Early gesture recognition method with an accelerometer

Izuta, Ryo
Murao, Kazuya
Terada, Tsutomu
Tsukamoto, Masahiko

(Citation)
International Journal of Pervasive Computing and Communications,11(3):270-287

(Issue Date)
2015

(Resource Type)
journal article

(Version)
Accepted Manuscript

Early Gesture Recognition Method with an Accelerometer

Ryo Izuta
Graduate School of Engineering,
Kobe University
1-1 Rokkodai-cho, Nada,
Kobe, Hyogo 657-8501, Japan
Izuta.r@stu.kobe-u.ac.jp

Kazuya Murao
College of Information Science and Engineering,
Ritsumeikan University
1-1-1 Nojihigashi, Kusatsu,
Shiga 525-8577, Japan
murao@cs.ritsumei.ac.jp

Tsutomu Terada
Graduate School of Engineering,
Kobe University
PRESTO, Japan Science and
Technology Agency,
1-1 Rokkodai-cho, Nada,
Kobe, Hyogo 657-8501, Japan
tsutomu@eedept.kobe-u.ac.jp

Masahiko Tsukamoto
Graduate School of Engineering,
Kobe University
1-1 Rokkodai-cho, Nada,
Kobe, Hyogo 657-8501, Japan
tuka@kobe-u.ac.jp

ABSTRACT

An accelerometer is installed in most current mobile phones, such as iPhones, Android-powered devices, and video game controllers for the Wii or PS3, which enables easy and intuitive operations. Therefore, many gesture-based user interfaces that use accelerometers are expected to appear in the future. Gesture recognition systems with an accelerometer generally have to construct models with user's gesture data before use, and recognize unknown gestures by comparing them with the models. Since the recognition process generally starts after the gesture has finished, the output of the recognition result and feedback delay, which may cause users to retry gestures and thus degrade the interface usability. We propose a gesture recognition method at an early stage that sequentially calculates the distance between the input and training data, and outputs recognition results only when one output candidate has a stronger likelihood than the others. Gestures are recognized in the early stages of an input gesture without deteriorating the degree of accuracy. Our evaluation results showed that the recognition accuracy for the proposed method approached 1.00 and the recognition results were output 1,000 msec on average before a gesture had finished.

Article Classification

Research paper

Keywords

Early Recognition, Gesture Recognition, Accelerometer

Professional Biography

Ryo Izuta: Student in master course at Kobe University

Kazuya Murao: Assistant professor at Ritsumeikan University

Tsutomu Terada: Associate professor at Kobe University

Masahiko Tsukamoto: Professor at Kobe University

1. INTRODUCTION

Downsizing the computers has led to mobile computing that has recently attracted a great deal of attention. Mobile computing enables users to use small computers and devices anytime and anywhere; however, their usability is lower than that of desktop computing because they have few buttons and a small display. In the contrary, a lot of devices with sensors have recently been released. In particular, an accelerometer is installed in most current mobile phones, such as iPhones, Android-powered devices, and video game controllers for Wii or PS3, which enables easy and intuitive operations such as scrolling browsers and drawing 3D objects by detecting the inclination and motion of the devices.

Gesture recognition methods with accelerometers generally learn a given user's gesture data before using the system, then recognizes any unknown gestures by comparing them with the training data. The recognition process starts after a gesture has finished, and therefore, any interaction or feedback depending on the recognition result is delayed. For example, an image on a smartphone screen rotates a few seconds after the device has been tilted, which may cause the user to retry tilting the smartphone even if the first one was correctly recognized. Although many studies on gesture recognition using accelerometers have been done, as far as we know none of these studies have taken the potential delays in output into consideration. The simplest way to achieve early recognition is to start it at a fixed time after a gesture starts. However, the degree of accuracy would decrease if a gesture in an early stage was similar to the others. Moreover, the timing of a recognition has to be capped by the length of the shortest gesture, which may be too early for longer gestures. On the other hand, retreated recognition timing will exceed the length of the shorter gestures. In addition, a proper length of training data has to be found since the full length of training data does not fit the input data until halfway. In order to recognize gestures in an early stage, proper recognition timing and a proper length of training data have to be decided.

We propose a gesture recognition method used in the early stages that sequentially calculates the distance between the input and training data. The proposed method outputs the recognition result when one candidate has a stronger likelihood of recognition than the other candidates so that similar incorrect gestures are not output. We experimentally evaluated our proposed method on 27 kinds of gestures and confirmed that the recognition process finished 1,000 msec before the end of the gestures on average without deteriorating the level of accuracy. Gestures were recognized in an early stage of motion, which would lead to an improvement in the interface usability and a reduction in the number of incorrect operations such as retried gestures. Finally, we implemented a gesture-based photo viewer as a useful

application of our proposed method and used the proposed early gesture recognition system in a live unscripted performance.

The remainder of the paper is organized as follows. Section 2 describes the related work on gesture recognition using accelerometers. Section 3 proposes an early gesture recognition method. Section 4 evaluates the proposed method and Section 5 introduces its applications. Finally, Section 6 concludes this paper.

2. RELATED WORK

In this section, we introduce about applications using gesture recognition and the early gesture recognition as related works.

2.1 Application using gesture recognition

Tomibayashi et al. proposed a disk jockey (DJ) support system [1]. The system recognizes gestures through the cue of accelerometers on both hands and enables DJs to adjust the volume and start/stop their music using specific gestures, which accentuates the DJ's performance. The system proposed by Liu et al. recognized eight kinds of gestures, such as drawing a line or a circle recommended by the Nokia Research Institute, using a 3-axis accelerometer [2]. They also implemented a gesture-based video clip viewer for a social networking based video-sharing service for mobile devices using their system. For example, a video clip of your friend scrolls according to your own scroll. Ruiz et al. presented DoubleFlip, a unique motion gesture designed as an input of delimiter for mobile motion-based interaction [3]. Based on a collection of 2,100 hours of motion data captured from 99 users, they found that the DoubleFlip recognizer is extremely resistant to false positive conditions, while still achieving a high recognition rate. Agrawal et al. proposed a system that recognizes alphabetic characters written in the air with a cell phone by converting the acceleration data into spatial motion, which achieved an 83% recognition accuracy by adhering to some restrictions [4]. Amma et al. presented a hands-free input method by using 3D handwriting recognition [5]. A user can write text in the air by wearing a glove that contains an accelerometer and a gyroscope. They proposed a two-stage approach: spotting and recognition. The spotting uses a Support Vector Machine and the recognition uses Hidden Markov Models. A sentence recognition experiment for nine subjects was conducted using person-dependent and person-independent setups, resulting in an 11% word error rate for the person-independent setup and a 3% one for the person-independent setup. In this system, the sensor data is sent to another device for analysis. Yatani et al. proposed a communication interface called Toss-It which can intuitively exchange information [6]. Using a personal digital assistant (PDA) containing an accelerometer, the speed, angle, and direction of the user's toss are detected, and the user can send information using intuitive motions such as a toss.

2.2 Early gesture recognition

Some studies on early gesture recognition using image processing have been reported. Mori et al. proposed the early gesture recognition system using video images [7]. They classified some gestures to some patterns by focusing on the gesture motion and discussed whether or not gestures can be recognized in the early stages. They implemented the early gesture recognition method for the gestures that had been classified to ones that can be recognized in an early stage, and achieved early gesture recognition at a high degree of accuracy. They, however, did not mention gestures that were recognizable in an early stage. The timing to output the recognition results should be flexibly changed depending on the gestures for many kinds of gestures. Scharenborg et al. investigated the ability of the speech recognition system called SpeM [8]. SpeM has a recognition system that is based on the combination of an automatic phone recognizer and a word search module in order to determine as early as possible whether a word is likely to be correctly recognized during the word recognition process. They evaluated the recognition accuracy of SpeM using 1,463 polysyllabic words in 885 continuous speech utterances, and presented that the Bayesian activation can be used as a predictor for the online early recognition of polysyllabic words. SpeM recognizes only polysyllabic words, and it is difficult to use this the system in recognition using accelerometers. Some studies related to early gesture recognition using accelerometers have been reported. Kanke et al. proposed a system for drums called Airstick Drum [9]. This system recognizes beating a drum or striking at the air with a drum stick with an accelerometer and gyro. Since the delayed sound of drum is not allowed in the drum play, the system has to finish judging if the drum has been beaten or the air has been struck. They focused on the characteristic waves in the sensor values and set the threshold to classify them. The threshold, however, is tuned for the movements. It would be difficult to set the thresholds for many kinds and more complicated gestures. The system proposed by Fujimoto et al. recognized dance steps and output sound in response to the steps [10]. The main advantages of this study are that the timing of the sound output was adjusted to the sound beats, and an incorrect sound is modified within a couple of beats so that the modification is not noticed by the audience. It is practicable to modify the output sound for dance; however, it is not easy to modify or cancel commands on a gesture-based interface once they are executed.

There is a trade off between the earliness of a recognition and the accuracy of the recognition. It is important to decide the timing to output a result with a higher degree of accuracy. Murao et al. reported that the recognition accuracies have been improved by removing gestures that are similar to the others from the candidates, which means the appropriate timing for a recognition may depend on the kinds and number of gestures [11].

We propose an early gesture recognition method that successively calculates the distances between the input sequence and training data, and compares the best candidate with the second best one. Finally, the recognition result is output when the distances to the nearest one and the second nearest one differ by more than a certain threshold. Our proposed method does not require full length of gestures and is able to control the trade off of the earliness and accuracy.

3. METHOD FOR EARLY RECOGNITION

In this section, we explain about the early gesture recognition method as the proposed method.

3.1 System Structure

Gesture recognition systems recognize gestures by comparing the unknown gesture data with the training data captured beforehand. The input and training data are supposed to be full length. We propose a system for the early recognition of gestures as shown in Figure 1. Our system recognizes gestures using the input data at an early stage of the motion. The proposed system involves two steps of calculating the distance and the relative score. In the first step, the system calculates the distances between the input data and all the training data, and finds the training data with the shortest and the second shortest distances. In the second step, the

system calculates the relative score between the best matching gestures by comparing the shortest and the second shortest distances. When both distances are almost the same, the system does not output the result but waits for the next input. We will explain the algorithm of calculating the distance and relative score in the follows.

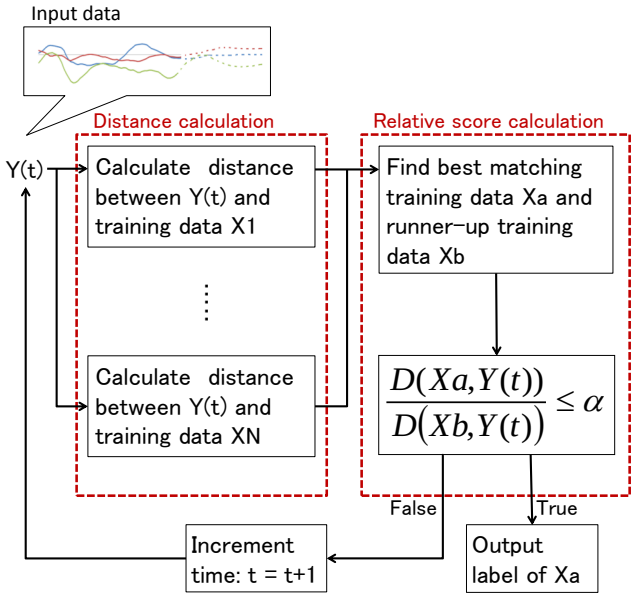

Figure 1. System structure.

3.2 Distance calculation algorithm

Time-series data are widely used in various fields such as science, medicine, economics, and engineering. Calculation of the similarity between a time-series is required in order to do datamining. Although the simple approach to measuring the similarity is to use the Euclidean distance, the results are susceptible to temporal distortion, and the number of samples in two data sequences must be equal.

Dynamic Time Warping (DTW) [12] is an algorithm to measure the similarity between two time-series data, which redeems the drawbacks of the Euclidean distance. As the features of DTW, it calculates the temporal nonlinear elastic distance, the similarity between two sequences, which may vary in time or speed, can be measured, and the number of both samples need not to be equal. For example, comparing two kinds of data for drawing a circle in the air whose rotating speeds are different, DTW can find their similarities. In addition, in case a part of each data differs, DTW is applicable because of its non-linear elasticity.

However, as the original DTW algorithm assumes that the input and training data are full lengths of the gestures, we improve the algorithm to calculate the distance using the gesture data until the early stage of the motion. The algorithm works as follows. For the sake of simplicity, we assume there is data for one axis.

Supposing the length of an input sequence is t, our algorithm confines the training data length from $t-\varepsilon$ to $t+\varepsilon$, where ε is the searching width, provided that the distance is calculated using the full length of the training data. When training data $X = (x_1, \cdots x_{t+\varepsilon})$ and input data $Y = (y_1, \cdots y_t)$ with length $t+\varepsilon$ and t, are compared, an $(t+\varepsilon) \times t$ matrix d is defined as $d(i,j) = \sqrt{(x_i - y_j)^2}$. Next, warping path $F = (f_1, \cdots f_k)$, which is the path of the pairs of indices of X and Y, is found. F meets three conditions.

- Boundary:
  $w_1 = (1,1), w_k = (m,n)$
- Seriality:
  $w_k = (a,b), w_{k-1} = (a',b') \Rightarrow a - a' \leq 1 \wedge b - b' \leq 1$
- Monotony:
  $w_k = (a,b), w_{k-1} = (a',b') \Rightarrow a - a' \leq 0 \wedge b - b' \leq 0$

The following steps are used to find the path with the lowest cost that satisfies these conditions.

Initialization:

$$f(0,0) = 0$$
$$f(i,0) = \infty \text{ for } i = 1, \cdots t+\varepsilon$$
$$f(0,j) = \infty \text{ for } j = 1, \cdots, t$$

Do for $i = 1,2, \cdots, t+\varepsilon$

Do for $j = 1,2, \cdots, t$

$$f(i,j) = d(x_i, y_j) + \min \begin{cases} f(i-1,j-1) \\ f(i-1,j) \\ f(i,j-1) \end{cases}$$

Output:

Return $D(X, Y(t)) =$

$$\min \begin{cases} f(t-\varepsilon,t)/(t-\varepsilon+t) \\ f(t-\varepsilon+1,t)/(t-\varepsilon+1+t) \\ \vdots \\ f(t+\varepsilon-1,t)/f(t+\varepsilon-1+t) \\ f(t+\varepsilon,t)/(t+\varepsilon+t) \end{cases}$$

The obtained cost D(X,Y(t)) is the distance between X and Y. The returned D(X, Y(t)) is divided by the sum of the length of the input and training data since the DTW distance increases with the length of the sequences. The values from $f(t,t-\varepsilon)$ to $f(t,t+\varepsilon)$ represent the distances between the input data until time t and the training data until $t-\varepsilon$ to $t+\varepsilon$ as shown in Figure 2. The shaded path is the shortest one and $f(t,t-1)$ is the shortest distance, resulting in the distance between X and Y at time t. The main reason for taking the margin of $t-\varepsilon \leq j \leq t+\varepsilon$ is that the speed of the input and training data may differ.

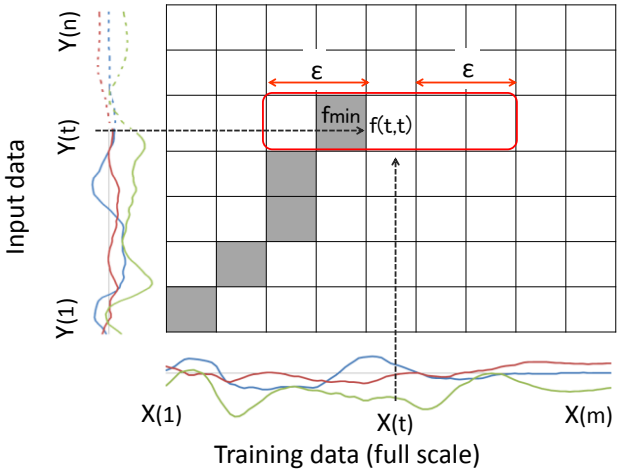

Figure 2. Distance calculation algorithm.

The distances for all the training data are derived from the distance calculation above and we denote the gestures with the shortest and the second shortest distances as g1 and g2. In other words, $D(X_{g1}, Y(t))$ is the shortest distance and $D(X_{g2}, Y(t))$ is the second shortest distance. The distance score is calculated every time a data is input.

Rakthanmanon et al. discussed the optimization of a sequential search under DTW [13]. A classic trick to speed up the sequential search using an expensive distance measure such as DTW is to use a lower bound to prone off the unpromising candidates. They incrementally calculate the partial DTW distance using LB_{Keogh} lower bound [14] instead of calculating the full DTW. This idea, however, only reduces the computational complexity and does not recognize gestures in the early stage.

An example of a calculation when using our proposed algorithm using the matrix in Figure 3 is explained as follows. In this example, the lower bound for the input data length is not considered. Given training data $X = \{5,12,6,10,6,5,18,20,10,7\}$ and input data $Y(t) = \{11,6,9,4,2,\cdots\}$, the DTW distance is calculated as outlined in Figure 3. First, when Y(1)=11 is given, the first row of the matrix is calculated. Setting $\varepsilon = 1$, f(1,1) and f(2,1) as the candidates for the distance and the shortest distance of all, f(1,1)=6 is the distance between X and Y(t) at time $t = 1$. Moreover, when $Y(2) = 6$ is given, the second row of the matrix is also calculated. Then, f(1,2), f(2,2), and f(3,2) are the candidates and $f(3,2) = 7$, the shortest distance of the three, becomes the distance at time $t = 2$. If there are some shortest distances such as $f(1,2) = f(3,2)$. The shortest distance of larger index for the training data is selected. The matrix is calculated every time input data arrives.

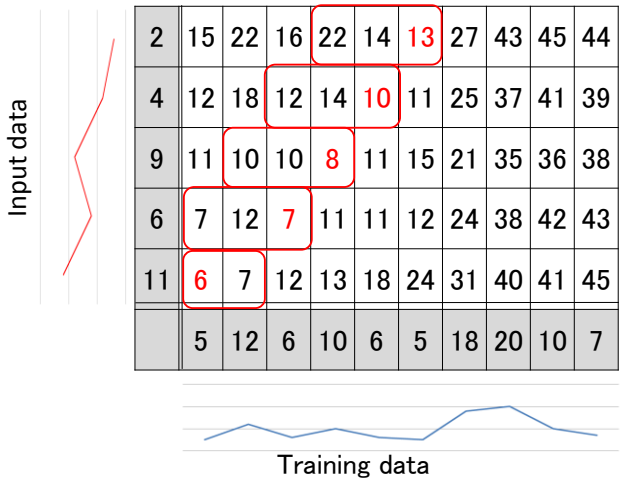

Figure 3. Detailed example of distance calculation.

3.3 Relative score calculation algorithm

The distances between input data Y(t) and all the training data, X, are obtained using the distance calculation algorithm. This step calculates the relative score of the shortest distance to the second shortest distance using the following equation.

$$\text{RelativeScore(t)} = \frac{D(X_{g1}, Y(t))}{D(X_{g2}, Y(t))}$$

The reason for using the relative score is that if the input data are similar to that for multiple gestures in an early stage of motion, all these distances are short and the training data with the shortest distance may not be correct. At time t, recognition process works as follows.

$$\text{if RelativeScore} < \alpha \Rightarrow \text{Output g1}$$
$$\text{othewise} \Rightarrow \text{wait for the data of } t+1.$$

If the RelativeScore is smaller than α, g1 is output as the recognition result; otherwise, the system does not output a recognition result, updates the input data to $Y(t+1)$, and calculates the distance. The α $(0 < \alpha < 1)$ is a parameter that confines the output unless the shortest distance is far from the second shortest distance. When multiple training examples are registered for one gesture label, the gestures with the shortest and the second shortest distances may have the same gesture labels. In this case, the second shortest distance will be selected from the samples with different gesture labels to that of the shortest distance. Our proposed method also sets a lower bound for the input data. The input data are not processed until the length of the input reaches ten samples (0.2 sec). When the relative score is not lower than α even when the gesture ends, the leading candidate at that time is output.

4. EVALUATION

In this section, we evaluate the early gesture recognition system with the proposed method explained in the previous section.

4.1 Environment

Data on 27 kinds of gestures listed in Table 1 that we assumed would be performed when using a mobile tablet such as an iPad were captured using an accelerometer eight times for each gesture, one of which was used as the training data and the remainders were used as the testing data. The sensor used in the evaluation was a WAA-006, made by Wireless Technologies, Inc.[15], which has a wireless 3-axis accelerometer. The sample frequency was 50 Hz. The instructions were not given verbally to reduce the error due to individual interpretation. Instead, one of the authors demonstrated the actual movement for each gesture. The average motion time for the 27 kinds of gestures was 1.40 sec, while the longest motion time was 2.34 sec, and the shortest motion time was 0.66 sec. The data were transmitted to a laptop (Let's note, CF-SX-1, made by Panasonic Inc.) from the sensor via Bluetooth.

The recognition accuracies and the remaining time to the end of the gestures were calculated by changing ε from 0 to 20 with 10 intervals. Next, the recognition accuracies and the remaining time to the end of the gestures were calculated using a comparison method and the proposed one. The comparison method is described below.

4.2 Comparison method

This section introduces two comparison methods in order to evaluate the distance calculation and relative score algorithms in the proposal.

4.2.1 Comparison method 1

Comparison method 1 provides a simple way for recognizing gestures in an early stage, starting at the recognition process when the fixed time has elapsed from the beginning of the gestures. More concretely, at time t, the DTW distance is calculated between the t-length training data and t-length input data. Though the original DTW algorithm requires full length of data, it cannot be obtained at time t. Therefore, this method uses t-length input data and the first t samples of training data for DTW distance calculation. The label of the training data whose distance is the shortest overall is the recognition result.

Table 1. List of gestures.

ID	Description (Length[sec])	Illustration
1	Tilt to the near side (1.57)	
2	Tilt to the far side (1.43)	
3	Tilt to the left side (1.65)	
4	Tilt to the right side (1.73)	
5	Tap upper side twice (0.84)	
6	Tap left side twice (0.78)	
7	Quickly swing twice to the left side (1.36)	
8	Quickly swing twice to the right side (1.44)	
9	Shuffle cards (1.25)	
10	Tap lateral edge as though sifting (1.30)	
11	Scoop (1.84)	
12	Lay cards out (1.57)	
13	Gather cards (1.60)	
14	Rap table with the longer lateral edge (1.23)	
15	Rap table with the surface of the board (1.28)	
16	Knock the board twice (0.88)	
17	Turn the board over (1.30)	
18	Rotate clockwise on the table (1.55)	
19	Shift up (1.33)	
20	Shift down (1.09)	
21	Shift left (1.29)	
22	Shift right (1.17)	
23	Shift diagonally up (1.13)	
24	Shift diagonally down (1.06)	
25	Draw a circle (1.75)	
26	Draw a triangle (1.88)	
27	Draw a square (2.25)	

The recognition timing is set to the shortest length of all the training data. However, the recognition accuracies when using the comparison method 1 may decrease where some gestures are similar.

4.2.2 Comparison method 2

Comparison method 2 provides a simple way of deciding the recognition timing. At time t, the DTW distance is calculated between the first samples of training data and the t-length input data. This method calculates the distances for all the training data. If the shortest distance is smaller than β, the recognition result is output, otherwise wait for the next input at time $t+1$. The β $(10 \leq \beta \leq 500)$ is a threshold for the DTW distance.

4.3 Results and Considerations

Figure 4 shows the results of the average recognition accuracy vs. remaining time to the end of gesture for the comparison and proposed methods with each ε. For comparison method 1, the length of the input data is changed in the step of one sample. For comparison method 2, threshold β is changed from 10 to 500 with 10 intervals. For the proposed method, threshold α is changed from 0 to 1 with 0.1 intervals and searching width ε is set from 0 to 20 with 10 intervals. The vertical axis in the figure indicates the average recognition accuracy for the gestures and the horizontal axis is the remaining time from the output of the result to the ends of the gestures. The remaining time does not include the processing time for recognition, but only includes the length of the input data.

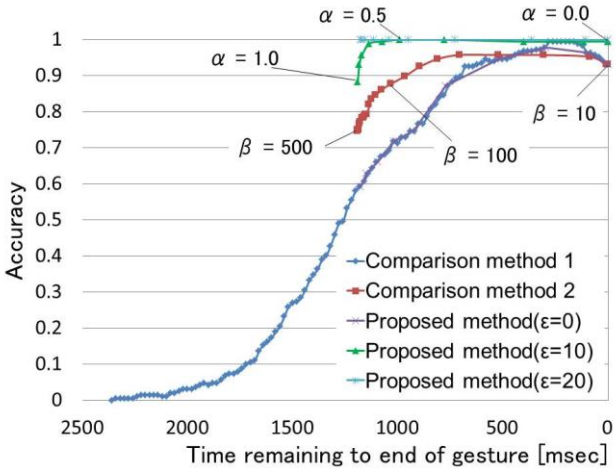

Figure 4. Accuracies vs. time remaining to the end of gesture.

The average accuracy for comparison method 1 shows more than 0.8 at 800 msec, which indicates that most of the gestures are correctly recognized 800 msec before the gesture finishes. However, gestures 8, 12, 18, 22, 25, and 27 are misrecognized at that time. Figures 5 to 10 show the results of the accuracy and the remaining time for the gestures. For gesture 8 (Quickly swing twice to the right side), comparison method 1 showed a low degree of accuracy at an early stage. This is because gesture 6 (Tap left side twice) is similar to gesture 8 in the early stage of the gestures. For the same reason, gesture 18 (Rotate clockwise on the table) was misrecognized for gesture 10 (Tap lateral edge as though sifting), and gesture 22 (Shift right) was misrecognized for gesture 8. In addition, gesture 12 (Lay cards out) was misrecognized for gesture 13 (Gather cards). Although these motion are reverse direction, misrecognition occurs in the early stage since the motions of the hand approaching the table before gathering or laying cards are similar. Although the accuracy of comparison method 1 reached 0.99 at 200 msec. From the results in Figure 4, comparison method 2 outputs results earlier when β is high and the accuracy is higher than that of comparison method 1. This is because comparison method 2 does not output results whose distances are large, while comparison method 1 outputs results regardless of the distance. However, these accuracies are reversed since comparison method 2 cannot output results due to severe threshold. For the results of comparison method 2 in

Figures 5 to 10, the accuracies of gestures 8 and 22 were improved at 950 msec in Figure 5 and at 500 msec in Figure 8. This is because the output of the gestures that comparison method 1 had misrecognized were suspended until the distance falls below the threshold.

Although the output timing can flexibly be controlled using the threshold, the threshold with which the recognition results are output in an early stage with a high degree of accuracy are different for the gestures. In Figure 9 the recognition accuracy is 1.00 at 580 msec with $\beta = 100$ and, in Figure 10 the accuracy with $\beta = 100$ had been in decline. It is hard to recognize all the gestures as early as possible at a high level of accuracy using a fixed threshold.

The proposed method recognizes gestures and outputs results 900 msec before a gesture finishes by setting the threshold to between 0.4 and 0.6. The recognition accuracy reaches 1.00 especially at $\alpha = 0.4$ and 0.5, which indicates the output is not delayed while maintaining high recognition accuracies. At points of $\alpha = 0$ to 0.4, the recognition accuracies are high, however, it takes a long time to output the recognition results. This is because the threshold is hard and longer input data is required. In the other hand, at $\alpha = 0.7$ to 1.0, the recognition accuracies drop sharply, which is caused by the low threshold level. This enables the system to output results earlier, but the number of misrecognitions increases since the input data length is too short.

With respect to the searching width ε, the low degree of average recognition accuracies are appeared with $\varepsilon = 0$ at low α value compared with $\varepsilon = 10$ and $\varepsilon = 20$ as shown in Figure 4. This is because the length of the training data is set to the same length of the input data when ε is set to 0, which was so strict that the distance increased. Spreading searching width, recognition accuracies with each α are improved. However, when searching width is large, DTW distance may not be calculated, which deteriorate recognition accuracy. Therefore, max of searching width needs to set.

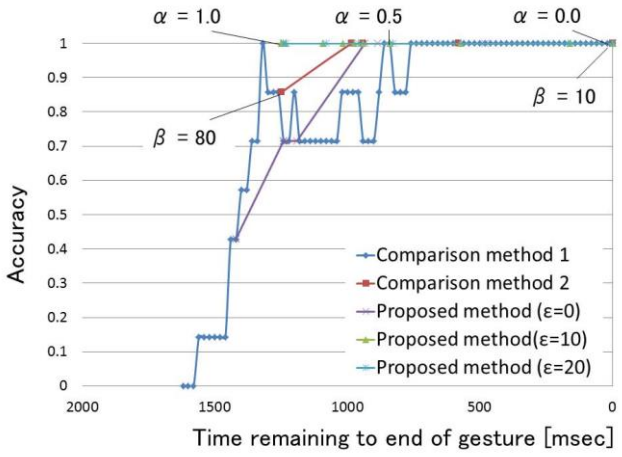

Figure 5. Gesture 8: Quickly swing twice to the right.

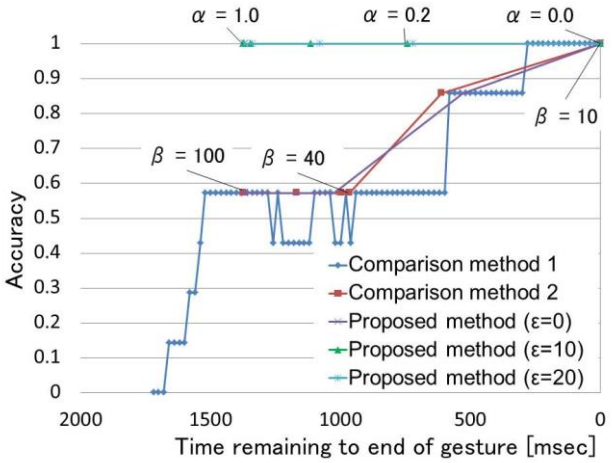

Figure 6. Gesture 12: Lay cards out.

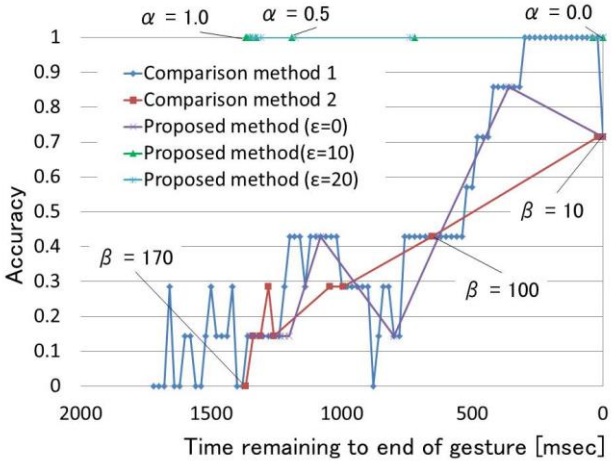

Figure 7. Gesture 18: Rotate clockwise on the table.

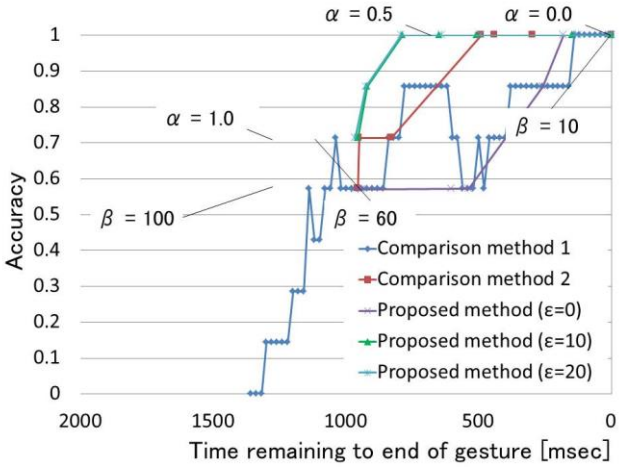

Figure 8. Gesture 22: Shift right.

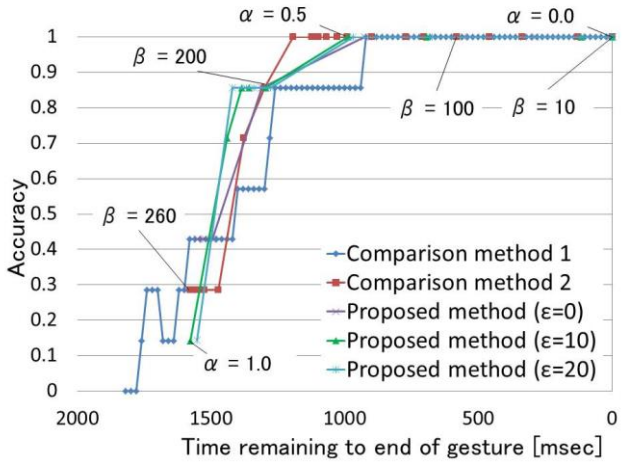

Figure 9. Gesture 25: Draw circle.

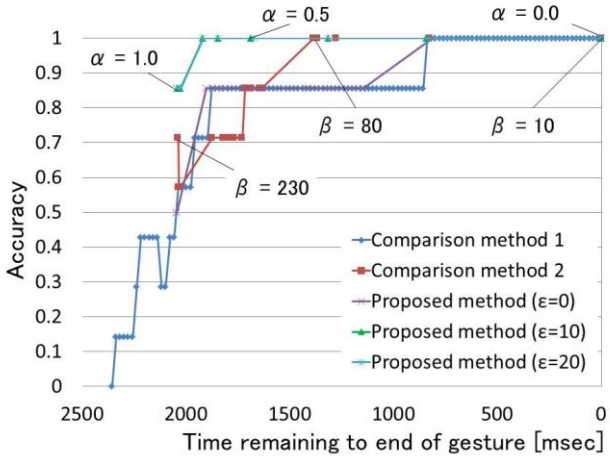

Figure 10. Gesture 27: Draw square.

5. APPLICATIONS

This section presents applications using our proposed method.

5.1 Gesture-based photo viewer

An example of applications that was enhanced with the early gesture recognition technology is the gesture-based photo viewer shown in Figure 11. This photo viewer uses several of the gestures we used in the evaluation. For example, tilt invokes a page to go back/forward, shift diagonal up invokes a zoom in on a photo, tap lateral edge as though shifting invokes a photo deletion a photo. Since conventional recognition processes generally start after the gestures have finished, the output of the recognition results and the feedback have a certain delay, which deprives of a comfortable operation. By using the early gesture recognition algorithm, the feedback is conducted before the gestures have finished.

In terms of the usability and recognition accuracy, we do not think all of the functions should be assigned to gestures. However, assigning many functions to a limited number of buttons forces users to select functions from a pop-up menu after they press the "menu" button. The first level has six to eight choices at most, forcing users to a select the "more" button to go to the next pop-up to select the other functions. Such functions can be invoked by a single action by using a gesture.

Recent smart phones have an approximately 5.0-inch display; for example, the GALAXY S Ⅲ by SAMSUNG has a 4.8-inch display and the ONE X by HTC has a 4.7-inch display. Users sometimes use both hands to touch the display since the thumb usually does not reach the far side of the display. Gesture-based interaction can be done to counter this. Our proposed method would help with the interaction to more comfortably use it.

Figure 11. Gesture-based photo viewer.

5.2 Practical use in theatrical performance

5.2.1 Participatory theatrical performance "YOUPLAY"

YOUPLAY is a 30-minute live entertainment. Anyone can join the theatrical performance as not an audience member but a player. Figure 12 shows some scenes of YOUPLAY. Most of the players are people who meet each other for the first time. We call this kind of performance a "participatory theatrical performance". The participants can play one of ten characters who have a special item such as a gun or butterfly net that plays an important role in the story. Depending on how they are used, the contents of the next scene changes. For example, a gun can be shot only one time in the story and its timing to shoot depends on the participant. YOUPLAY performed 40 stages as volume one from November 16th to 24th, 2013 at HEP HALL in Osaka, Japan. We joined it as technical staffs.

Figure 12. YOUPLAY Vol. 1

5.2.2 System structure

Figure 12 shows the system structure and the device used in the performance. An accelerometer, Arduino Fio, XBee, and Lithium ion battery are equipped in the middle of the butterfly net. We expanded the memory of the Arduino Fio by using a Serial SRAM chip made by Microchip Technology, Inc. because the existing Arduino Fio has a small SRAM memory. Transmitting heavy sensor data to the PC via wireless communication during a theatrical performance is not preferred because the other wireless equipment will be disrupted. Therefore, only the recognition results calculated on Arduino Fio are sent to the PC and the sound effects corresponding to the results are played from the stage speaker.

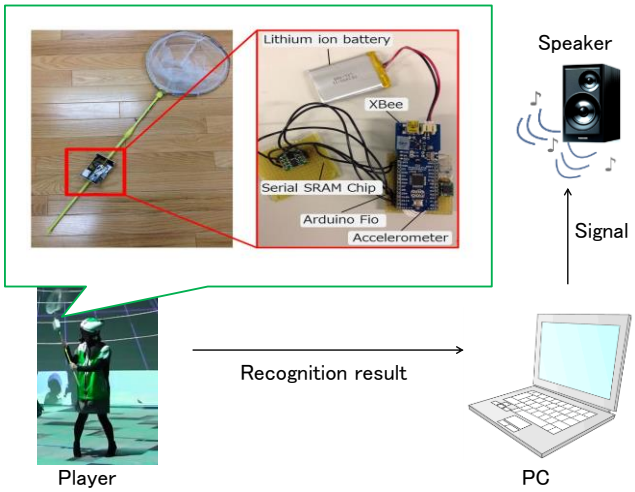

Figure 13. Structure of device.

This butterfly net is used in two scenes. One is a scene where the players introduce their characters by rotation as shown in Figure 7, and the other is a scene for capturing a rat projected on the floor as shown in Figure 8. This system is turned off in the other scenes so that incorrect sound effects are not produced. During the character introduction scene, the players were surprised at the sound effects. However, when the player who has the butterfly net was clapping hands, the sound effects were incorrectly output and obstructed the introduction of the other characters. Therefore, after this trouble, the recognition system was turned off after the player's introduction finished. In the scene of capturing a rat, when the player is swing the butterfly net, the sound effect was output in the middle of the motion and the gap between the sound and motion was reduced. When the player, however, runs around, the motion of the arm was misrecognized, which output the sound effect at inappropriate scene. This problem can be solved by registering the training data with null labels.

Figure 14. Introduction of characters.

Figure 15. Capturing a rat.

6. ACKNOWLEDGMENTS

This research was supported in part by a Grant in aid for Precursory Research for Embryonic Science and Technology (PRESTO) from the Japan Science and Technology Agency and by a Grant-in-Aid for Challenging Exploratory Research (25540084) from the Japanese Ministry of Education, Culture, Sports, Science and Technology.

7. CONCLUSION

We proposed an early gesture recognition system with an accelerometer in this paper. Our method successively calculates the distance between the input and training data to recognize gestures in an early stage with the high accuracy. Our system outputs recognition results at a high level of confidence by calculating the relative distance between the shortest and second shortest distance. The evaluation results showed that our proposed method was able to recognize gestures in an early stage of motion without deteriorating the accuracy.

In addition, we introduced gesture-based applications using our proposed method: photo viewer and sound effect system for participatory theatrical performance. The usability of the gesture interface was improved by reducing the delay in the output. In future work, we plan to investigate the ease and relations between the gestures, functions, and prediction by conducting a user-study on early recognition.

8. REFERENCES

[1] Y. Tomibayashi, Y. Takegawa, T. Terada, and M. Tsukamoto: Wearable DJ System: a New Motion-Controlled DJ System, Proc. of the International conference on Advance in Computer Entertainment Technology (ACE 2009), pp. 132--139 (2009).

[2] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan: uWave: Accelerometer-based Personalized Gesture Recognition and its Applications, Proc. of the IEEE International Conference on Pervasive Computing and Communication (PerCom 2009), pp. 1--9 (2009).

[3] J. Ruiz, and Y. Li: Double Flip: A Motion Gesture Delimiter for Mobile Interaction, Proc. of the International Conference on Human Factors in Computing Systems (CHI 2011), pp. 2717—2720 (2011).

[4] S. Agrawal, I. Constandache, S. Gaonkar, R. Choudhury, K. Caves, and F. Deruyter: Using Mobile Phones to Write in Air, Proc. of the International Conference on Mobile Systems, Applications, and Services (Mobisys 2011), pp. 1--28 (2011).

[5] C. Amma, M. Georgi, and T. Schultz: Hands-Free Mobile Text Input by Spotting and Continuous Recognition of 3d-Space, Proc. of the International Symposium on Wearable Computers (ISWC 2012), pp. 52--59 (2012).

[6] K. Yatani, K. Tamura, K. Hiroki, M. Sugimoto, and H. Hashizume: Toss-it: Intuitive Information Transfer Techniques for Mobile Devices, Proc. of the International Conference on Human Factors in Computing Systems (CHI 2005), pp. 1881--1884 (2005).

[7] A. Mori, S. Uchida, R. Kurazume, R. Taniguchi, T. Hasegawa, and H. Sakoe: Early Recognition and Prediction of Gestures, Proc. of the International Conference on Pattern Recognition (ICPR 2006), pp. 560--563 (2006).

[8] O.E. Scharenborg, L.W.J. Boves, L.F.M. Bosch, and S. Cassidy: Online Early Recognition of Polysyllabic Words in Continuous Speech, Proc. of the International Conference on Speech Science and Technology (SST 2004), pp. 387--392 (2004).

[9] H. Kanke, Y. Takegawa, T. Terada, and M. Tsukamoto: Airstic Drum: a Drumstick for Integration of Real and Virtual Drums, Proc. of the International Conference on Advance in Computer Entertainment Technology (ACE 2012), pp. 57--69 (2012).

[10] M. Fujimoto, N. Fujita, Y. Takegawa, T. Terada, and M. Tsukamoto: A Motion Recognition Method for a Wearable Dancing Musical Instrument, Proc. of the International Symposium on Wearable Computers (ISWC 2009), pp. 11--18 (2009).

[11] K. Murao, T. Terada, A. Yano, and R. Matsukura: Evaluation Study on Sensor Placement and Gesture Selection for Mobile Devices, Proc. of the International Conference on Mobile and Ubiquitous Multimedia (MUM 2012), No. 7, pp. 1--8 (2012).

[12] C.S. Myers and L.R. Rabiner: A Comparative Study of Several Dynamic Time Warping Algorithms for Connected Word Recognition, The Bell System Technical Journal, Vol. 60, pp. 1389--1409 (1981).

[13] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh: Search and Mining Trillions of Time Series Subsequences under Dynamic Time Warping, Proc. of the International Conference on Knowledge Discovery and Data mining (KDD 2012), pp. 262--270 (2012).

[14] E. Keogh and C.A. Ratanamahatana: Exact indexing of Dynamic Time Warping, The Knowledge and Information Systems Jounal, Vol. 7, pp. 358--386 (2004)

[15] Wireless technologies, Inc., available from<http://www.wireless-t.jp/>.