract and instruct local features is necessary.

Next, since the system evaluates written characters with the number of corrected strokes, it is hard to know how good the characters are. So it is very important to improve more understandable evaluation for them. For this, it is very important to pilot this system.

In addition, more characters can be dividable by modifying RAD, such as increasing division patterns. Otherwise, there are some elements to modify which confuse learners; some reference characters are recorded with wrong stroke orders, others are recorded with clearly strange meaning.

6. CONCLUSION

This paper has presented a way of stroke correction using Radical Analysis. The system can correct strokes for each part by dividing characters into some parts with Radical Analysis. At the same time, learners can set each threshold individually for each part. By setting plural thresholds freely, learners can freely determine parts they want to practice intensively. From now on, it is necessary to improve so that the system can abstract and instruct local features. In addition, it is necessary to modify some basic database, meaning and stroke orders. The only antidote for this is a steady manual labor.

REFERENCES

CHISE project http://www.kanji.zinbun.kyoto-u.ac.jp/projects/chise/

A. Hasegawa, J. Shin, "Handwritten Style Font Generation Using Radical Information", Mar. 2005.

T. Hayashi, Y. Hayashida, "KanjiMaster: Kanji Learning System with a Part Structure Assembly Game", *Transactions of Japanese Society for Information and Systems in Education*, Vol.19, No.4, pp240-245, Oct. 2002.

T. Hayashi, Y. Oda, K. Sawada, Y. Hayashida, "Development of a Game Style Drill System for Learning Kanji Strokes", *Transactions of Japanese Society for Information and Systems in Education*, Vol.18, No.1, pp7-15, Apr. 2001.

Y. Nagano, J. Shin, "Improvement of Chinese Character Study System Using Threshold ", Mar. 2007.

J. Shin, H. Sakoe, "Stroke Correspondence Search Method for Stroke-Order and Stroke-Number Free On-Line Character Recognition-Multilayer Cube Search-", *IEICE Trans. & Syst.* , vol.J82-D-II, No.22, pp.230-239, Feb. 1999.

N. Takesue, K. Mochida, A. Kitadai, M. Nakagawa, "A handwriting-based Kanji learning system enabling teachers to designate evaluation points", *IPSJ SIG*, 2005-CE-78, pp15-22, Feb. 2005.

Jungpil Shin and Atsushi Takeda, "Character Learning System Using Inter-stroke information,"the Eight International Conference on Knowledge-Based Intelligent Information, pp. 165-174, Sep. 2004, Wellington, New Zealand.

R. Tatsuoka, M. Yoshimura, "Development of A Kanji Learning System for Foreign Students or Elementary Students", *IEICE Trans. & Systv*, ET96-36, Jun.1996.

Author Biographies

Jungpil Shin received a B.A. in computer science and statistics and an M.S. in computer science from Pusan National University, Korea in 1990 and 1994, respectively. He received a Ph.D. in communication engineering from Kyushu University, Japan in 1999. He became an assistant professor and associate professor in the Department of Computer Software, the University of Aizu, Japan, in 1999 and 2004, respectively. His research interests include pattern recognition, character recognition, image processing, and computer vision. He is currently researching the following advanced fields: pen-based interaction system, realtime system, oriental character processing, mobile computing, computer education, human recognition, and machine intelligence. He is a member of the IEEE Computer Society; Information Processing Society of Japan; Institute of Electronics, Information, and Communication Engineers; and Korea Information Science Society.

Yusuke Shimizu received a B.A. and M.S. in computer science and engineering from University of Aizu, Japan in 2008 and 2010, respectively. His research interests include human computer interaction, signature verification, pattern recognition, image processing and computer vision.