IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 12, 2017, accepted April 30, 2017, date of publication May 11, 2017, date of current version June 27, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2703783

Non-Touch Character Input System Based on Hand Tapping Gestures Using Kinect Sensor

JUNGPIL SHIN[1], (Member, IEEE), AND CHEOL MIN KIM [2]
[1]School of Computer Science and Engineering, The University of Aizu, Fukushima, 965-8580, Japan
[2]Department of Computer Education, Jeju National University, Jeju, 63243, Korea

Corresponding author: Jungpil Shin (jpshin@u-aizu.ac.jp)

ABSTRACT There have been a lot of studies on the text input system using the image-based hand gesture recognition. However, hand gesture languages such as sign languages, finger alphabets, and aerial handwriting treated in the previous works have some problems to be commonly used. The aerial handwriting requires much time for writing and recognition. The sign languages and finger alphabets demand quite a knowledge and practice for using it, which results in restricting the number of their users. As a solution to the problems, this paper proposes a new character input system based on hand tapping gestures for Japanese hiragana and English characters that can be used to facilitate human–computer interaction. The hand tapping gestures are motions for tapping keys on aerial virtual keypads by hands, which can be effectively used as a hand alphabet by anyone, including hearing impaired individuals. For hiragana characters, the hand used for tapping a key and the number of stretched fingers of the hand decide the consonant part of characters, and thereby the aerial virtual keypad. The character to be entered is determined by tapping the key on the virtual keypad corresponding to the desired vowel. Because we adopt a key layout similar to the Japanese and English flick keyboard of smart phones, our hand tapping gestures can be easily used by anyone with only a brief description. The users can effectively interact with computers by using our non-touch input system where only the Kinect sensor is used without any keyboard, mouse or body-worn device. We expect that our character input system will open a new channel for human-computer interaction.

INDEX TERMS Non-touch character input, hand gesture, fingertip detection, Kinect sensor.

I. INTRODUCTION

There have been a lot of studies on the text input system using the image-based hand gesture recognition. Some studies aim at non-touch input methods of computer systems and other ones focus on support for the deaf and hard of hearing people. The non-touch image-based input methods do not require keyboards, mouse devices, and body-worn devices, e.g., cyber-gloves but image capture devices such as cameras. The methods can be commonly applied because most of mobile devices have an equipped camera. And they satisfy requirements for hygiene and cleanliness. Although voice recognition supports non-touch input, it has some drawbacks such as being vulnerable to ambient noise, privacy problems related to being overheard, and problems of mispronunciation and a speech disorder of users.

Generally hearing-impaired people use sign languages and finger alphabets. Each sign language has its own finger alphabet to express things such as proper nouns, objects that users do not know the signs for, and new concepts that there are no signs for. According to statistics presented by the Ministry of Health, Labour and Welfare [1], the number of people with hearing and speech disabilities was reduced from about 36 million in 2006 to about 32 million in 2011 in Japan. However, in light of previous data, we consider that such trend is unlikely to last. Against this backdrop, the Tottori, Gunma and Kanagawa Prefectures made a sign language and finger alphabet language in 2015. Moreover, the Prefectures enacted an ordinance to support active communication with hearing-impaired people. However, it is not easy to learn the sign language and finger alphabet. In particular, the users have difficulty in expressing proper nouns with the finger alphabet.

We have investigated the previous research on the methods for the image-based hand gesture recognition. The results are summarized in Table 1. We have found the following problems in them.

1. No little knowledge on hand gesture mapping is required for users of sign languages and finger alphabets. In the case of Japanese, 46 characters of

TABLE 1. **Results of the previous research experiments.**

Reference	Gesture	Camera	Matching	Character	Recognition rate
[2]	Finger alphabet	Kinect	Template matching	Japanese (79 words)	Involve motion (approx. 76%) Not involve motion(approx. 69%) Sum total (approx. 73%)
[3]	Finger alphabet	Depth and image camera of infrared ToF	Template matching	Japanese (75 words)	Sum total (approx. 85%)
[4]	Finger alphabet	Depth camera	Template matching	Japanese (41words)	Sum total (approx. 91%)
[5]	Sign language	Magnetism sensor	Hidden Markov Model (HMM)	Sign language word (81 words)	50.27%
[6]	Sign language	RGB camera、Time-of-Flight camera (ToF camera)	Template matching	Sign language word (12 type)	RGB camera (58.7%) ToF camera (91.3%)
[7]	Sign language	Kinect	Hidden Markov Model (HMM)	Sign language sentence	Approx. 86%
[8]	Handwriting	RGB camera	Template matching	Numeral character (10: 0 to 9)	91.9%
[9]	Handwriting	Wearable video camera	DP matching	Alphanumeric character (36 words)	75.5%
[10]	Handwriting	Pen installed on RGB camera	DP matching	Japanese (46 words)	87.2%
[11]	Handwriting	Kinect	DP matching	Numerical, Alphabetical characters	95.0% and 98.9%, respectively

hiragana have to be included in the fingertip alphabet: five characters can be mapped to simple finger movements, whereas the other 41 ones cannot. Moreover, if the users want to express some special characters such as the voiced sound mark (" : voiced dot), the semi-voiced sound mark (°: semi-voiced dot), the long sound (-), and small characters (ya, yu, yo, wa, a, i, u, e, and o) noticing contracted sounds and the double consonant (tu), they have to memorize how to express them with their hands.

2. A substantial practice is needed for users to be adapted to the hand gesture communication. This restricts hand gesture languages and finger alphabets from being commonly used not only in communication among people including hearing-impaired people but also in human-computer interaction. Some research has included the practice of the sign language and finger alphabet [4], [12], [13]. However, it takes a long time for beginners to be familiar with the language and alphabet.
3. The methods of character handwriting in the air require a considerable time for character handwriting and hand gesture recognition. Some systems have problems related to the writing speed of the user and processing time for recognizing characters handwritten in the air [8]–[11].

As a solution to these problems, this paper proposes a new character input system based on hand tapping gestures for Japanese hiragana and English characters that can be used to facilitate human-computer interaction. The hand tapping gestures are motions for tapping a key on aerial virtual keypads by hands, which can be effectively used as a hand alphabet by anyone including hearing impaired individuals. The hand alphabet is an alphabet whose letters are represented by the hands. The key point of the hand tapping gestures is to tap an aerial virtual key by a hand with some fingers stretched. We call the hand used to tap the key P(rimary)-hand and the other hand S(econdary)-hand. If the user raises P-hand around his/her shoulder, an aerial virtual keypad associated with the number of stretched fingers of P-hand is conceptually spread in front of him/her by the system. When the user taps a key on the keypad by P-hand, the system recognizes the input character according to the tap position. The layout of each virtual keypad is similar to a flick keyboard of smart phones. The system supports character input for character sets such as Japanese hiragana and the English alphabet.

The flick keyboard has been adopted primarily in touch screens of smart phones. Households that own at least one smart phone now stand at more than 60% in the world; therefore, this input method is ubiquitous in the world. In addition, [14] reports that the proportion of smart phone users will increase. The number of keys of a flick keyboard is much

FIGURE 1. Kinect Sensor.

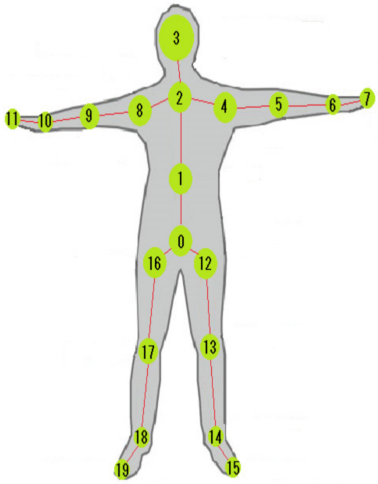


FIGURE 2. Skeleton recognition by Kinect.

smaller than that of a traditional keyboard because each key of a flick keyboard has a popup keypad spread when the key is touched. So it is possible to increase the size of each key in the keyboard and to reduce touches to incorrect keys. In addition, it is easy for users to adapt to the character input system without a substantial knowledge and practice if the keys are well grouped into keypads according to relations among the corresponding characters. In our system, the character group is decided by the hand used in tapping and the number of stretched fingers of the hand and the target character is selected from the group by the tap position.

Our system enables users to input characters in the air without touching any keyboard/mouse and wearing any body-worn computer. This allows users to effectively input characters without onerousness to avoid causing hygiene or cleanliness problems. We use a Kinect sensor for non-touch character input. The Kinect for Windows is a controller for the home video game machine Xbox 360 from the Microsoft Corporation. ‘‘Kinect for Windows SDK’’ was released for a development tool. Fig. 1 shows the Kinect for Windows v1 used in our experiments. The most distinctive part of the Kinect is that it can measure the movements of the skeleton in real time by estimating and detecting each part of the human body. Fig. 2 shows a skeleton recognized by the Kinect.

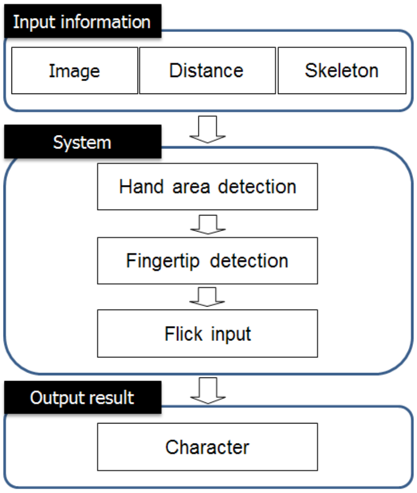


FIGURE 3. Overall system of character input.

In the proposed character input system, the combinations of the number of stretched fingers of P-hand, the tap position of P-hand and the position of S-hand are mapped to characters. This character input method has not hitherto been studied. The Kinect provides the information on the real time movement of the human body's skeleton, from which our system detects the number of stretched fingers of P-hand and the positions of both hands when the user taps a virtual key in the air.

Fig. 3 shows an overview of our system. As input data, the Kinect provides the system with user image obtained from the RGB camera of the Kinect, distance from the Kinect to the user, and skeleton data, i.e., the joint positions of the user. From the data, the system extracts the region of the user's P-hand raised around his/her shoulder and detects stretched fingers of P-hand. After the keypad associated with the number of stretched fingers of P-hand is virtually spread in the air, the key at the position where the hand tapping occurs determines the input character.

The remainder of this paper is organized as follows. Section II describes the motivation and approach of our research. Section III describes how to detect stretched fingers in detail and Section IV explains how characters are entered with the aerial virtual keypads. The experimental results are presented in Section V. After evaluating and discussing the results in Section VI, we conclude in Section VII.

II. MOTIVATION AND APPROACH

Non-touch character input systems increase the possibility of human-computer interaction. Especially, they help users to interact with computers in the situations where they have difficulty in touching input devices. The users can input some characters by using those systems even when their hands are wet or dirty and when they have to manage their hands in a sanitary way. There are various studies on the non-touch text input method, which mostly are based on the voice recognition, the hand gesture recognition, and wearable input devices.

The finger alphabet that is generally a part of sign languages used in the deaf and hearing-impaired community is widely used as a hand gesture based character input method. However, it is not in broad use because it is difficult to learn and practice. If there is a simple hand gesture based character expression method, as an alternative of such finger alphabets, that is easy to learn and practice, more people can use it as an effective non-touch character input method for computing devices. Moreover, it can be used as a simple means of communications between hearing-impaired persons and others. The purpose of this research is to develop such a method.

There are two main goals that we have considered in developing the method. One is the simplicity of the hand gestures and the other is the accuracy and the efficiency of the hand gesture recognition. For simplicity, we have thought out the hand tapping gesture for the keys on an aerial conceptual keyboard instead of mapping a unique hand posture to each character. The keyboard has the layout similar to that of the flick keyboard with pop-up keypads that is familiar to most of Japanese smart phone users. We have focused primarily the hand tapping gestures for Japanese hiragana characters. Each hiragana character can be fundamentally determined with one consonant and one vowel. We have devised the hand tapping gestures in which the consonant factor of hiragana character is determined by the combination of the hand used for the key tapping and the number of stretched fingers of that hand and the vowel factor by the hand tapping position. Japanese users can easily learn our hand tapping gesture based hand alphabet and express hiragana characters with it.

For accuracy and efficiency, we have decided to use the capabilities of the Kinect sensor for recognizing our hand alphabet. Our system should perform the following tasks to recognize the hand tapping gestures.

1. The system must catch the movement direction or the current location of hands in real time to know which key is tapped or which function is requested.
2. The system has to filter out the pixels of overlapping objects behind the hand to find the correct outline of the hand.
3. The system should detect and count all the stretched fingers of the hand used for tapping a key.

The Kinect provides the locations of the joints representing body parts such as hands, shoulders, and head through skeletal tracking in real time. In addition, the system can get the depth values of the pixels from the Kinect. The data obtained from the Kinect enables the system to perform the above tasks more accurately and efficiently.

III. FINGERTIP DETECTION

A. HAND AREA DETECTION

The Kinect recognizes a user at 0.8m ∼ 4.0m distance. Our system obtains the skeleton data and images of the user in real time from the Kinect. In the skeleton data, there are coordinates of 20 joint positions of the user's body shown in Fig. 2. The system uses the coordinates of head, both shoulders, and both hands among them, which respectively correspond to the position 3, 4, 8, 7, and 11 in Fig. 2. The skeleton data is sufficient to know the position of both hands relative to the head and both shoulders of the user. However the system cannot obtain the number of stretched fingers of P-hand directly from the data. Since we needed to count stretched fingers of P-hand, we developed a method that recognizes effectively how many fingers are stretched by detecting fingertips from the user image provided by the Kinect. This is reasonable because each stretched finger has a distinguishable contour line from which fingertip can be easily detected.

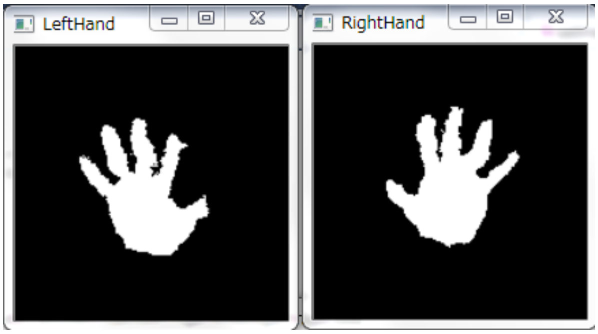


FIGURE 4. Mask images of hand areas of both hands.

To reduce the cost of fingertip detection, it needs to reduce the actual detection area in the real time image of user. It is the reason why we detect fingertips only in the region called hand area. The hand area of P-hand is a small region of the user image where all the fingertips of P-hand can be found. When locating the hand area, our system preliminarily form a square on the xy-plane centered at the three dimensional coordinates of the center position of P-hand. With referring to [19], we set the length of the square's side to 36cm, because the average size of Japanese hands is less than 20cm. The system then extracts only pixels whose xy-coordinates are in the square region. Finally, the system detects the hand area by filtering out the pixels whose z-coordinate is more than 10cm far from the z-coordinate of the center position of P-hand, which prevents the false detection of background things such as the face and the chest of the user. Mask images of hand areas of the right and left hands are shown in Fig. 4.

B. FINGERTIP DETECTION ALGORITHM

Since a fingertip corresponds to a characteristic steep curvature in the contour line of the finger, we can detect the fingertips of P-hand by discriminating such changes in the curvature of the contour line of P-hand [20]. The steps to detect fingertips from the mask image of P-hand area are as follows:

1. Detect the outline of P-hand. To do that, the system uses the function FindContours of OpenCV that features the Suzuki85 algorithm. The function can extract the contours from a binary image [15]. When the mask image of the hand area in Fig. 5 (a) is given, the detected outline of the hand is illustrated in Fig. 5 (b). The point in the hand palm is the center of the hand.

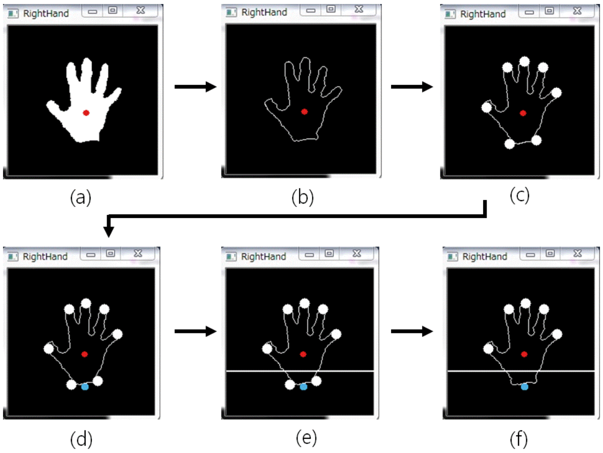


FIGURE 5. A process of fingertip detection.

2. Find all fingertip candidates. The fingertip candidate is the farthest point from the center of the hand among the points on each ‘long enough’ convex part of the outline of P-hand. For the purpose, we calculate the distance from each point on the outline to the centre of P-hand and compare it with the distances of the closest consecutive 40 points on the outline. The point whose distance is greater than those of the 40 points is selected as a fingertip candidate. For the outline of the hand in Fig. 5 (b), the fingertip candidates to be found are shown in Fig. 5 (c)
3. Select real fingertips from the fingertip candidates. It can be done by filtering out the points near the wrist from the fingertip candidates. The fingertips to be finally selected are shown in Fig. 5 (f).

If the distance of each point on the outline is compared only with those of its too close and too few points, the farthest point among the points on a too-small convex part can be chosen as a fingertip candidate. To prevent such points from being selected, we compare the distance of a point with the distances of all the consecutive points whose two end points are sufficiently far from each other. The number of the points that are compared is determined through the experimental analysis. Moreover, for noise removal, we added the proviso that the positions of the fingertips are not so far from or close to the center of P-hand.

In spite of the above efforts, the incorrect points around the wrist can be detected as the fingertip candidates as shown in Fig. 5 (c). We remove the incorrect points according to the following steps. First, we set a quasi wrist point based on the coordinates of the center of the hand as shown Fig. 5 (d). Then, we find the mid-point Z between the quasi wrist and the center of P-hand. We draw a straight line passing Z that is perpendicular to the line from the quasi wrist to the center of P-hand as shown in Fig. 5 (e). Finally, our system selects the points in the opposite side of the quasi wrist as the real fingertips of P-hand. The finally detected fingertips are shown in Fig. 5 (f). Fig. 6 shows the detected fingertips in a real time image.

FIGURE 6. Detected fingertips in a real time image.

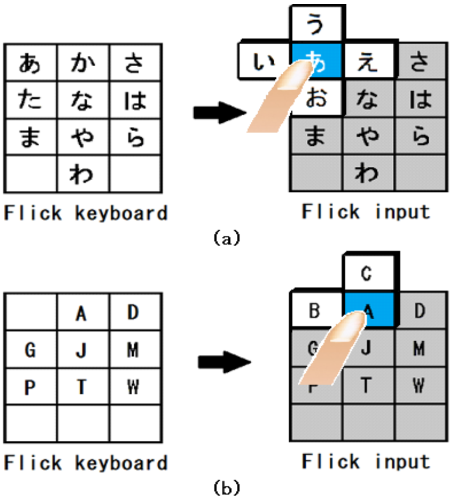


FIGURE 7. The flick keyboards of smart phones (a) Japanese (b) English.

IV. CHARACTER INPUT SYSTEM

A. FLICK INPUT ON TOUCH SCREEN

Smart phones and Tablet PCs have a mounted touch screen as an input device. Software keyboards are used to input characters on the touch screens. Many character input systems of smart phones and Tablet PCs draw software keyboards with the shape of a physical keyboard of PCs or feature phones on the touch screen. The user enters characters by manipulating the virtual keys that are drawn on the touch screen. Since the flick keyboards as software keyboards have a small number of keys, their key arrangement are similar to the numerical keypad that has been mounted on feature phones. If we enter a hiragana character in the flick keyboard as shown in Fig. 7 (a), the consonant is selected by the tap position and the vowel by the flick direction. The flick keyboard is widely used, because it reduces the possibility to tap on an incorrect key even though there are a lot of characters in the character set to be entered. Fig. 7 shows typical examples of the flick keyboard layouts for Japanese and English on smart phones.

B. VIRTUAL KEYPAD INPUT IN THE AIR

Our system provides the user with aerial virtual keypads similar to popup keypads of flick keyboards for Japanese and English. Conceptually, the keypad is placed at the clear space in front of the user as shown in Fig. 8. The user can manipulate the virtual keypads in the air with his/her hands and fingers. The system supports character input in the air using the user's real time skeleton data obtained from the Kinect. Among the skeleton data, the system mainly uses the coordinates of five joint positions, viz. head, both shoulders, and both hands. The system calculates the distances among the five positions and sets a proper size and position of each key of virtual keypads based on the distances.

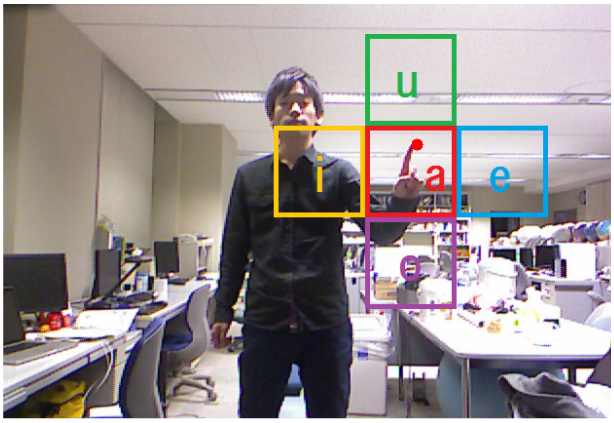


FIGURE 8. An aerial virtual keypad for character input.

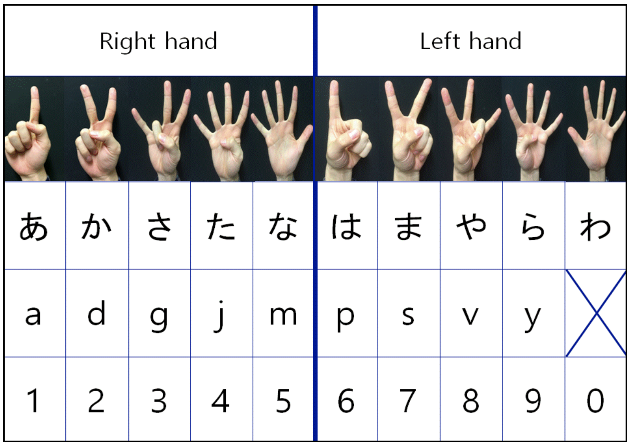
When the user wants to use a virtual keypad, he/she has to raise P-hand around his/her same side shoulder with some fingers stretched. And then the system places a virtual keypad at the calculated position for character input as shown in Fig. 8. The virtual keypad spread in the air is determined according to the number of stretched fingers of P-hand. Fig. 9 shows the characters at the center of virtual Keypads spread according to the number of P-hand's stretched fingers.

The consonant factor of the hiragana character to be entered is determined by which hand is being used as P-hand and the number of the stretched fingers of P-hand. After the virtual keypad for the decided consonant factor is spread, the user can select a vowel factor for the hiragana character by positioning P-hand on a virtual key and enter the corresponding character by tapping the key with P-hand.

V. EXPERIMENTS

A. EXPERIMENTAL ENVIRONMENT

As shown in Fig. 10, the position of the Kinect was arranged at the height of the user's P-hand to perform more precise fingertip detection when the user protruded P-hand to input characters. The pattern of infrared that is illuminated by the infrared projector of the Kinect is distorted by the object and spreads wider with distance. The Kinect detects the object from the statistical analysis of the infrared pattern. However the Kinect cannot detect small parts such as fingertips if the distance is too far. Therefore, we made the examinee keep his/her P-hand 1.0–2.0m far from the Kinect in the experiments.



Right hand					Left hand				
あ	か	さ	た	な	は	ま	や	ら	わ
a	d	g	j	m	p	s	v	y	
1	2	3	4	5	6	7	8	9	0

FIGURE 9. The characters at the center of virtual keypads spread according to the number of P-hand's stretched fingers.

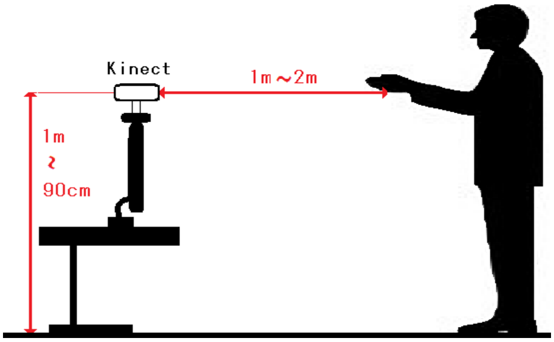


FIGURE 10. The position of the Kinect.

TABLE 2. Hand gesture mapping for characters.

Character		Hand Gesture
Japanese	Characters with the voiced sound mark	Tap an aerial virtual key with P-hand in the state of raising S-hand closely to S-hand side shoulder
	Characters with the semi-voiced sound mark	Tap an aerial virtual key with P-hand in the state of raising S-hand far sideways from S-hand side shoulder
	Small letters	Tap an aerial virtual key with P-hand in the state of raising S-hand closely to the head
English Alphabet	Small letters	Tap an aerial virtual key with P-hand
	Capital letters	Tap an aerial virtual key with P-hand in the state of raising S-hand closely to S-hand side shoulder

Our system uses the virtual keypads similar to popup keypads shown in Fig. 7 for hiragana and the English alphabet. Table 2 shows how the user inputs hiragana characters with the voiced sound mark or the semi-voiced sound mark and how the user input small version characters of hiragana. It also describes how small and capital letters of the English alphabet can be entered. Table 3 shows hand gesture mappings for a few function. The user can delete the entered characters by moving the right hand from right to left. A space character can be provided as an input by moving the left hand from left

TABLE 3. **Hand gesture mapping for functions.**

Function	Hand Gesture
Delete	Move the right hand from right to left
Space	Move the left hand from left to right
Character set switch	Cross both hands in front of chest

ようこそ　にほんへ
アルファベット入力に切り替えました。
Welcome to JAPAN

FIGURE 11. **Simulation of text input with aerial virtual keypads.**

TABLE 4. **Character recognition accuracies when using right hand as P-hand.**

Virtual key position	The number of stretched fingers					Average
	1	2	3	4	5	
Center	100%	98.3%	98.3%	91.7%	95.0%	96.7%
Left	100%	98.3%	96.7%	93.3%	93.3%	96.3%
Top	100%	100%	95.0%	81.7%	93.3%	94.0%
Right	100%	93.3%	86.7%	90.0%	83.3%	90.7%
Bottom	100%	90.0%	90.0%	98.3%	90.0%	93.7%
Average	100%	96.0%	93.4%	91.0%	91.0%	94.3%

to right. The user can change the character set by crossing both hands. Fig. 11 shows a situation where a user provides our system with the input text "Welcome to JAPAN" using the aerial virtual keypads for the English alphabet.

B. EXPERIMENTAL RESULTS

We performed the experiments to measure the accuracies of detecting stretched fingers of P-hand and recognizing characters entered with the aerial virtual keypads. The examinees entered all the hiragana characters with changing P-hand, the number of the stretched fingers, and tap position. The examinees entered each character repeatedly once per second for one minute.

Table 4 and Table 5 show the results measured while each examinee used his/her right and left hand as P-hand respectively.

Fig. 12 shows the accuracies in the rows labeled 'Center' of Table 4 and Table 5. Those are corresponding to fingertip detection accuracies since they were measured when the examinees tapped the virtual keys in the center of the aerial virtual keypads. The tap position is a location where the palm of P-hand naturally faces the Kinect when the examinee raises his/her P-hand. Fig. 13 shows the measured results from the viewpoint of the number of stretched fingers.

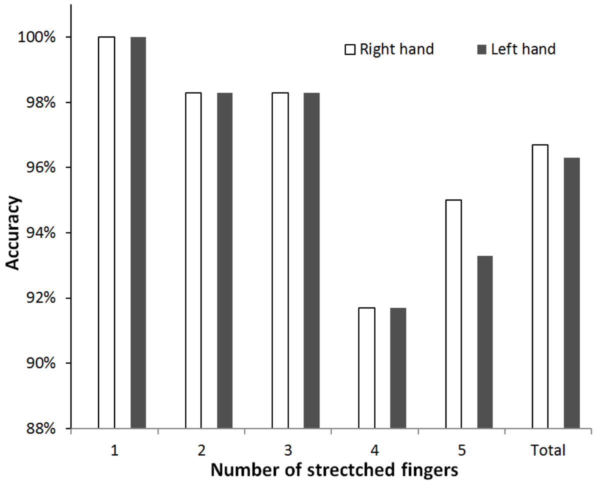


FIGURE 12. **Fingertip detection accuracies of right and left hand.**

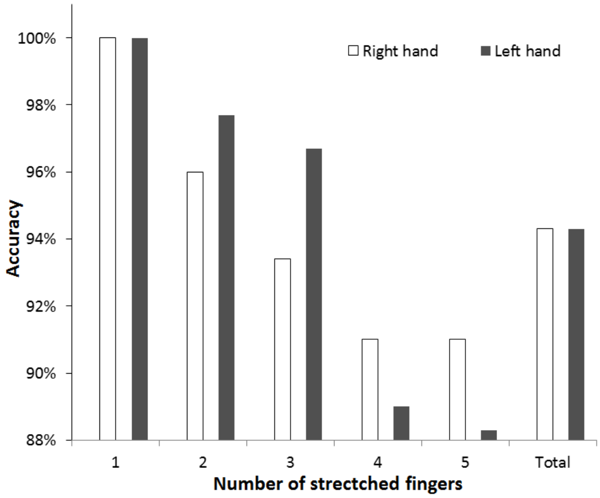


FIGURE 13. **Character recognition accuracies according to the number of stretched fingers.**

TABLE 5. **Character recognition accuracies when using left hand as P-hand.**

Virtual key position	The number of stretched fingers					Average
	1	2	3	4	5	
Center	100%	98.3%	98.3%	91.7%	93.3%	96.3%
Left	100%	95%	91.7%	88.3%	86.7%	92.3%
Top	100%	96.7%	98.3%	100%	85.0%	96.0%
Right	100%	100%	98.3%	80.0%	85.0%	92.7%
Bottom	100%	98.3%	96.7%	85.0%	91.7%	94.3%
Average	100%	97.7%	96.7%	89.0%	88.3%	94.3%

TABLE 6. Accuracy comparison with other related works.

Reference	Functions	Recognition accuracy	Accuracy in our research
[16] [17]	Fingertip Detection	1 finger (99.2%) 2 fingers (97.5%)	1 finger (100%) 2 fingers (98.3%)
[18]	Fingertip detection / Character input	Character input 66%	Character input 94.3%

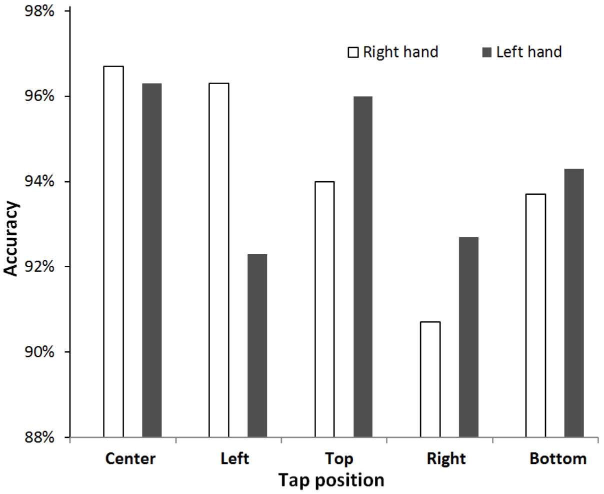


FIGURE 14. Character recognition accuracies according to tap positions.

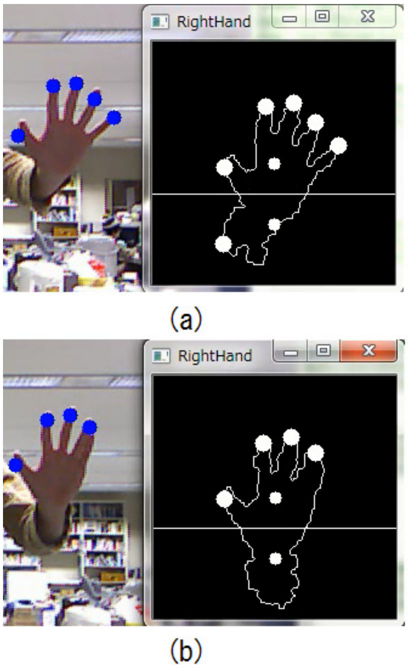


FIGURE 15. The effect of the space between stretched fingers to fingertip detection (a) wide inter-finger space (b) narrow inter-finger space.

Fig. 14 shows the same results from the viewpoint of the tap position for the aerial virtual keypads.

VI. DISCUSSIONS

The rows 'Center' of Table 4 and Table 5 show that the average detection accuracy of stretched fingers of the right hand is 96.7%; that of the left hand is 96.3% when the examinees easily raise their P-hands so that the palm of P-hand faces the Kinect. From the results shown in the tables, we can find out that the overall average of character recognition accuracies for both hands is 94.3%. Those are meaningful improvements in the accuracy of fingertip detection and character recognition when compared with other works shown in Table 6.

The detection accuracy becomes lower as the number of fingers increases as shown in Fig. 13. The lowest detection accuracies when using the right and left hand as P-hand are 91.0% and 88.3% respectively. Moreover, the accuracy is lowest when the tap position is far from the body of the user as shown in Fig. 14. When using the right hand as P-hand, the tap position where the detection accuracy is worst, i.e., 90.7%, is the right key of the virtual keypad spread at the right side of the human body. For the left hand, the lowest accuracy 92.3% is found in the case of tapping the left key of the virtual keypads spread at the left side of the human body.

False detections are mainly from the following two causes. The first cause is that when the number of the stretched fingers increase, each space between the stretched fingers becomes narrow and two or more fingers may agglomerate together to form a larger finger in the mask image. We detect the fingertips by finding the farthest point from the center of the hand among the points on each 'long enough' convex part of the outline of P-hand with some fingers stretched. However, the Kinect cannot provide the precise outline so that the system can clearly distinguish between stretched fingers close to each other. Fig. 15 shows the two mask images when the spaces between stretched fingers are wide and narrow respectively. We can see a mask image where two close fingers are merged together in Fig. 15 (b).

The second cause is that when the palm of P-hand does not face the Kinect frontally, the system may recognize the overlapped two or more fingers as one finger and a long finger as a short finger, which results in miscounting the number of stretched fingers. Such situations are occasionally occurred at hand tapping because a tapping motion involves quite a change in the angle of P-hand. Fig. 16 (b) illustrates a missing finger in counting due to the cause.

There are solutions to the above problems such as changing the Kinect from v1 to v2, using the Multi-Kinect, and detecting fingertips from an image captured just before or after key tapping. Because the Kinect v2 has better performance in resolution and distance than the Kinect v1, the possibility of finger miscounting caused by agglomerating close stretched fingers will be reduced by using the Kinect v2. Since the Multi-Kinect provides images of fingers from different viewpoints, we will be able to reduce the possibility of finger miscounting caused by finger overlapping with them. When using the image captured just before or after key tapping, we can reduce the number of false detection occurred from the second cause stated before.

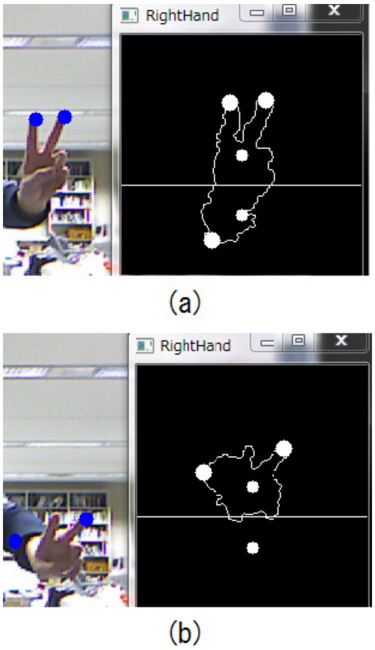


FIGURE 16. **The effect of direction of the palm of the hand to fingertip detection (a) the palm of the hand facing frontally to the Kinect (b) the palm facing another direction.**

The proposed input system solves the following problems described in Section 1.

1. No little knowledge on hand gesture mapping is required for users of sign languages and finger alphabets.
2. A substantial practice is needed for users to be adapted to the hand gesture communication.
3. The methods of character handwriting in the air require a considerable time for character handwriting and hand gesture recognition.

For problem 1 and problem 2, our system is easy to learn and practice because it is based on simple hand gestures for tapping keys on aerial virtual keypads similar to popup keypads of the Japanese and English flick keyboard. The users only need to memorize the layout of the aerial virtual keypads associated with the order of English characters and the order of consonants and vowels of hiragana characters and to practice hand tapping gestures for entering the desired character. For problem 3, the users can save time in aerial handwriting since our system enables them to efficiently input each character with a simple hand tapping gesture by recognizing which key is entered from the Kinect data.

VII. CONCLUSIONS

This paper proposes a new character input system based on hand tapping gestures for Japanese hiragana characters that can be used to facilitate human-computer interaction. The hand tapping gestures are motions for tapping a key on aerial virtual keypads by hands. Each aerial keypad consists of hiragana character keys with the same consonant. The user can use a virtual keypad for character input by raising a hand with some stretched fingers around his/her shoulder. The virtual keypad to be used is decided by which hand the user raises for tapping a key and how many fingers of the hand are stretched. The user can enter the target character by tapping the key on the virtual keypad corresponding to the desired vowel.

Our input system enables the user to input characters without touches on any device by recognizing the posture and movement of his/her hands from the real-time images provided by the Kinect. The system knows the number of stretched fingers of the raised hand by detecting its fingertips from the image and counting them. The fingertip detection accuracies for the right and left hand were 97.0% and 96.3% respectively. The average recognition accuracy of finger alphabet was 94.3%. The proposed finger alphabet and character input system solve the problems such as quite a required knowledge and practice, long time for character input, and practicality of the system. We will do additional research on the method to solve problems occurred from close or overlapped fingers.

REFERENCES

[1] Ministry of Health, Labour and Welfare, "The survey of difficulties of daily life (survey of nationwide home children with disabilities, and others, 2011)," (in Japanese), Dept. Health Welfare Persons Disabilities, Tokyo, Japan, Tech. Rep., 2013.

[2] S. Tanaka, M. Takuma, and Y. Tsukada, "Recognition of finger alphabet remote image sensor," (in Japanese), in *Proc. 76th Nat. Conv. (IPSJ)*, 2014, vol. 76. no. 2, pp. 2275–2276.

[3] T. Miyake, D. Wakatuki, and I. Naito, "A basic study on recognizing fingerspelling with hand movements by the use of depth image," (in Japanese), Nat. Univ. Corporation Tsukuba, Univ. Technology, Tsukuba, Japan, Tech. Rep., Dec. 2012, vol. 20, no. 1, pp. 7–13.

[4] D. Takabayashi *et al.*, "Training system for learning finger alphabets with feedback functions," (in Japanese), IEICE (HIP), Tech. Rep., 2013, vol. 112, no. 483, pp. 79–84.

[5] M. Maehatake, M. Nishida, Y. Horiuchi, and A. Ichikawa, "A study on sign language recognition based on gesture components of position and movement," (in Japanese), in *Proc. Workshop Interact. Syst. Softw. (WISS)*, Japan, Dec. 2007, pp. 129–130.

[6] A. Sato, K. Shinoda, and S. Furui, "Sign language recognition using time-of-flight camera," (in Japanese), in *Proc. Meeting Image Recognit. Understand. (MIRU)*, 2010, vol. 3, no. 44, pp. 1861–1868.

[7] Y. Nishimura, D. Imamura, Y. Horiuchi, K. Kawamoto, T. Shinozaki, and S. Kuroiwa, "HMM sign language recognition using kinect and particle filter," (in Japanese), IEICE (PRMU), Tech. Rep., 2012, vol. 111, no. 430, pp. 161–166.

[8] M. Ohkura, R. Manabe, H. Shimada, and Y. Shimada, "A recognition algorithm of numerals written in the air by a finger-tip," (in Japanese), *J. IPSJ*, vol. 52, no. 2, pp. 910–916, 2011.

[9] T. Sonoda and Y. Muraoka, "A letter input system of handwriting gesture," (in Japanese), *IEICE Trans. Commun.*, vol. J86-D-II, no. 7, pp. 1015–1025, 2003.

[10] Y. Fujii, M. Takezawa, H. Sanada, and K. Watanabe, "An aerial handwritten character input system," (in Japanese), IPSJ (MBL), Tech. Rep., 2009, vol. 50, no. 6, pp. 1–4.

[11] T. Murata and J. Shin, "Hand gesture and character recognition based on kinect sensor," *Int. J. Distrib. Sensor Netw.*, vol. 10, Jul. 2014, Art. no. 278460. [Online]. Available: http://dx.doi.org/10.1155/2014/543278460

[12] Y. Tanaka, "Training system for learning finger alphabets with feedback functions," M.S. thesis, Faculty Ind. Technol., Tsukuba Univ. Technol., Tsukuba, Japan, 2014.

[13] N. Suzuki, K. Oohashi, and S. Horiguchi, "Performance evaluation of finger character learning support system using a vibrator with the hand shape input device," (in Japanese), Res. Rep. Jpn. Adv. Inst. Sci. Technol., Ishikawa, Japan, Tech. Rep. IS-RR-99-0017, 1999, pp. 1–21.

[14] "Information and communications in Japan," White Paper, Ministry of Internal Affairs and Communications, 2015.
[15] S. Suzuki and K. Abe, "Topological structural analysis of digital binary image by border following," *J. Comput. Vis. Graph. Image Process.*, vol. 30, no. l, pp. 32–46, 1985.
[16] K. Oka, Y. Sato, and H. Koike, "Real-time fingertip tracking and gesture recognition," *IEEE Comput. Graph. Appl.*, vol. 22, no. 6, pp. 64–71, Nov. 2002.
[17] K. Oka, Y. Sato, and H. Koike, "Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems," (in Japanese), in *Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit. (FGR)*, May 2002, pp. 429–434.
[18] N. Matsui and Y. Yamamoto, "Virtual keyboard: Realization of robust real-time fingertip detection from video-image," (in Japanese), in *Proc. Japan Soc. Softw. Sci. Technol. (JSSST)*, Japan, 2000.
[19] M.Kawauchi. (2012). *Dimension Data of the Hands of the Japanese*. [Online]. Available: https://www.dh.aist.go.jp/database/hand/index.html
[20] K. Nakamura, KINECT for Windows SDK Programming C++, Shuwa System, Japan, 2012 (in Japanese).

JUNGPIL SHIN (M'99) received the Ph.D. degree in communication engineering from Kyushu University, Japan, in 1999. He became an Associate Professor and a Senior Associate Professor with the School of Computer Science and Engineering, 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, real-time system, oriental character processing, mobile computing, computer education, human recognition, and machine intelligence. He is a member of the ACM, the IEICE, the IPSJ, the KISS, and the KIPS. He served several conferences as a Program Chair and a Program Committee Member for numerous international conferences and serves as a Reviewer for several SCI major journals and an Editor of journals.

CHEOL MIN KIM received the Ph.D. degree from Seoul National University, South Korea, in 1996. His major is operating system in computer engineering. He became an Instructor and a Professor with the Department of Computer Education, College of Education, Jeju National University, South Korea, in 1997 and 2010, respectively. He has continuously done research in the field of system software. And also, he has been interested in the computer science education for secondary school students and gifted teenagers. Thus, he has given his lectures for them and has performed his research related to the instructional methods, the instructional contents, and teaching-and-learning systems for them. His research interests include system software, computer education, algorithm visualization, machine intelligence, and computational thinking.

• • •