S & M 2440

Estimating Timing of Specific Motion in a Gesture Movement with a Wearable Sensor

Kazuya Murao,[1,2*] Hiroshi Yamada,[3] Tsutomu Terada,[3] and Masahiko Tsukamoto[3]

[1]Graduate School of Information Science and Engineering, Ritsumeikan University,
1-1-1 Nojihigashi, Kusatsu, Shiga 525-8577, Japan
[2]Strategic Creation Research Promotion Project (PRESTO) of the Japan Science and Technology Agency (JST),
4-1-8 Honmachi, Kawaguchi, Saitama 332-0012, Japan
[3]Graduate School of Engineering, Kobe University, 1-1 Rokkodaicho, Nada, Kobe, Hyogo 657-8501, Japan

(Received June 22, 2020; accepted November 6, 2020)

Keywords: accelerometer, gesture recognition, timing estimation

Wearable devices with motion sensors, such as accelerometers and gyroscopes, are expected to become popular. There is a lot of research on recognizing gestures using data obtained from motion sensors. A gesture is a one-off motion and its trajectory, i.e., the waveform of the gesture part, is considered to be important. After segmenting the data, gestures are recognized using a template matching method. However, there is no method to accurately detect when a specific action is performed during a gesture. Although it is possible for a player in a game to perform a throwing motion by the user performing a pitching action, it is difficult to reflect a particular moment, such as the user's release point, in the player. The authors previously proposed a method using a wrist-worn sensor for determining the moment of touching a card in competitive *karuta* (a Japanese card game) and developed a system that judges the player who took a card first in a competitive *karuta* match. As reported in this paper, we improved the estimation method to apply our study to a variety of gestures other than those in competitive *karuta* and propose a method of detecting the timing of a specific action. Our system was evaluated for three types of release points, baseball throws, basketball free throws, and dart throws, with 11 subjects who had an accelerometer and a gyroscope attached to the wrist. The percentage of release point estimation errors of 12 ms or less was determined to be 100% for baseball, 87.6% for basketball, and 91.1% for darts.

1. Introduction

Along with the spread of wearable devices embedded with motion sensors, such as smartwatches and smart glasses, the research and development of applications that recognize gestures using data obtained from motion sensors has been actively conducted. The Moff Band by Moff, Inc.(1) has an installed motion sensor, enabling sound play, such as ninja throwing knife and guitar. Smartphones, such as the iPhone by Apple Inc. and Android-powered devices, and remotes of video games, such as those of Nintendo Switch, also have installed motion sensors to

[*] Corresponding author: e-mail:murao@cs.ritsumei.ac.jp

https://doi.org/10.18494/SAM.2021.2964

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

detect the tilting and motion of the device, enabling the user to control the game characters and draw objects intuitively.

Human activities that have been dealt with in many studies are postures, such as sitting, and behaviors, such as walking, which are states in human activities lasting for a certain length of time. They are generally recognized with a classifier such as a support vector machine (SVM) or random forest (RF) operating on extracted feature values, such as the mean, variance, and fast Fourier transform (FFT) power spectrum, that express body orientation and exercise intensity. Other important activities in daily life include gestures, e.g., punches. Gestures are not states but once-off actions, and they can be recognized with a template matching algorithm such as dynamic time warping (DTW)(2) after trimming the waveform of the gesture.

DTW calculates the temporal nonlinear elastic distance between two sequences of the same gestures that vary in time or speed, therefore timings of specific motions in a gesture are not taken into account in gesture recognition. By applying gesture recognition technology to video games such as Wii Sports,(3) a game user can make a character in the game throw a ball by making a gesture of throwing a ball, but information of specific timings such as the release point cannot be reflected by the character in the game.

In theory, given a timestamp of a specific motion labeled in the training data, the time of the specific motion in the input data can be estimated with DTW since the DTW algorithm can find the correspondence of samples of training and input data. However, waveforms of complicated gestures, such as that of throwing a ball, include many peaks, and these peaks generally do not match completely in the DTW algorithm, resulting in a large estimation error.

The authors previously proposed a system that judges which player took a card first in a competitive *karuta* (Japanese cards) match.(4) The system measures the motion data when players take a card by using a wrist-worn accelerometer and a gyroscope, and estimates the times when the players touch the card. Generally, competitive *karuta* is played without a referee, so players must judge themselves (self-judgement) even if a difficult situation arises. Most rounds are not controversial, but sometimes players get into an argument over who touched a card first, which disrupts the other matches in the room because multiple matches are simultaneously played in parallel with one reciter in a large room.

In this paper, in order to apply the method of estimating the card touch time to other gestures, we improve the method to find the threshold parameter that had been set manually in our previous work. We assume that the users wear a sensor on their wrist such as a smartwatch and we evaluate our method for baseball pitching, basketball free throws, and dart throws. Our motion timing estimation method can be used for video games and virtual reality/augmented reality (VR/AR) systems in which the characters can be manipulated by moving the user's body. This research has been approved by Human Ethics Committee of Graduate School of Engineering, Kobe University.

This paper is organized as follows. Section 2 introduces related work on gesture recognition and motion timing estimation, Sect. 3 explains the proposed system, and Sect. 4 evaluates the performance of our system. Lastly, Sect. 5 concludes this paper.

2. Related Work

Activity recognition using wearable sensors has mainly tackled tasks to classify an unknown activity into one of a number of predefined activity classes, while there have been some studies on detecting the moment when the motion changes.

2.1 Gesture recognition

There have been many studies on activity recognition using wearable sensors, some of which have been applied to sports. Kos and Kramberger(26) proposed a miniature wearable device for detecting and recording the movement and biometric information of a user during sport activities. The device weighs 5.8 g and has an accelerometer, a gyroscope, a temperature sensor, and a pulse sensor. Lapinski *et al.*(5) evaluated professional baseball pitchers and batters by using wearable sensor systems, and Ladha *et al.*(6) proposed a climbing performance analysis system using a watch-like sensing platform that measures acceleration. Kosmalla *et al.*(7) also proposed a system for climbing using wrist-worn inertia measurement units. The system can automatically recognize the route that a climber took during a climbing session. Bächlin *et al.*(8) built a system consisting of sensing and feedback hardware for swim analysis. The system opens up exciting new possibilities in the field of swimming training, as objective values can be provided at all times for complete training. Lee *et al.*(18) proposed a hand gesture recognition algorithm with an inertial sensor and a magnetometer. Six gestures were tested and achieved an average recognition accuracy of 98.75%. None of these systems, however, can estimate the instant an action is performed.

Zhou *et al.*(9) constructed a system that uses textile pressure-sensing matrices. The system can distinguish different ways in which a player's foot strikes the ball. Connaghan *et al.*(10) investigated tennis stroke recognition using a single inertial measuring unit (IMU) attached to a player's forearm. They classified tennis strokes into serves, forehand, and backhand. However, these studies did not measure the timing of the ball being struck.

Blank *et al.* presented an approach for ball impact localization on table tennis rackets using piezoelectric sensors.(11) However, they did not examine the precision of the ball impact timing. The same group also proposed a system that uses inertial sensors attached to a table tennis racket.(12) The system detected table tennis strokes by using an event detection method. This method detected strokes with an accelerometer installed on the racket grip and achieved a precision of 0.957 and a recall of 0.982.

2.2 Motion timing estimation

Chi *et al.*(19) proposed a system that assists the umpires in Taekwondo matches by attaching piezoelectric sensors to the body protectors of the players. Helmera *et al.*(20) proposed an automated scoring system for amateur boxing by attaching an array of piezoelectric sensors to the players' vests. Maglott *et al.*(27) investigated the difference of arm motion during basketball shooting. They used a tight-fitting stretchable sleeve embedded with two 9-axis inertial measurement units (IMUs). From their experiment, it is reported that trained shooters shot free throws faster than novice shooters; however, only the timing of peaks is compared and the motion is not considered. Kim and Park(29) developed a golf swing segmentation algorithm from 3-axis acceleration and 3-axis angular velocity data. The algorithm divides the input sequence into five major predefined phases with an average segmentation error of 5–92 ms. Lian *et al.*(28) developed a recognition algorithm for six serial phases of a throwing action in baseball from acceleration data. They achieved a recognition accuracy of 91.42–95.14% for three test subjects for the six phases; however, the estimation error of the segmentation was not evaluated. Moreover, Mencarini *et al.*(25) surveyed and reviewed a corpus of 57 papers published from 1999 to 2018 regarding HCI research tackling on wearable technology in the sports domain.

Kanke *et al.*(13) proposed the Airstic Drum, which is a drumstick with an accelerometer to play an actual drum by physically striking the drum surface in front of the user and a virtual drum by striking the air. When hitting the real drum, the actual sound is produced, and when hitting the virtual drum, the sound is output from the system. Airstic Drum identifies whether the object hit is a real or virtual drum before the moment that the drum is hit, and only outputs a sound when the virtual drum is hit. However, the difference between the moment of striking and the moment of sound output was not quantitatively evaluated. In addition, the algorithm was specialized for detecting drum strikes and it is unknown whether the system can be applied to other activities.

The current authors(14) proposed a method that recognizes gesture activities while moving with high accuracy. The method judges the constancy, i.e., the periodicity of the waveform, of human activities by calculating the autocorrelation of acceleration values and conducts gesture recognition only when the constancy breaks. In this study, we did not evaluate whether the moment when the constancy breaks is the correct starting point of the gesture, and the constancy decision was made every 800 ms; therefore, it is difficult to use the method for detecting timing with an accuracy of 10 ms. Yoshizawa *et al.*(15) proposed a method that finds the changing point of activities from acceleration data and obtained a precision of 50% for changing point detection when the allowable error was within 1800 ms.

We also proposed a system that judges which player took a card first in a competitive *karuta* match.(4) In competitive *karuta*, the time difference between different players touching a card is extremely small, and our proposed system distinguishes time differences of milliseconds. In this study, we improve the method of finding the threshold parameter that had been set manually in our previous work and evaluate our method for baseball pitching, basketball free throws, and dart throws.

3. Proposed System

In this section, we explain the proposed method used to estimate the timing of a specific motion in a gesture.

3.1 System structure

We propose a system that uses an inertial sensor attached to the wrist of the user's dominant hand, as shown in Fig. 1. The sensor used in the system contains a wireless three-axis

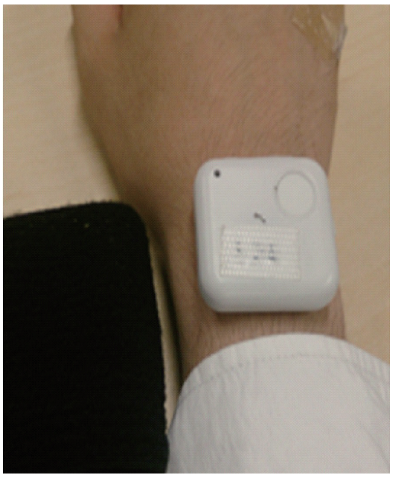

Fig. 1. (Color online) Loading position of three-axis accelerometer and gyroscope.

accelerometer and a gyroscope (WAA-010 by Wireless Technologies, Inc.(16)). The sensor has dimensions of W39 × H44 × D12 (mm^3) and weighs only 20 g. In other words, the sensor is small and light and does not interfere with gestures. The proposed system estimates the time of a specific motion based on the gesture data labeled with correct motion timing.

Figure 2 shows the flow of the system. A user's movement is captured through the small wrist-worn sensing device, which is configured to record three-axis acceleration and angular velocity data. The sensor data is sent to a device such as a smartphone via Bluetooth, and the system installed on the device compares the input data with the training data. Then, the time at which the user performed the specific motion is estimated. The exact time of the specific motion is labeled with the training data, which is collected in advance. The confidence of the time estimation is then calculated. Lastly, our system outputs the estimated time.

3.2 Data segmentation

Since the sensor data is captured before and after the gesture, the system detects the gesture within the stream of acceleration and angular velocity data, and extracts the data. The system calculates the composite value of the three-axis accelerometer $A(t)= \sqrt{a_x^2(t)+a_y^2(t)+a_z^2(t)}$, where $a_x(t)$, $a_y(t)$, and $a_z(t)$ are the acceleration values in the x-, y-, and z-axes directions, respectively. If condition $A(t) > Th_s$ is first satisfied for T_s ms, the proposed system determines that the gesture movement begins at time T_{start}. Then, if condition $A(t) > T_e$ is satisfied for T_e ms, the system determines that the gesture finishes at time T_{end}. α and β are thresholds of the start and end of a gesture, set to 1300 and 1100 mG on the basis of a pilot study, respectively. The following segmented data $\boldsymbol{G}$ is obtained through the data segmentation, where $g_x(t)$, $g_y(t)$, and $g_z(t)$ are the angular velocities around the x-, y-, and z-axes, respectively.

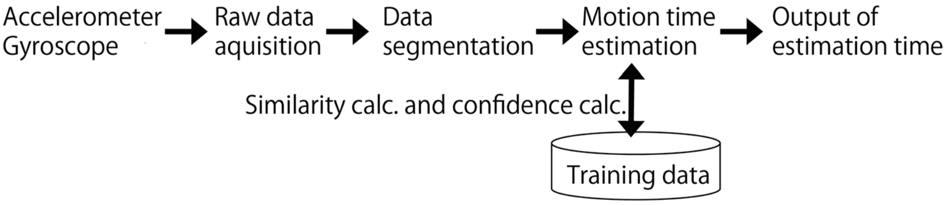

Fig. 2. Algorithm for estimating timing of specific motion.

$$\boldsymbol{G}=\begin{pmatrix} a_x(T_{start}) & a_x(T_{start}+1) & \cdots & a_x(T_{end}) \\ a_y(T_{start}) & a_y(T_{start}+1) & \cdots & a_y(T_{end}) \\ a_z(T_{start}) & a_z(T_{start}+1) & \cdots & a_z(T_{end}) \\ g_x(T_{start}) & g_x(T_{start}+1) & \cdots & g_x(T_{end}) \\ g_y(T_{start}) & g_y(T_{start}+1) & \cdots & g_y(T_{end}) \\ g_z(T_{start}) & g_z(T_{start}+1) & \cdots & g_z(T_{end}) \end{pmatrix} \tag{1}$$

3.3 Acquisition of training data

The proposed system compares the segmented and training data to estimate the specific motion time. One might think that the motion time in the segmented data can easily be estimated only by seeing the waveform, without training data. However, this is difficult because the motion time in the segmented data does not always correspond to the peak of the data obtained from the sensor attached to the wrist. For example, the ball release point is the point of maximum palm speed in baseball pitching, which means that acceleration does not show a peak.(21,22) The gestures are video-recorded with a high-speed camera (SONY Cyber-shot RX100ICV (DSC-RX100M4)) at 960 fps and the exact time of each motion is manually obtained and given to the training data afterward.

3.4 Estimation of motion time

The proposed system utilizes two methods of estimating the motion time: a feature-value-based method (method 1) and a waveform-similarity-based method (method 2). By combining these two methods, the proposed system estimates the motion time. The algorithms of the methods are described in detail below.

3.4.1 Method 1: feature-value-based method

The feature-value-based method uses a sliding-window approach. Given the segmented data of gesture ***G*** in Eq. (1), feature values are extracted over a three-sample window that is slid in steps of one sample. The feature values used are max, min, and variance for the six axes (6 axes × 3 features = 18 dimensions) and the angle of the wrist for the three axes (3 axes × 1 feature = 3 dimensions), giving 21 dimensions in total. The angle of the wrist is calculated by integrating the angular velocity. These feature values $\boldsymbol{F}(t) = (f_1(t), f_2(t), ..., f_{21}(t))$ are calculated over the window $[t-1, t, t+1]$ from $t = T_{start} + 1$ to $T_{end} - 1$. Feature values are also calculated from the training data at motion time T_{true} only, i.e., $\boldsymbol{F}(T_{true})$ is calculated.

Feature vector $\boldsymbol{F}(t)$ is standardized using $\boldsymbol{Z} = (\boldsymbol{F}(\mathrm{t}) - \boldsymbol{M})/\boldsymbol{S}$ since the scales of the feature values are different, where $\boldsymbol{M} = (m_1, ..., m_{21})$ and $\boldsymbol{S} = (s_1, ..., s_{21})$ are respectively the mean and standard deviation of $\boldsymbol{F}(t)$ over the training data. After this conversion, the 21-dimensional feature vector $\boldsymbol{Z}(t) = (z_1(t), z_2(t), ..., z_{21}(t))$ is obtained, whose mean and variance become 0 and 1, respectively.

The Euclidean distance between the ith training data $\boldsymbol{Z}_{tr} = (Z_{tr,1}^{(i)}, Z_{tr,2}^{(i)}, ..., Z_{tr,21}^{(i)})$ and the input data $\boldsymbol{Z}_{in} = (Z_{in,1}^{(i)}, Z_{in,2}^{(i)}, ..., Z_{in,21}^{(i)})$ is calculated as

$$Euclid\left(\boldsymbol{Z}_{tr}^{(i)}, \boldsymbol{Z}_{in}(t)\right) = \sqrt{\sum_{j=1}^{21}\left(z_{tr,j}^{(i)} - z_{in,j}(t)\right)^2} \ . \tag{2}$$

The system calculates $Euclid(\boldsymbol{Z}_{tr}^{(i)}, \boldsymbol{Z}_{in}(t))$ from $t = T_{start} + 1$ to $t = T_{end} - 1$ and from $i = 1$ to N for all the training data collected and finds T_{min} when $Euclid(\boldsymbol{Z}_{tr}^{(i)}, \boldsymbol{Z}_{in}(t))$ shows the minimal value, where N is the number of training data. T_{min} is estimated as the motion time of the input data since the waveform of the input data around T_{min} is similar to that of the training data at the motion time.

3.4.2 Method 2: waveform-similarity-based method

The waveform-similarity-based method calculates the similarity between training and input data by DTW,[2] which measures the similarity of two time-series data. Advantages of DTW include the ability to calculate the temporal nonlinear elastic distance, the ability to measure the similarity between two sequences that may vary in time or speed, and the fact that the number of samples in the two time series need not be equal. The details of the algorithm are as follows. For simplicity, we explain the algorithm for one-dimensional data.

When training data $\boldsymbol{X} = (x_1, ..., x_m)$ and input data $\boldsymbol{Y} = (y_1, ..., y_n)$ with lengths m and n, respectively, are compared, an m×n matrix is defined as $d(i, j) = |x_i - y_i|$. Next, warping path $\boldsymbol{W} = (w_1, ..., w_k)$, which is the path of the pairs of $\boldsymbol{X}$ and $\boldsymbol{Y}$ indices, is found. $\boldsymbol{W}$ satisfies three conditions.

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

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

Initialization:

$f(0, 0) = 0$

$f(i, 0) = \infty$ for $i = 1, 2, \ldots, m$

$F(0, j) = \infty$ for $j = 1, 2, \ldots, n$

```
Do for $i = 1, 2, \ldots, m$
    Do for $j = 1, 2, \ldots, n$
        $f(i,j) = d(i,j) + \min \begin{cases} f(i-1, j-1) \\ f(i-1, j) \\ f(i, j-1) \end{cases}$
Output:
    Return $\frac{f(m,n)}{(m+n)}$
```

The obtained cost $\frac{f(m,n)}{(m+n)}$ is the distance between ***X*** and ***Y***. The returned value is divided by the sum of the lengths of the input and training data since the DTW distance increases with the length of the sequences.

The motion time for the input data is estimated by finding the index of input data corresponding to the index of the motion time in the training data on the warping path, as shown in Fig. 3. If multiple indices of input data correspond to the index of the motion time in the training data, the estimated motion time is set to the earliest index.

3.5 Judgement of confidence flag

Since motions are not always similar even for the same gesture, the proposed system has to consider anomalous input data. Even if the input data is completely different from the training data owing to an unintended motion or hitting an object, the motion time is estimated, resulting in an inaccurate measurement. To address this problem, our system utilizes a confidence flag that represents whether the estimated motion time is reliable or not. If it is reliable, the confidence flag becomes “HIGH”, and if not, it is “LOW”.

Figure 4 shows the algorithm used to judge the confidence flag. Suppose that N samples of training data are collected in advance. The actual motion time is labeled with the training data and the motion time is estimated for the training data by methods 1 and 2 independently in an N-fold cross-validation manner. The difference between the labeled motion time (ground truth) and the estimated motion time becomes the error. If the error is less than or equal to α ms, the confidence flag HIGH is given to the training data, if not, LOW is given. How to set α is

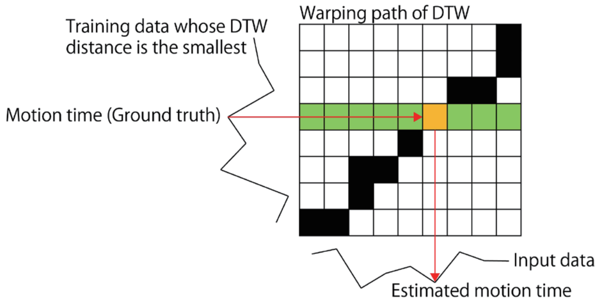
Warping path of DTW
Training data whose DTW distance is the smallest
Motion time (Ground truth)
Input data
Estimated motion time

Fig. 3. (Color online) Estimation of motion time by using DTW.

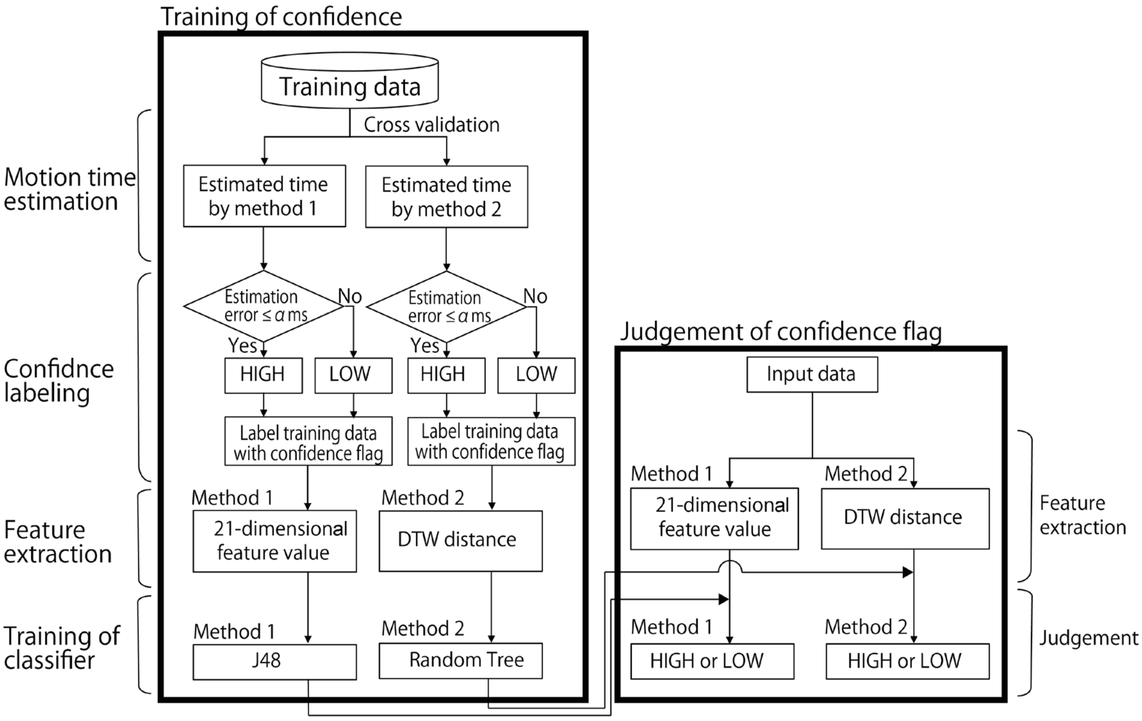

Fig. 4. Algorithm for judging confidence flag.

explained later. The training data with the LOW flag is considered as the anomalous data since no other training data is close to it.

The proposed system classifies the input data into HIGH and LOW using the model that has learned training data labeled with the confidence flags in parallel with estimating the motion time by methods 1 and 2. The confidence flags are trained and classified, and the motion time is estimated for each method. For method 1, 21-dimensional feature values over the window at $t = T_{min}$ extracted in method 1 are trained with the J48 classifier, which is a C4.5 algorithm implemented on WEKA.(17) For method 2, DTW distances between the input data and the best-matching training data calculated in method 2 are trained with Random Tree, which is a decision tree algorithm implemented on WEKA. The reason that Random Tree is employed is that J48 did not work well for the scalar explanatory variable, i.e., DTW distance. The confidence of methods 1 and 2 can be obtained by classifying the input data.

3.6 Calculation of the threshold

Here, the method of determining the threshold α when labeling the confidence flag as HIGH or LOW is described. Figure 5 shows the algorithm for determining α.

Firstly, the estimation error is calculated in the same manner as in Sect. 3.4. The estimation error is an integer multiple of the sampling interval T_{inr}. Let the maximal and minimal estimation errors be $N_{max}T_{inr}$ and $N_{min}T_{inr}$, respectively. The α range in which the confidence

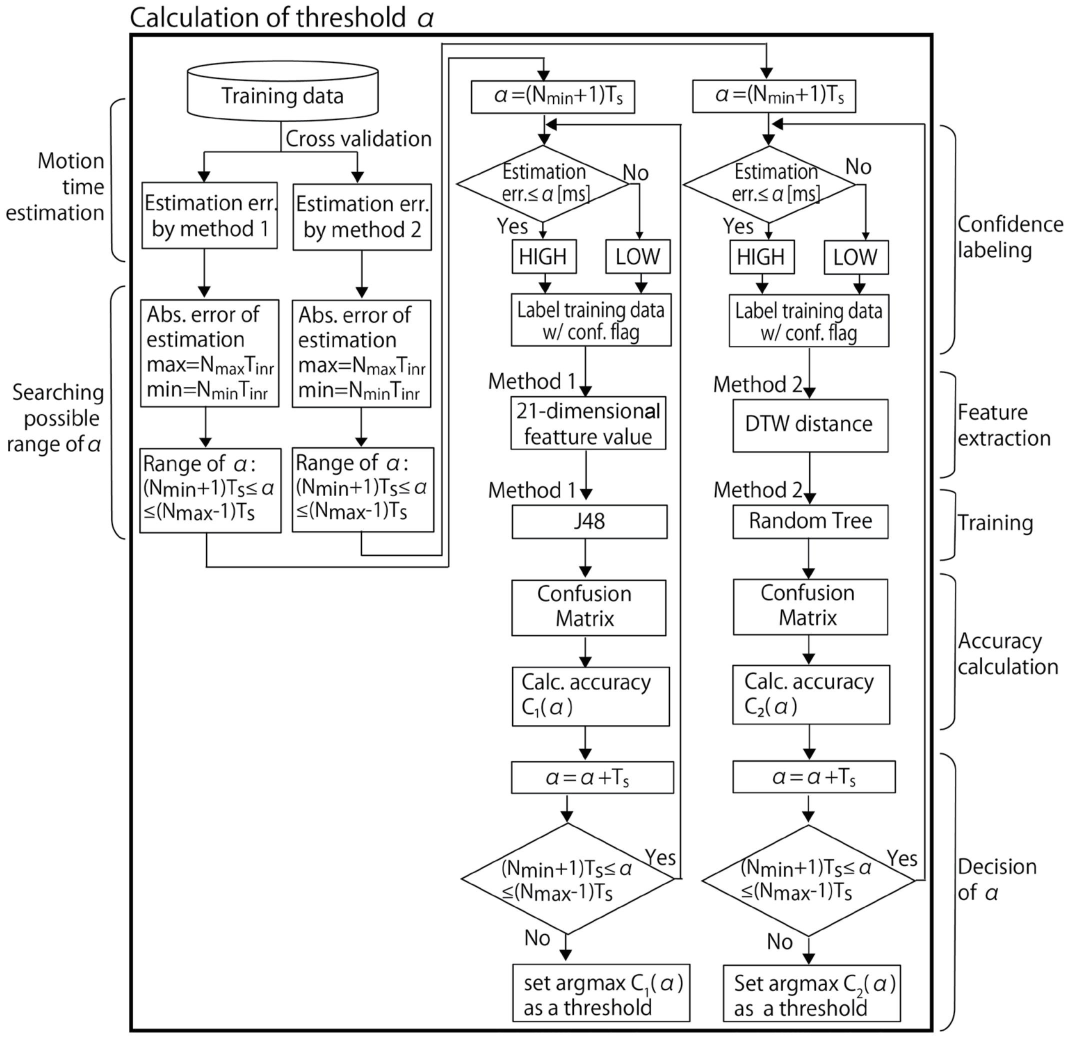
Calculation of threshold α
Motion time estimation
Searching possible range of α
Training data
Cross validation
Estimation err. by method 1
Estimation err. by method 2
Abs. error of estimation max=N_max T_inr min=N_min T_inr
Abs. error of estimation max=N_max T_inr min=N_min T_inr
Range of α: (Nmin+1)Ts≤ α ≤(Nmax-1)Ts
Range of α: (Nmin+1)Ts≤ α ≤(Nmax-1)Ts
α=(Nmin+1)Ts
α=(Nmin+1)Ts
Estimation err.≤ α [ms]
Estimation err.≤ α [ms]
Yes
No
HIGH
LOW
Label training data w/ conf. flag
Label training data w/ conf. flag
Method 1
Method 2
21-dimensional featture value
DTW distance
J48
Random Tree
Confusion Matrix
Calc. accuracy C1(α)
Calc. accuracy C2(α)
α=α+Ts
(Nmin+1)Ts≤ α ≤(Nmax-1)Ts
set argmax C1(α) as a threshold
Set argmax C2(α) as a threshold
Confidence labeling
Feature extraction
Training
Accuracy calculation
Decision of α

Fig. 5. Algorithm for calculating threshold *α*.

flags of training data include both HIGH and LOW is $(N_{min} + 1)T_s \leq \alpha \leq (N_{max} - 1)T_s$. The assignment of HIGH and LOW flags changes in this range since all the confidence flags are HIGH in the range $\alpha < (N_{max} - 1)T_s$ and LOW in the range $\alpha > (N_{min} + 1)T_s$. Then, the classifiers are trained using the feature values labeled with the confidence flags in methods 1 and 2.

In order to find the best *α*, the proposed method creates a confusion matrix of the classification results of the confidence flags in a cross-validation manner to evaluate the classification accuracy for methods 1 and 2 by changing *α*. A confusion matrix is a table showing the classification results for both the input and the output: e.g., a 2 × 2 matrix in a two-class classification problem. True positives (TPs) and true negatives (TNs) are the correct classifications and false positives (FPs) and false negatives (FNs) are incorrect classifications. Then, accuracy $C(\alpha) = \frac{(TP + TN)}{(TP + TN + FP + FN)}$ is calculated for *α* in the range $(N_{min} + 1)T_s \leq \alpha \leq (N_{max} - 1)T_s$, and $C_1(\alpha)$ and $C_2(\alpha)$ are obtained for methods 1 and 2, respectively. Lastly, the value of α when $C_1(\alpha)$ and $C_2(\alpha)$ show the highest value is set to the threshold. A high $C(\alpha)$ means that the confidence flags are classified with high accuracy; therefore, strange input data, i.e., data potentially producing a large estimation error, will be given a LOW flag in the judgement of the confidence flag phase explained in Sect. 3.5.

3.7 Output of estimation timing

Since preliminary experiments showed that the motion times estimated by the methods are not always accurate, the proposed method outputs the estimated motion time by combining both methods and considering the confidence flags. There are four combinations: two confidence flags and two methods. The conclusive estimated motion time is adopted according to the following rules.

- If the confidence flag of method 2 is HIGH, the motion time estimated by method 2 is adopted regardless of the confidence of method 1. This is because a preliminary experiment showed that the accuracy of estimating the motion time by method 2 was superior to that by method 1.
- Otherwise, if the confidence flag of method 1 is HIGH, the motion time estimated by method 1 is adopted.
- If the confidence flags of both methods are LOW, the motion time is not estimated and UNIDENTIFIED is output.

4. Evaluation

This section evaluates the estimation error of the proposed method applied to baseball pitching, basketball free throws, and dart throws. All three gesture throwing, but we adopted them as different movements. In baseball pitching, a whole arm is used. Basketball free throws mainly use the wrist. Dart throws use the arm with fixing the elbow.

4.1 Baseball pitching

4.1.1 Environment

We evaluated the performance of estimating the ball release time in baseball pitching. Data of pitching action were captured 70 times in total from three subjects A, B, C (all right-handed males) through the proposed system by attaching a wireless sensor to their dominant hand through the proposed system. As an indicator of the performance, we measured the error of the estimated motion time, which is the difference between the estimated release time and the exact release time. The experiment was video-recorded with a 960 fps high-frame-rate camera. We examined the video and added the release time to the sensor data. Acceleration and angular velocity data were collected at 333 Hz. We estimated the release time for each subject independently in a leave-one-sample-out cross-validation manner.

4.1.2 Results

Table 1 shows the threshold α giving the highest accuracy of confidence flag classification. For example, for subject A, the minimal estimation error $N_{min}T_s$ and maximal estimation error $N_{max}T_s$ obtained through cross-validation by method 1 within the training data are 0 and 18 ms, respectively. The sampling interval T_s is 3 ms so $N_{min} = 0$ and $N_{max} = 6$ are calculated, resulting in α in the range $3 \leq \alpha \leq 15$. By calculating the accuracy of confidence flag classification $C(\alpha)$ in the range of α, the maximal accuracy of 0.92 is obtained at $\alpha = 9$ (ms).

Figure 6 shows the histogram of the estimation error of the release point. Outputs of UNIDENTIFIED, i.e., the decision when the confidence flags for both methods 1 and 2 are LOW, are removed from the results. From the results, the largest error is 12 ms and 15.9% of the estimated motion times are exact (±0 ms error). Considering that the sampling interval of the system is 3 ms, 61.9% of the errors are within ±3 ms. The mean absolute error is 3.75 ms.

A 3 ms estimation error means a 10 cm error for baseball pitching as the speed of the hand immediately before releasing a ball is 30 m/s. UNIDENTIFIED was output seven times out of 70 trials, all of which occurred for subject B. This is because method 1 set α to 3 ms for subject B; therefore, even the outputs whose estimation error was 3 ms were given the LOW confidence flag, resulting in UNIDENTIFIED.

Table 1
Results of threshold α for release point of baseball pitching.

Method	Subject A		Subject B		Subject C	
	1:Feature	2:DTW	1:Feature	2:DTW	1:Feature	2:DTW
$N_{min}T_s$ (ms)	0	0	0	0	0	0
$N_{max}T_s$ (ms)	18	18	48	36	9	6
Range of α	[3, 15]	[3, 15]	[3, 45]	[3, 33]	[3, 6]	[3, 3]
max$C(\alpha)$	0.92	0.96	0.90	0.90	0.90	0.90
$\underset{\alpha}{\mathrm{argmax}}\, C(\alpha)$	9	12	3	6	6	3

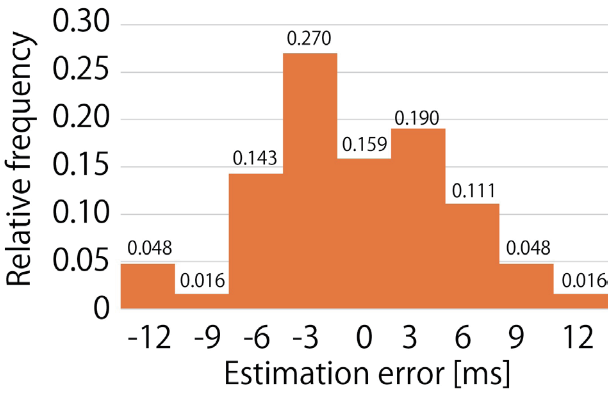

Fig. 6. (Color online) Histogram of error of estimated release point of baseball pitching.

4.2 Basketball free throws

4.2.1 Environment

We evaluated the performance of estimating the ball release time in basketball free throws. Data of free throws were captured 20 times from five subjects D, E, F, G, and H (all males) by attaching a wireless sensor to their dominant hand through the proposed system; a total of 100 samples were collected. Four subjects were right-handed and one subject was left-handed. Two of the subjects had more than three years of experience playing basketball. As an indicator of the performance, we measured the error of the estimated motion time, which is the difference between the estimated release time and the exact release time. The experiment was video-recorded with a 960 fps high-frame-rate camera. We examined the video and added the release time to the sensor data. Acceleration and angular velocity data were collected at 1000 Hz. We estimated the release time for each subject independently in a leave-one-sample-out cross-validation manner.

4.2.2 Results

Table 2 shows the threshold α giving the highest accuracy of confidence flag classification, and Fig. 7 shows the histogram of the estimation error of the release point. Outputs of

Table 2
Results of threshold α for release point of basketball free throws.

Method	Subject D		Subject E		Subject F		Subject G		Subject H	
	1:Feature	2:DTW	1:Feature	2:DTW	1:Feature	2:DTW	1:Feature	2:DTW	1:Feature	2:DTW
$N_{min}T_s$ (ms)	0	1	2	1	0	0	52	0	0	0
$N_{max}T_s$ (ms)	8	15	115	113	22	18	366	171	11	17
Range of α	[1, 7]	[2, 14]	[3, 114]	[2, 112]	[1, 21]	[1, 17]	[53, 365]	[1, 170]	[1, 10]	[1, 16]
$\max C(\alpha)$	0.70	0.90	0.95	0.90	0.85	0.90	0.95	0.90	0.95	0.90
$\underset{\alpha}{\operatorname{argmax}}\, C(\alpha)$	6	14	113	13	15	8	53	4	9	11

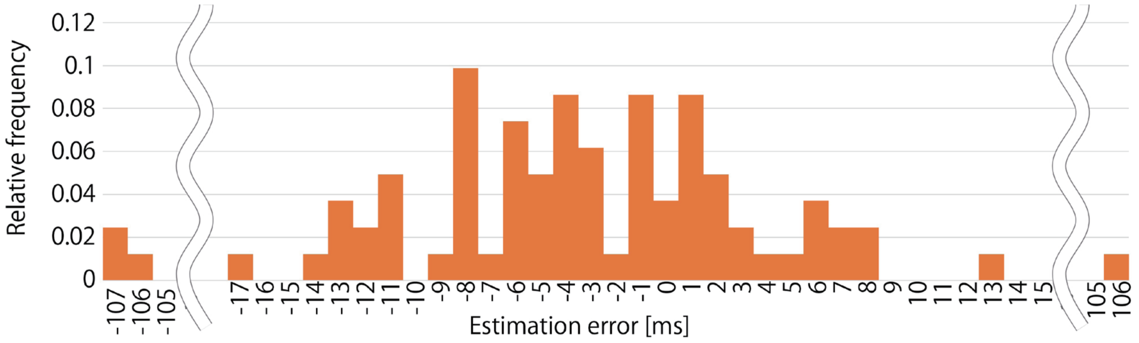

Fig. 7. (Color online) Histogram of error of estimated release point of basketball free throws.

UNIDENTIFIED, i.e., the decision when confidence flags for both methods 1 and 2 are LOW, are removed from the results. From the results, the largest error is −107 ms and 21.0% of the estimated release points have an error within ±1 ms. The mean absolute error is 6.28 ms.

UNIDENTIFIED was output 19 times out of 100 trials, all of which occurred for subject G. The estimation error for subject G was large even when α was set to as much as 53 ms, resulting in 19 out of 20 samples having the LOW confidence flag. From the table, subjects E and G showed larger estimation errors than subjects D, F, and H for both methods 1 and 2. This was considered to be because subjects E and G had more than three years of experience playing basketball and their free throw motions were flexible, while the other subjects performed stable wrist movements.

4.3 Dart throws

4.3.1 Environment

We evaluated the performance of estimating the timing of releasing darts. Data of throwing action were captured 30 times from three subjects I, J, and K (all right-handed males) by attaching a wireless sensor to their dominant hand through the proposed system; a total of 90 samples were collected. As an indicator of the performance, we measured the error of the estimated motion time, which is the difference between the estimated release time and the exact release time. The experiment was video-recorded with a 960 fps high-frame-rate camera. We examined the video and added the release time to the sensor data. Acceleration and angular velocity data were collected at 1000 Hz. We estimated the release time for each subject independently in a leave-one-sample-out cross-validation manner.

4.3.2 Results

Table 3 shows the threshold α giving the highest accuracy of confidence flag classification, and Fig. 8 shows the histogram of the estimation error of the release point. Outputs of UNIDENTIFIED, i.e., the decision when confidence flags for both methods 1 and 2 are LOW, are removed from the results. From the results, the largest error is −107 ms and 20.0% of the estimated release points are within ±1 ms. The mean absolute error is 4.51 ms. No UNIDENTIFIED was output for the dart throw data.

Table 3
Results of threshold α for release point of dart throws.

Method	Subject I		Subject J		Subject K	
	1:Feature	2:DTW	1:Feature	2:DTW	1:Feature	2:DTW
$N_{min}T_s$ (ms)	0	0	0	0	0	0
$N_{max}T_s$ (ms)	12	11	17	17	97	83
Range of α	[1, 11]	[1, 10]	[1, 16]	[1, 16]	[1, 96]	[1, 82]
max$C(\alpha)$	0.97	0.97	0.97	0.97	0.97	0.90
$\underset{\alpha}{\text{argmax}}\, C(\alpha)$	10	12	10	12	86	55

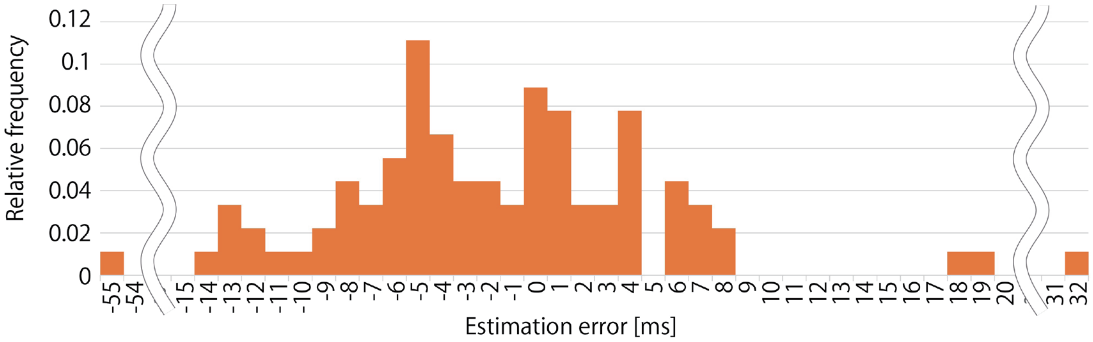

Fig. 8. (Color online) Histogram of error of estimated release point of dart throws.

4.4 Discussion

We evaluated the performance of the proposed system through experiments where 10 subjects attached a wireless sensor to their wrist and performed baseball pitching, basketball free throws, and dart throws. The time resolution of the human eye is about 50 ms(24). From this point of view, the error of 50 ms could be one of the requirements. For baseball pitching, all the estimation errors were within ±50 ms and the best results were obtained among the three gestures. For basketball free throws, 96% of errors were within ±50 ms and, for dart throws, 99% of errors were within ±50 ms.

The reason that the error of the estimated release point of basketball free throws was highest was considered to be that the subjects who had experience of basketball performed more varied throws; they flexibly adjusted their way of throwing according to the angle and distance to the goal. In addition, the ball size may have also affected the results, i.e., a basketball is larger than a baseball, meaning that it takes more time for the ball to be released from the fingers. Since the sensor is attached to the wrist, the sensor data hardly changes when the ball is released from the fingers. When estimating the basketball release point, sensor data at the release point in the training data matched with the number of data points in the testing data that was larger than that of a baseball, meaning that the release point was not estimated correctly.

Furthermore, when the proposed method estimates the time of the release point, not the frame immediately before the fingers completely separate from the ball, but the frame where the fingers begin to separate from the ball was often chosen. It is probable that the histograms of the release point estimation error were negatively biased.

In addition, the reason for the estimation accuracy of darts being lower than that of baseball is considered to be that the movement of darts released from the pinched state is less stable than that of a baseball. However, since darts are smaller than a basketball, the time for darts to separate from the fingers is shorter, so it is considered that the estimation error is smaller than that for a free throw motion.

4.5 Limitations

Lastly, we describe the limitations of the proposed method. With regard to the threshold α, in the evaluation experiments, α was determined for each gesture and subject from the training data, and these values were different for each gesture and user, so the reusability of α was low. We assume that there are several ways of collecting training data, such as using a ball with a sensor, or using a remote for video game to throw a ball while holding down the button and releasing the button at the moment when the ball is released. If it is not possible to collect training data with motion occurrence times from the user himself/herself, we can collect training data with motion occurrence times from multiple users for each gesture in advance and use a general value of α, although the estimation error would be large.

In the evaluation, the models were built for each subject and several subjects showed a large maximum error. This is an issue of stability of the subject's movements, as mentioned for basketball. If there is a large difference between the training and testing data, the estimation error becomes large. We have confirmed that this is a limitation of the proposed method because the gesture recognition is generally erroneous if the training data and testing data come from different users.(23) We need to increase the number of training data to cope with this problem. It is also possible to reduce the output with a large error by setting a strict threshold value; however, in this case, even if the output has a small error, it will be treated as undecidable (UNIDENTIFIED) and recall will be reduced.

With respect to the types of motion that can be detected by the proposed method, this paper focused on a characteristic moment, i.e., the release point, in baseball, basketball, and darts. The proposed method can estimate the time if it is a unique moment of action that occurs during a gesture. However, we have not been able to determine the extent to which the proposed method is applicable if there is no unique moment. For example, it is theoretically difficult to estimate the time at a single point in a period of time during a stationary state using the current proposed method. The types of behavior that can be estimated by the proposed method should be verified as future work.

5. Conclusion

In this paper, we proposed a method of estimating the time of a moment when a particular action occurs during a gesture. A motion sensor is attached to the user's wrist that measures the acceleration and angular velocity during the movement, and estimates the time when a specific movement occurs. We estimated the release points for three types of movement, baseball pitching, basketball free throws, and dart throws, using the proposed method, and assuming that the estimation error is less than the sampling interval of the sensor, the estimation accuracy was determined to be 61.9% for baseball, 21.0% for basketball, and 20.0% for darts. The percentage of release point estimation errors below ±12 ms was obtained to be 100% for baseball, 87.6% for basketball, and 91.1% for darts.

In the future, we will focus on wrist movements to extract specific movements in a gesture and evaluate the estimation accuracy. Furthermore, we will propose a method of detecting multiple specific actions in a gesture, which will expand the range of gesture recognition.

Acknowledgments

This research was funded by Japan Science and Technology Agency, PRESTO (grant number JPMJPR1937, 369) and by Japan Science and Technology Agency, CREST (grant number JPMJCR18A3).

References

1 Moff, Inc.: http://www.moff.mobi/ (accessed June 2020).

2 C. Myers and L. Rabiner: Bell Syst. Tech. J. **60** (1981) 1389. https://doi.org/10.1002/j.1538-7305.1981.tb00272.x

3 Wii Sports: https://www.nintendo.co.jp/wii/rspj/ (accessed June 2020).

4 H. Yamada; K. Murao, T. Terada, and M. Tsukamoto: J. Info. Proc. **26** (2018) 38. https://doi.org/10.2197/ipsjjip.26.38

5 M. Lapinski, E. Berkson, T. Gill, M. Reinold, and J. A. Paradiso: Proc. 13th Int. Symp. Wearable Computers (ISWC 2009) pp. 131–138. https://doi.org/10.1109/ISWC.2009.27

6 C. Ladha, N.Y. Hammerla, P. Olivier, and T. Plötz: Proc. 15th Int. Conf. Ubiquitous Computing (UbiComp 2013) pp. 235–244. https://doi.org/10.1145/2493432.2493492

7 F. Kosmalla, F. Daiber, and A. Krüger: Proc. 33rd Annu. ACM Conf. Human Factors in Computing Systems (CHI 2015) pp. 2033–2042. https://doi.org/10.1145/2702123.2702311

8 M. Bächlin, K. Förster, and G. Tröster: Proc. 11th Int. Conf. Ubiquitous Computing (UbiComp 2009) pp. 215–224. https://doi.org/10.1145/1620545.1620578

9 B. Zhou, H Koerger, M. Wirth, C. Zwick, C. Martindale, H. Cruz, B. Eskofier, and P. Lukowicz: Proc. 20th Int. Symp. Wearable Computers (ISWC 2016) pp. 64–71. https://doi.org/10.1145/2971763.2971784

10 D. Connaghan, P. Kelly, N. O'Connor, M. Gaffney, M. Walsh, and C. O'Mathuna: Proc. IEEE Sensors 2011 Conf. (2011) pp. 1437–1440. https://doi.org/10.1109/ICSENS.2011.6127084

11 P. Blank, T. Kautz, and B. M. Eskofier: Proc. 20th Int. Symp. Wearable Computers (ISWC 2016) pp. 72–79. https://doi.org/10.1145/2971763.2971778

12 P. Blank, J. Hoßbach, D. Schuldhaus, B. M. Eskofier: Proc. 19th Int. Symp. Wearable Computers (ISWC 2015) pp. 93–100. https://doi.org/10.1145/2802083.2802087

13 H. Kanke, Y. Takegawa, T. Terada, and T. Tsukamoto: Airstic Drum: Proc. 9th Int. Conf. Advances in Computer Entertainment Technology (ACE 2012) pp. 57–69. https://doi.org/10.1145/2802083.2802087

14 K. Murao and T. Terada: Proc. IEEE Int. Symp. Wearable Computers (ISWC 2010) pp. 69–72. https://doi.org/10.1109/ISWC.2010.5665870

15 M. Yoshizawa, W. Takasaki, and R. Ohmura: Proc. Int. Workshop Human Activity Sensing Corpus and Its Application (HASCA2013) pp. 653–664. https://doi.org/10.1145/2494091.2495986

16 Wireless Technologies Inc.: http://www.wireless-t.jp/ (accessed June 2020).

17 WEKA: http://www.cs.waikato.ac.nz/ml/weka/ (accessed June 2020).

18 M. Lee, K. Kim, M. Ryu, and J. Kim: Sens. Mater. **28** (2016) 655. https://doi.org/10.18494/SAM.2016.1319

19 E. H. Chi, J. Song, and G. Corbin: Proc. 17th Annu. ACM Symp. User Interface Software and Technology (UIST 2004) pp. 277–285. https://doi.org/10.1145/1029632.1029680

20 R. J. N. Helmera, A. G. Hahnb, L. M. Staynesa, R. J. Denninga, A. Krajewskia, and I. Blanchonetta: Procedia Eng. **2** (2010) 3065. https://doi.org/10.1016/j.proeng.2010.04.112

21 K. Naitoa, T. Takagi, H. Kubota, and T. Maruyama: Hum. Mov. Sci. **54** (2017) 363. https://doi.org/10.1016/j.humov.2017.05.013

22 Y. Kaizu, H. Watanabe, and T. Yamaji: J. Phys. Ther. Sci. **30** (2018) 223. https://doi.org/10.1589/jpts.30.223

23 K. Murao, T. Terada, A. Yano, and R. Matsukura: Trans. Hum. Interface Soc. **15** (2013) 281. https://doi.org/10.11185/imt.8.1154

24 F. Farzin, S. M. Rivera, and D. Whitney: Psychol. Sci. **22** (2011) 1004. https://doi.org/10.1177%2F0956797611413291

25 E. Mencarini, A. Rapp, L. Tirabeni, and M. Zancanaro: IEEE Trans. Hum. Mach. Syst. **49** (2019) 314. https://doi.org/10.1109/THMS.2019.2919702

26 M. Kos and I. Kramberger: IEEE Access **5** (2017) 6411. https://doi.org/10.1109/ACCESS.2017.2675538

27 J. C. Maglott, J. Xu, and P. B. Shull: Proc. 2017 IEEE 14th Int. Conf. Wearable and Implantable Body Sensor Networks (BSN) (2017). https://doi.org/10.1109/BSN.2017.7936000

28 K. Lian, W. Hsu, D. Balram, and C. Lee: Sensors **20** (2020) 1344. https://doi.org/10.3390%2Fs20051344

29 M. Kim and S. Park: Sensors **20** (2020) 4466. https://doi.org/10.3390%2Fs20164466

About the Authors

Kazuya Murao received his B.Eng., M.Info.Sci, and Ph.D. degrees from Osaka University, Japan, in 2006, 2008, and 2010, respectively. From 2011 to 2014, he was an assistant professor at Kobe University, Japan. From 2014, he was an assistant professor at Ritsumeikan University, Japan. Since 2014, he has been an associate professor at Ritsumeikan University. He is also concurrently a JST PRESTO researcher since 2019. His research interests are in wearable computing and human activity recognition. He is a member of IEEE and ACM. (murao@ritsumei.ac.jp)

Hiroshi Yamada received his B.Eng. and M.Eng degrees from Kobe University, Japan, in 2015 and 2017, respectively. His research interests are in wearable computing and ubiquitous computing.

Tsutomu Terada received his B.Eng., M.Eng., and Ph.D. degrees from Osaka University, Japan, in 1997, 1999, and 2003, respectively. He became an assistant professor at Cybermedia Center of Osaka University and a lecturer in 2000 and 2005, respectively. He is an associate professor at the Graduate School of Engineering, Kobe University. He is currently investigating wearable computing, ubiquitous computing, and entertainment computing. He is a member of IEEE, ACM, and IEICE. (tsutomu@eedept.kobe-u.ac.jp)

Masahiko Tsukamoto received his B.Eng., M.Eng., and Ph.D. degrees from Kyoto University, Japan, in 1987, 1989, and 1994, respectively. From 1989 to 1995, he was a research engineer of Sharp Corporation. From 1995 to 1996, he was an assistant professor at the Department of Information Systems Engineering, Osaka University, and from 1996 to 2004, he has been an associate professor at the same department. He is currently a professor at the Graduate School of Engineering, Kobe University. He is currently investigating wearable computing and ubiquitous computing. He is a member of eight academic societies, including ACM and IEEE. (tuka@kobe-u.ac.jp)