

Handwritten Character Recognition in the Air by Using Leap Motion Controller

Kazuki Tsuchida, Hidetoshi Miyao^(✉), and Minoru Maruyama

Computer Science and Engineering, Shinshu University, Nagano, Japan
{miyao, maruyama}@cs.shinshu-u.ac.jp

Abstract. In order to develop a system which can precisely and quickly recognize handwritten characters in the air by using a Leap Motion Controller, we propose the following method: (1) A user has to register handwritten characters as template patterns before use. Each pattern is represented by a sequence of motion vectors calculated by using adjacent sampling data. (2) In the recognition phase, an input pattern is represented in the same method as above. The input pattern is compared with each of the registered template patterns by using DP matching and we can obtain a distance (degree of similarity) between them. Our system outputs the character class corresponding to the pattern with a minimum distance as a recognition result. In our experiments for recognition of 46 Japanese hiragana characters and 26 alphabets, a high average recognition rate of 86.7 % and a short average processing time of 196 ms were obtained.

Keywords: Character recognition · Leap motion · DP matching

1 Introduction

When a user enters character information to a computer, in general, he/she uses a computer keyboard and a mouse. However, if he/she wears a head-mounted display or his/her hands are too dirty, he/she could not use the devices. Even in such cases, users may want to enter character information. For example, they may want to search the Web while they are cooking and their hands are too dirty to touch a keyboard. Therefore, the purpose of our study is to develop a system which can recognize handwritten characters in the air by using a hand motion-sensing device.

In related works, personal authentication systems using hand or pen gestures in the air have been developed [1, 2]. On the other hand, a recognition method of handwritten alphabet characters in the air by Leap Motion controller has been proposed [3]. In this method, many people registered variety shapes of alphabet characters beforehand and they are used as template patterns in the character recognition phase. When a word is handwritten, the system allows users not to denote pen down (start) and pen up (end) points of writing a character. However, the system has the following drawbacks:

- In order to get a high recognition rate for many kinds of user input, many template patterns are needed.
- Since a character segmentation process is needed, the search area of character recognition phase is expanded.

Therefore, it is more likely to decrease recognition rates and slow down response time. To solve these problems, our system adopts the following ways:

- We assume that our system is used personally and the user has to register handwritten characters as template patterns before use.
- We introduce gestures which can indicate start /end points of writing a character.

Moreover, in addition to recognizing alphabet characters, we try to deal with recognizing Japanese hiragana characters whose shapes are more complex than those of alphabets.

2 Entering Characters and Their Representation

In order to get information of handwritten characters in the air, we use Leap Motion controller [4] which is a hand motion-sensing device. It can detect 3D positions of human hands and 10 fingers with up to 0.01 mm accuracy. Its sensor field is a roughly hemispherical area above the device, whose diameter is about 0.6 m. Using the device, an input character pattern is sampled every 75 ms and it is recorded by time series data which consists of the center positions of a palm of a user's hand at the sampling points. The palm position is represented by the x and y coordinates, in other words, we ignore z coordinate (depth direction) since characters are mainly handwritten on the x-y plane.

To segment each handwritten character, the first data is captured when the user flexes his/her thumb and the data acquisition process is completed when his/her palm stays in the air for 225 ms. Finally, the input pattern is represented by a sequence of motion vectors calculated by using adjacent data.

In the registration phase, handwritten characters represented by the above method are registered as template patterns. On the other hand, in the recognition phase, an input pattern is also captured and represented in the same method as shown in Fig. 1.

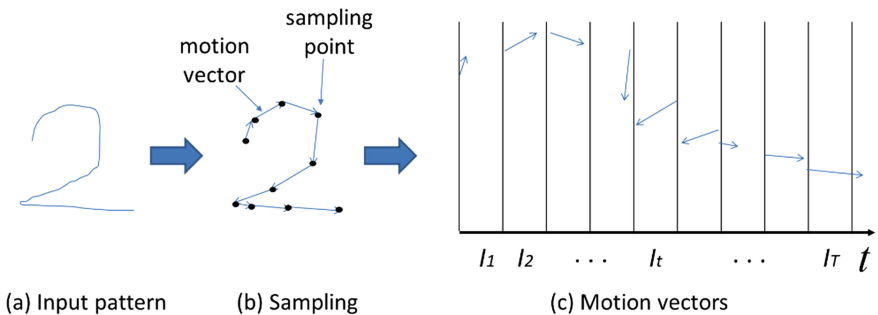


Fig. 1. Input pattern and its motion vectors

3 Character Recognition Using DP Matching

A user has to register handwritten characters as template patterns before use. After that, an input pattern written by the same user is compared with each of the template patterns and the most similar pattern is found in the recognition phase. In this comparison process, we use dynamic programming matching (DP matching) since it can properly calculate the similarity between patterns even when the numbers of motion vectors for both the patterns do not coincide with each other.

Next, we explain a matching method between one of the template patterns and an input pattern. For template pattern c and the input pattern, time series motion vectors of them are represented by J_1^c, \dots, J_T^c and I_1, \dots, I_T , respectively. If the t^{th} motion vector of the input pattern coincides with the τ^{th} motion vector of the template pattern c , the local cost of DP matching is calculated by the following equation:

$$d_c(t, \tau) = \|J_\tau^c - I_t\|^2. \quad (1)$$

The accumulated distance $g_c(t, \tau)$ is calculated as follows:

Initial value:

$$g_c(t, 1) = \begin{cases} 3d_c(t, 1), & (t = 1) \\ g_c(t-1, 1) + 3d_c(t, 1), & (t > 1) \end{cases} \quad (2)$$

$$g_c(t, 2) = \min \begin{cases} g_c(t-2, 1) + d_c(t-1, 2) + 2d_c(t, 2) \\ g_c(t-1, 1) + 2d_c(t, 2) \\ g_c(t, 1) + 2d_c(t, 2) \end{cases} \quad (3)$$

Recurrence formula ($\tau \geq 3$):

$$g_c(t, \tau) = \min \begin{cases} g_c(t-2, \tau-1) + d_c(t-1, \tau) + 2d_c(t, \tau) \\ g_c(t-1, \tau-1) + 2d_c(t, \tau) \\ g_c(t-1, \tau-2) + d_c(t, \tau-1) + 2d_c(t, \tau) \end{cases} \quad (4)$$

Using the above formulas, the minimum accumulated distance $g_c(T, T)$ is calculated. To reduce the effect of difference number of motion vectors, this value is normalized as follows:

$$G_c(T, T) = \frac{g_c(T, T)}{3T}. \quad (5)$$

For all template patterns, the minimum accumulated distances are calculated and they are sorted in ascending order. The system outputs the character class corresponding to the pattern with the minimum distance as a recognition result. Moreover, the character classes for the 2nd, 3rd, 4th and 5th minimum distance are also output as recognition candidates.

4 Experimental Results

Our system was developed by Microsoft Visual Studio C#. Figure 2 shows the screenshot of the system when one of the Japanese hiragana characters is entered. The left large window displays trajectory pattern of the handwritten character and the middle large window shows the corresponding recognition result. The four small windows on the right represent the 2nd, 3rd, 4th, and 5th candidates.

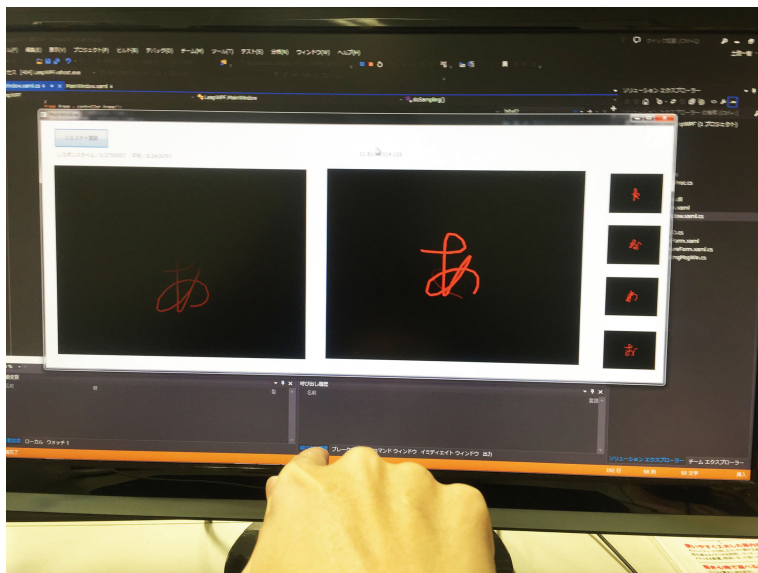


Fig. 2. An example of system output for handwritten character

In our experiment, 5 test users registered 46 Japanese hiragana characters and 26 lower-case letters of the alphabet as template patterns and they also wrote each character three times as test patterns. Table 1 shows the recognition rates for the test patterns. In total, a high average recognition rate of 86.7 % were obtained. However, some confusions occurred for several patterns. The error-prone types are described below the table. Since each character is represented by one stroke even if the character consists of 2 or more strokes, it is difficult to distinguish patterns which differs only in the connection part of strokes. For this reason, the input patterns “e and t” tend to be misrecognized each other.

Next, we will discuss about the processing time. The average processing time from a user finishes writing a character till the recognition result is displayed, was 196 ms by using a PC (CPU: Intel Core-i7-4796, 3.60 GHz and Memory: 32.0 GB). Therefore, the system can operate in real time.

Finally, to evaluate the usability of the system, we heard opinions of the test users. The obtained opinions are as follows:

- Response time is fast.
- Since the system can register shape and writing order of input characters personally, the recognition rate tends to be high. However, the registration process is tedious task.
- Thumb gets tired as many characters are entered.
- It is difficult to get the timing of the start point of character input.

Table 1. Recognition rates for each character

Japanese	あ	い	う	え	お	か	き	く	け	こ	さ	し	す
rate[%]	100	87	67	87	100	100	100	93	87	60	67	87	73
Japanese	せ	そ	た	ち	つ	て	と	な	に	ぬ	ね	の	は
rate[%]	93	80	87	87	80	73	100	80	87	100	93	73	80
Japanese	ひ	ふ	へ	ほ	ま	み	む	め	も	や	ゆ	よ	ら
rate[%]	93	100	93	100	80	93	73	87	100	87	93	100	53
Japanese	り	る	れ	ろ	わ	を	ん						
rate[%]	93	100	100	93	100	93	87						
Alphabet	a	b	c	d	e	f	g	h	i	j	k	l	m
rate[%]	80	60	93	87	60	73	93	93	93	93	100	93	73
Alphabet	n	o	p	q	r	s	t	u	v	w	x	y	z
rate[%]	87	87	47	67	100	100	73	87	93	93	93	87	93

Types of conflict : “う, ら”, “こ, て, z”, “さ, せ”, “a, q”, “b, p”, “e, t”

5 Conclusion

To develop a system which can precisely and quickly recognize handwritten characters in the air by using Leap Motion controller, we have proposed the system where a user can register characters as template patterns in advance, the input characters are segmented properly by pre-defined gestures, and they are recognized by DP matching method. In our experiments, for 46 Japanese hiragana characters and 26 alphabets, a high average recognition rate of 86.7 % and a short average processing time of 196 ms were obtained. Therefore, it should be suitable for practical use. However, we have to improve the system according to the test users’ opinions.

References

1. Kratz, S., Aumi, T.: AirAuth: evaluating in-air hand gestures for authentication. In: MobileHCI 2014 (2014)
2. Bashir, M., Scharfenberg, G., Kempf, J.: Person authentication by handwriting in air using a biometric smart pen device. In: BIOSIG 2011, pp. 219–226 (2011)
3. Vikram, S., Li, L., Russell, S.: Handwriting and gestures in the air, recognizing on the fly. In: CHI 2013 Extended Abstracts (2013)
4. Leap Motion. <https://www.leapmotion.com/>. Accessed 25 March 2015