Regular Paper

# A Combined-activity Recognition Method with Accelerometers

Kazuya Murao[1,a) ] Tsutomu Terada[2,3,b)]

Received: August 26, 2015, Accepted: February 8, 2016

**Abstract:** Many activity recognition systems using accelerometers have been proposed. Activities that have been recognized are single activities which can be expressed with one verb, such as sitting, walking, holding a mobile phone, and throwing a ball. In fact, combined activities that include more than two kinds of state and movement are often taking place. Focusing on hand gestures, they are performed not only while standing, but also while walking and sitting. Though the simplest way to recognize such combined activities is to construct the recognition models for all the possible combinations of the activities, the number of combinations becomes immense. In this paper, firstly we propose a method that classifies activities into postures (e.g., sitting), behaviors (e.g., walking), and gestures (e.g., a punch) by using the autocorrelation of the acceleration values. Postures and behaviors are states lasting for a certain length of time. Gestures, however, are sporadic or once-off actions. It has been a challenging task to find gestures buried in other activities. Then, by utilizing the technique, we propose a recognition method for combined activities by learning single activities only. Evaluation results confirmed that our proposed method achieved 0.84 recall and 0.86 precision, which is comparable to the method that had learned all the combined activities.

**Keywords:** activity recognition, combined activities, wearable computing, accelerometers

## 1. Introduction

Along with the progress in wearable computing, many activity recognition systems with various kinds of sensors have recently been introduced, such as systems with electromyographs [16], electrocardiograms [17], Galvnanic skin response (GSR) [18], and manually configured devices [19]. Activity recognition systems are applied to many services i.e., health care [18], recognition of workers' routine activities [5], and support during assembly and maintenance tasks [20]. A health-care system [18] recognizes daily activities such as eating, walking, and working in real time by using a heat sensor, GSR sensor, accelerometer, electric sphygmograph, and geomagnetic sensor. The system alerts lack of exercise and occupational fatigue, and advises the user on how to make improvements in the user's life.

Camera, GPS, gyroscope, and geomagnetic sensor are also known as devices that detect/obtain location and motions, however these sensors have low wearability and accuracy, and cannot obtain both motion and static direction simultaneously. An accelerometer can obtain motion and static direction by sensing earth's gravity, and has high accuracy and high resolution, and is small enough to be attached on the body. For this reason, an accelerometer is the best device for activity recognition.

In the process of making an activity recognition system, developers generally define the objective activities, collect their data, annotate them, and construct the recognition models. Therefore, recognition results are limited within the predefined activities. In other words, if we want to recognize the combined activity *holding a mobile phone while walking* [*1], training data for the activity has to be collected and annotated beforehand. Otherwise, the recognition result would be either *walking* or *holding a mobile phone*.

Here, we define two types of activities; global activity and local activity. Global activity is a bodywide movement, such as *walking* and *standing*. Multiple global activities can contradict each other at a time. Local activity is a movement of a specific part of the body, such as *throwing* and *holding something*. Multiple local activities coexist unless they are on the same body part. The simplest way to recognize combined activity is to construct recognition models for all the combined activity. However, this is difficult since the number of possible combinations of global and local activities can get immense. Suppose there are five global activities; *standing*, *sitting*, *walking*, *running*, and *cycling*, and ten local activities of hand gesture. Then, data for 50 patterns must be collected. As one kind of hand gesture is added, data for the gesture performed during five global activities must also be captured, which is a time-consuming task. Moreover, if we also consider foot gestures, the number of possible patterns is (# global activity) × (# local hand activity) × (# local foot activity). However, these combined activities are not negligible since they are physically possible and may occur in our daily lives.

[1] College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Shiga 525–8577, Japan
[2] Graduate School of Engineering, Kobe University, Kobe, Hyogo 567–8501, Japan
[3] Japan Science and Technology Agency, PRESTO, Kawaguchi, Saitama 332–0012, Japan
[a)] murao@cs.ritsumei.ac.jp
[b)] tsutomu@eedept.kobe-u.ac.jp

[*1] Names of activities are denoted in italics in this paper.

The problem of combined-activity recognition is not only the time needed for collecting training data, but also annotating the data as well. After collecting all the data through long experiments, the data has to be annotated with ground truth. **Figure 1** shows the acceleration of a chop gesture while running and while standing. It is easy to find the starting and end points for *chop while standing*, whereas it is harder to trim *chop while running* through visual inspection. Someone might think that recognizing the activity of each part of the body individually and integrating them produce the correct activity. However, as shown in Fig. 1, the gesture while running is different to the same gesture while standing. Moreover, the arm swing while running is included in the beginning and ending of the gesture unless the gesture is trimmed with a tightfitting window. Since the motion of running is propagated, the gesturing is slightly different, and part of the running motion is included before and after the gesture, which leads to misrecognition.

Our previous study worked on gesture recognition while moving. Recognized gestures in the research literature are not combined ones, but single ones which change smoothly from/to a global activity, such as *walking* [6]. An effective method to treat combined activity has not been reported.

The contribution of this paper is to recognize combined activities from training data of single activities, i.e., global activities and local activities while stationary. The proposed system classifies each part of the body according to the categories posture, behavior, and gesture, from the fluctuations and autocorrelations in the acceleration data. In this paper, we define posture, behavior, and gesture as follows.

- Posture: State of a user remaining stationary lasting for a certain length of time, e.g., *sitting* and *standing*.
- Behavior: State of a user doing periodical movement lasting for a certain length of time, e.g., *walking* and *running*.
- Gesture: Not a state but a once-off action having a starting point and an endpoint that sporadically occurs, e.g., *punch* and *draw a circle in the air*.

Postures can easily be detected since these are static and do not produce fluctuations in acceleration values. In general, autocorrelation plots of a periodical wave show high peaks. When parts of the sensors are showing high peaks, the corresponding body parts are meant to have constancy and are classified as behavior, otherwise the parts are classified as gesture. The proposed method then recognizes activities according to the activity type, which means that posture is recognized with a classifier that has learned postures only, and behavior and gesture are recognized as well. Finally, the system outputs a conclusive recognition result from the recognition results of each body part. Simply recognizing an activity for each body part and combining them cannot recognize the combined activity since the body part of a global+gesture (e.g., *running+punch*) activity would be recognized as a global activity (e.g., *running*). By using our system, combined activities such as *throw while walking* and *holding a mobile phone while running* are recognized only from single activities; *walking*, *running*, *throwing*, and *holding a mobile phone*, which is the main contribution of this paper compared with our previous work in Ref. [6].

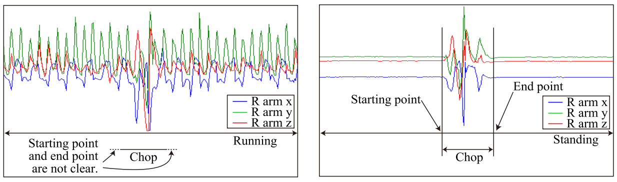

**Fig. 1** Waveform of an accelerometer mounted on the right wrist: a chop gesture while running (left) and while standing (right).

This paper is organized as follows. Section 2 introduces related works and Section 3 describes the system structure. The performance of our system is discussed in Section 4. Finally, Section 5 concludes this paper.

## 2. Related Work

Studies on activity recognition are listed in **Table 1**, however most of them focus on single activities, such as ambulation and posture.

One study recognizes eight activities; *sitting*, *standing*, *walking*, *walking up stairs*, *walking down stairs*, *riding elevator down*, *riding elevator up*, and *brushing teeth*, with a multimodal sensor board (MSB) that has seven kinds of sensors such as a microphone and an accelerometer [3]. The method proposed by Ravi et al. [7] recognizes eight activities including *vacuuming* and *brushing teeth* with an accelerometer attached to the pelvic region. Naya et al. [5] proposed a workers' routine activity recognition system in order to support their daily work. This system recognizes a nurse's activities such as *drip injection* and *vital check*. Another study employs twenty-two kinds of sensors to recognize *lying*, *rowing*, *running*, *Nordic walking*, *cycling*, *walking*, *sitting*, and *standing* [2]. Bao et al. [1] recognizes twelve activities including *walking while carrying items* and *folding laundry* with five biaxial accelerometers. Though the *walking while carrying items* activity seems like a combined activity, it cannot be separated and recognized by combining with other activities, i.e., the *running while carrying items* activity and the *holding items* activity cannot be recognized according to current work in the literature.

The following studies are focusing on local activities using parts of the body. The method proposed by Graeme et al. [4] annotates video-recorded activities by gesture recognition using one accelerometer mounted on the wrist since annotating video is difficult when only analyzing video. It uses a hidden Markov model (HMM) [14] for the recognition, resulting in one mistake in 30 trials for three kung-fu martial art movements; *cut*, *elbow*, and *punch*. The Georgia Tech Gesture Toolkit [8] is a tool for supporting gesture recognition that has been proposed by Westeyn et al. This is a toolkit that enables ordinary users who do not have enough knowledge about speech recognition to use the existing HMMtoolkit [9] with ease. In the literature, four applications are presented, one is gesture recognition with two 3-axis accelerometers positioned at the wrist and elbow, and achieves 93.3% accuracy for ten kinds of gestures such as *grinding*, *sawing*, and *screwing*. The system proposed by Junker et al. [10] recognizes ten short daily actions, such as *pushing a button* and *drinking*, and achieves approximately 80% precision and recall. The innovative point of this study is that it partitions the stream of sensor

**Table 1** Activities recognized and sensors used in past work on activity recognition.

| Ref. | Activities recognized | # of sensor | Kinds of sensors | Sensor positions |
|---|---|---|---|---|
| [1] | ambulation, posture, scrubbing, vacuuming, folding laundry, brushing teeth, cycling, working on computer, eating or drinking, walking carrying items | 5 | 2D acc×5 | left elbow, right wrist, torso, left knee, right ankle |
| [2] | ambulation, posture, putting clothes on, eating, rowing, cycling, respiratory | 22 | air pressure, ball sensor, switch, GPS, temperature, respiratory, heart rate×3, skin resistance, skin temperature, 3D compass×2 3D acc×2, light, humidity, SaO2×2, pulse, mic, EKG | wrist, upper back, below neck, finger, armpit, chest, forehead, shoulder |
| [3] | ambulation, posture, brushing teeth | 3 (All in one board) | light×3, temperature×3, 2D compass×3, humidity×3, 3D acc×3, barometer×3, mic×3 | shoulder, waist, wrist |
| [5] | ambulation, bed bath, carry patient, carry wheelchair, drip injection, vital check | 4 | acc×3, IR transmitter | head, chest, rear waist, upper arm |
| [7] | ambulation, posture, sit-ups, vacuuming, brushing teeth | 1 | acc | pelvic region |
| [4] | cut, elbow, punch | 1 | acc | wrist |
| [8] | hammer, file, sand, saw, screw, vise, drill, clap, use driver, grind | 2 | acc×2 | wrist, elbow |
| [10] | push button, handshake, phone up, phone down, door, coin, cutlery, drink, spoon, handheld | 5 | gyro×5 | wrists, upper arms, upper torso |
| [12] | draw '>' mark, square, shift left to right, shift right to left, shift bottom to up, shift up to bottom, clockwise circle, counter-clockwise circle | 1 | wii remote | hand |

into several segments that represent atomic human movement by using the sliding-window and bottom-up (SWAB) algorithm [11]. The method proposed by Liu et al. recognizes eight gestures such as *drawing a line* and *drawing a circle*, which are recommended by the Nokia laboratory, with one 3-axis accelerometer [12]. This research captured more than 4,000 samples from eight test subjects for a long period. They used the Dynamic Time Warping algorithm (DTW) [13] as a recognition algorithm and achieved 98.6% accuracy by successively renewing the training data.

Activities that have been targeted in these works are single activities. Combined activities have to be defined one by one. However, the number of possible combinations increases with the number of global and local activities, causing much time to capture the training data. Capturing ten global activities and ten 2-second local gestures for five times takes 1,000 seconds. This is the actual movement time and more than ten times this is needed for the interval and rest. In addition, the effect of fatigue is not negligible since gesture motion changes and decays through multiple iterations of gestures [15].

The following two works handle combined activities. Park et al. [27] proposes a gesture recognition system with a hand-worn sensor and mobile device. This system segments a gesture with a two-stage threshold-based filter with an accelerometer and a gyroscope. A novel point of the paper is to automatically adjust the threshold according to four mobility situations; RIDE, STAND, WALK, and RUN, based on the proportion of filtered data. Then features are extracted from the segment and recognized with an HMM. The HMM that they proposed is a multi-situation HMM, which changes the models according to the mobility situation. However, the models are trained with all combinations of gestures and mobility situations.

A system proposed by Korpela et al. [28] recognizes activities including gestures by considering the accuracy and power consumption. Activities that requires light processing are recognized on the wearable device and the output label is sent to a smartphone, while the data that requires heavy processing, i.e., gestures, are sent to the smartphone and recognized. This system does not segment data but uses a sliding window approach. Though they assume that gestures are conducted while standing, the data of gestures while walking was used in additional experiments in the paper. However, the latter experiments just examine the performance of gesture recognition while walking and does not recognize them as a combined activity.

## 3. Proposed System

### 3.1 System Structure

Posture and behavior are states lasting for a certain period of time and consist of periodic patterns of acceleration waveform. These activities are generally recognized with a classifier such as SVM [25] after converting raw data into feature values such as mean, variance, and fast Fourier transform (FFT) coefficient over a time window. This approach enables high-speed recognition since not all data in the window but only feature values are used. Moreover, one of the advantages of using feature values is that the recognition process does not have to consider which part of the movement is included in the window, e.g., beginning of the window does not have to fit the specific motion of steps of walking, since feature values discard temporal information.

On the other hand, a gesture is a once-off action that has a starting point and an endpoint, which is different from postures and behaviors. Feature-based approach hardly distinguishes similar gestures such as *rotating arm clockwise* and *rotating arm counter-clockwise* since general feature values do not have information on how it moved, therefore gestures are recognized in a different way. In general, gestures are recognized with a template matching algorithm such as dynamic time warping (DTW) or a statistical model such as hidden Markov models (HMM), after trimming actual movement from stream data. The conventional method forced users to indicate a gesture interval by pushing a button on a device or by standing still before and after the gesture [21]. It is hard to stop motions or perform specific gestures to indicate a starting point. Taking out a device or continually holding a device to push a button while performing gestures is also unrealistic. By recognizing a gesture without considering a starting point and an endpoint, misrecognition can occur or the gesture can be missed out entirely and taken to be a behavior. Our system classifies user activities into posture, behavior, and gesture for each body part, then applies DTW to the gesture or applies SVM to the posture and behavior.

Our system consists of three phases, as shown in **Fig. 2**. The first phase classifies activities of each part of the body into three types; posture, behavior, and gesture. The second phase recognizes activities according to the activity type. The third phase integrates the recognition results and outputs a conclusive result.

In this paper, we assume that a user attaches five accelerometers on both wrists, the hip, and both ankles. Activities are four postures (*sitting*, *standing*, *lying*, and *kneeling*), five behaviors (*walking*, *running*, *cycling*, *descending stairs*, and *ascending stairs*), five hand-gestures (*chop*, *throw*, *punch*, *draw a clockwise circle*, and *draw a counter-clockwise circle*), and two hand-postures (*holding a mobile phone*, and *raising a hand*). The sampling frequency is 20 Hz, which is sufficient for activity recognition as previously reported [22].

### 3.2 Activity Classification

The activity classification phase consists of movement detection that classifies the user state into posture or movement, and constancy decision that classifies the user movement to behavior or gesture.

#### 3.2.1 Movement Detection

The activity classification phase checks for movements in the sensed data. Suppose the current time is $t = T$. Let the moving average over 20-sample (1-second) sensed data be $\overline{x}(T)$ and the current value be $x(T)$, then our system detects a movement according to the following equation. Otherwise, our system judges that the user is maintaining a posture.

$$\begin{array}{lcl} \text{if } |x(T) - \overline{x}(T)| > \epsilon(T) & \Rightarrow & \text{Behavior or Gesture} \\ \quad\text{otherwise} & \Rightarrow & \text{Posture} \end{array} \tag{1}$$

The region of $x(t) \pm \varepsilon(T)$ is called the epsilon tube, which removes movements. In this paper, $\varepsilon(T)$ is set as follows:

$$\varepsilon(T) = \max \left\{ \begin{array}{l} \dfrac{\sigma_{R.leg}(T) + \sigma_{L.leg}(T)}{2} \\ 200 \end{array} \right. \tag{2}$$

where

$$\sigma_{R.leg}(T) = \sum_{t=T-19}^{T} \{\overline{a}_{R.leg}(T) - a_{R.leg}(t)\} \tag{3}$$

$$\sigma_{L.leg}(T) = \sum_{t=T-19}^{T} \{\overline{a}_{L.leg}(T) - a_{L.leg}(t)\} \tag{4}$$

$$a_{R.leg}(t) = \sqrt{x^2_{R.leg}(t) + y^2_{R.leg}(t) + z^2_{R.leg}(t)} \tag{5}$$

$$a_{L.leg}(t) = \sqrt{x^2_{L.leg}(t) + y^2_{L.leg}(t) + z^2_{L.leg}(t)} \tag{6}$$

$$\overline{a}_{R.leg}(T) = \sum_{t=T-19}^{T} a_{R.leg}(t) \tag{7}$$

$$\overline{a}_{L.leg}(T) = \sum_{t=T-19}^{T} a_{L.leg}(t) \tag{8}$$

$x_{R.leg}(t)$, $y_{R.leg}(t)$, $z_{R.leg}(t)$, $x_{L.leg}(t)$, $y_{L.leg}(t)$, and $z_{L.leg}(t)$ are acceleration values of x-axis, y-axis, and z-axis at time $t$ for the sensor on the right leg and the left leg, respectively. The vibration of the movement of legs is propagated to other regions such as the hand. Therefore $\varepsilon(T)$ is set in response to intensity of legs. While the movement of legs is not intense, $\varepsilon(T)$ is set to 200 mG since the fluctuation produced while being stationary was up to 100 mG.

Since the current value $x(t)$ might temporarily go into the epsilon tube even while moving, the posture begins only after $x(t)$ stays within the epsilon tube for more than 0.25 seconds. These values are obtained from our pilot studies. As shown in **Fig. 3**, while the data is within the epsilon tube, the system judges that the parts of the body are maintaining a posture. When the data indicates movement, this process goes into a constancy decision phase.

For a 3-axis accelerometer, the movement detection is applied to each axis of a sensor. If all three axes of the sensor are judged as posture, the activity of the body part is classified as posture, otherwise the activity is classified as behavior or gesture.

#### 3.2.2 Constancy Decision

Basically, data on walking includes iterations in rhythm with steps. On the other hand, gestures are once-off actions and do not have iterations. Note that we consider the iterations of once-off actions as behaviors.

In this phase, the autocorrelation function (ACF) finds iterations in the user's movements, and classifies the movement into behavior or gesture. The discrete ACF $Rxx(\tau)$ at lag $\tau$ for a data sequence $x(t)$ is defined as

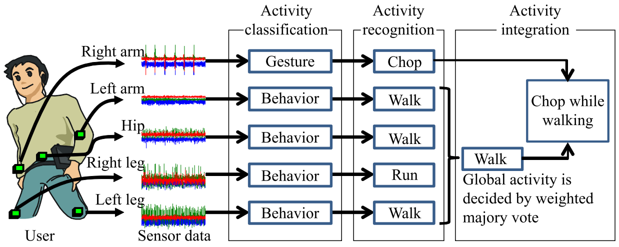

**Fig. 2** Recognition flow of the system for combined activities.

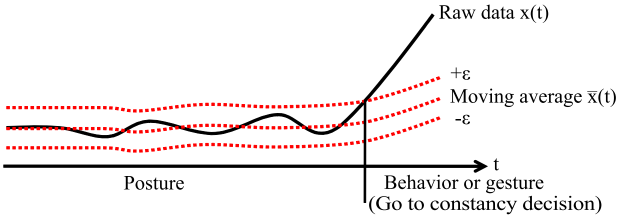

**Fig. 3** Movement detection.

$$Rxx(\tau) = \sum_{t=0}^{N-1} x(t)x(t-\tau), \tag{9}$$

where $N$ is the window size for the ACF calculation and set to 64 samples (3.2 seconds), which is long enough to capture at least two iterations. In addition, since ACF shows a maximum at $\tau = 0$, all the values of ACF are normalized by the following equation so that the range is $(-1, +1)$.

$$R'xx(\tau) = Rxx(\tau)/Rxx(0) \tag{10}$$

The system has to decide whether the movement has constancy or not. **Figure 4** shows an acceleration waveform and its ACF of *walk, chop, then walk again* activity. As shown in the figure, the ACF of walking shows clear peaks, whereas the ACF of chop does not have high peaks. Constancy is detected when the height of the first peak $R'xx(n)$ $(n > 0)$ exceeds $\alpha \cdot (1 - n/N)$, where $\alpha$ is a coefficient set to 0.6 through our pilot study changes $\alpha$ from 0.5 to 1.0 at intervals of 0.1.

$$\begin{array}{lcl} \text{if } \mathrm{R}'_{\mathrm{xx}}(\mathrm{n}) \geq \alpha \cdot (1 - \mathrm{n}/\mathrm{N}) & \Rightarrow & \text{Behavior} \\ \qquad \text{otherwise} & \Rightarrow & \text{Gesture} \end{array} \tag{11}$$

The reason $n/N$ is used here is that the height of the first peak linearly decreases as $\tau$ increases. Since the constancy hardly appears with large $\alpha$, the ratio of behavior which is misrecognized as gesture increases, whereas, the constancy is apt to appear with small $\alpha$. Therefore the ratio of gesture which is misrecognized as behavior increases.

For a 3-axis accelerometer, the constancy decision is applied to each axis of a sensor. If no axis of the sensor is judged as behavior, the activity of the sensor is classified as gesture; otherwise the activity is classified as behavior.

Intille et al. focused on acquiring in-situ training data and mentioned that acceleration data of walking in the laboratory displays consistent gait cycle. On the other hand, acceleration data of the same person outside the laboratory may display marked fluctuation in gait cycle and length [24]. We think that this observation is correct and the data obtained from the same subject on different days are different. Although acceleration data fluctuate in the range of day or hour, data in the range of few seconds or shorter are significantly periodic, which could produce constancy. Moreover, since our approach is unsupervised and does not require training, influence of differences among individuals and users' conditions are small.

### 3.3 Activity Recognition

The activity recognition phase identifies activities according to the activity types. For posture data, the mean value of the data in the window is calculated as a feature value and the posture is recognized with an SVM that has learned only postures. Since the variance of postures is almost zero, only the mean is used for the recognition. SVM operating on the mean and variance is used for behavior data, whereas DTW operating on the trimmed original wave over a window is used for recognizing gestures. SVM and DTW has learned behaviors only and gestures only, respectively. This section briefly explains both recognition algorithms.

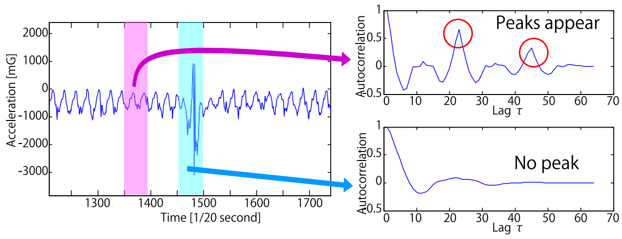

**Fig. 4** Accelerations of the *chop while walking* (left) and autocorrelation of *walking* (upper right) and *chop* (lower right).

#### 3.3.1 Support Vector Machine

SVM is a classification algorithm that often provides competitive or superior accuracy for a large variety of real-world classification tasks [25]. Consider the problem of separating a set of training data $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_J, y_J)$ into two classes, where $\mathbf{x}_i \in R^N$ is a feature vector and $y_i \in \{-1, +1\}$ is its class label. Assuming that the classes can be separated by the hyperplane, $\mathbf{w} * \mathbf{x}_i + b$, and no knowledge about the data distribution is given beforehand, the optimal hyperplane is the one with the maximum distance to the closest points in the training dataset. We can find the optimal values for $\mathbf{w}$ and $\mathbf{b}$ by solving

$$\begin{array}{l} \min \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to } \; y_i(\mathbf{w} * \mathbf{x}_i + b) \geq 1, \; \forall i = 1, \cdots, n. \end{array} \tag{12}$$

The multiplication factor 1/2 is used for mathematical convenience. By using Lagrange multipliers, $\lambda_i (i = 1, \cdots, n)$, this can be rewritten as:

$$\begin{array}{l} \max \; \sum_{i=1}^{N} \lambda_i - \sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \\ \text{subject to } \; \sum_{i=1}^{N} y_i \alpha_i = 0, \; \lambda_i \geq 0, \end{array} \tag{13}$$

and the results in the classification function are

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{n} \lambda_i y_i \mathbf{x}_i * \mathbf{x} + b\right). \tag{14}$$

The extension of a 2-class SVM to an N-class SVM can be achieved by training N SVMs such that one class will be separated from the others.

#### 3.3.2 Dynamic Time Warping

Time-series data has been used in various fields such as science, medicine, economics, and engineering. Calculation of similarity between time-series data is required in order to data-mine in these fields. Though the simple method to measure similarity is Euclidean distance, its drawbacks include susceptibility to temporal distortion and the number of samples of two data sequences must be equal.

DTW is an algorithm for measuring similarity of two time-series data, which redeems the drawbacks of Euclidean distance. Advantages of DTW include the ability to calculate temporal non-linear elastic distance, the similarity between two sequences which may vary in time or speed, and the number of both samples that are not equal in size. For example, DTW can find similarities in situations where two kinds of data of *draw a circle in the air* whose rotating speed are different are compared. In addition, in case that a part of each data differs, DTW is applicable because of its non-linear elasticity.

The detailed algorithm is as follows. When two time-series gesture data $X = (x_1, \cdots, x_m)$ and $Y = (y_1, \cdots, y_n)$ are compared, whose lengths are $m$ and $n$, respectively, an $m \times n$ matrix

**d** is defined by $\mathbf{d}(i, j) = |x_i - y_j|$. Subsequently, a warping path $W = (w_1, \cdots, w_k)$ is found, which is a path of pairs of indices of $X$ and $Y$. At that time, pass $W$ meets the following three conditions.

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

So as to find the path with the lowest cost that satisfies the above conditions, the following steps are applied.

```
(1) Initialization:
      $DTW(0, 0) = 0$
      $DTW(i, 0) = \infty$ for $i = 1, 2, \cdots, m$
      $DTW(0, j) = \infty$ for $j = 1, 2, \cdots, n$
(2) Do for $i = 1, 2, \cdots, m$:
      Do for $j = 1, 2, \cdots, n$:
        $DTW(i, j) = d(i, j) + min \begin{cases} DTW(i-1, j-1) \\ DTW(i-1, j) \\ DTW(i, j-1) \end{cases}$
(3) Output:
      Return $DTW(m, n)/n$
```

The obtained cost $DTW(m, n)$ becomes a distance between $X$ and $Y$. The returned $DTW(m, n)$ is divided by $n$ since the DTW distance increases with the length of the training data. The system recognizes gestures on the basis of the training data. The distances for all the training data are calculated, and the training data with the shortest distance is identified. A gesture labelled with the training data is then output. The DTW algorithm can be used for multiple axes of an accelerometer. The DTW calculation is carried out for each axis, and the sum of the distances for all axes is used as the distance of the gesture.

### 3.4 Activity Integration

Since each part of the body outputs activity recognition result individually, these have to be integrated in order to determine the conclusive recognition result. Even if the user is just walking, recognition results of all parts may not be *walking*.

In this paper, we assume that if the recognition result for a hand is posture, hand-posture or hand-gesture, it will be a local activity, otherwise local activity will be null. Then global activity is decided by a weighted majority vote. We used the recall of the recognition result for training data as a weight. This is because behavior data are partly misclassified as the gesture, which does not affect decreases in the number of false positives within the behavior type. Therefore, precision and F-measure are undefinable and inappropriate in this case, and recall reflects the results of both activity classification and activity recognition. If each body part has one vote, then the weighted votes are summed up over the body except for local activity. For example, in Fig. 2, the recognition results of the left arm, hip, right leg, and left leg are *walk*, *walk*, *run*, and *walk*, respectively. Suppose the recall of *walk* recognized with a sensor on the left arm, hip, and left leg are 0.7, 0.8, 0.4, respectively, and the recall of *run* recognized with a sensor on the right leg is 0.4, the number of votes for *walk* is 1.9 and the number of votes for *run* is 0.4, resulting in walking as the global activity. Finally, the combined activity is output by merging the global and local activities.

Even if a part of the body is doing an activity that is not a global activity, the user activity can still be correctly recognized. For the *holding a mobile phone while walking* activity, the hand holding a phone is classified into posture and recognized as *holding a phone*, whereas the remaining parts would be classified into behavior and recognized as *walking*.

## 4. Evaluation

In this section, we evaluate our system on the basis of recall, precision, and processing time.

### 4.1 Preliminary Experiment

Before evaluating activity recognition, we conducted a preliminary experiment to examine the constancy decision. Data on the *jump while walking* activity were taken from one subject, who wore three accelerometers [23] on the right wrist, hip, and right ankle. The sampling frequency was 20 Hz.

**Figure 5** shows the results. The horizontal axis and vertical axis indicates time and acceleration, respectively. The marks on the line of 6,000 mG show the results of constancy decision; the + mark shows "constancy" and the ● mark shows "no constancy". For the figure, constancy appears while *walking*, and constancy does not appear for all four *jump*s. Though several points of *walking* are judged as inconstant, these false positives can be omitted with a filter so that four-consequent constancy is meant to be a gesture. This is because the window of constancy decision is slid by 16 samples which is 1/4 of the window. Therefore one gesture is included in a window at least four times. Motions that do not have iterations like gesture produce inconstancy four consequent times, while a small decay in behavior produces inconstancy once or twice, which can be neglected by the filter.

We examined the performance of the filter for the data of behavior without any gestures. **Table 2** shows the ratio of the number of inconstancy for the number of trials of constancy decision for five kinds of behavior: *walking*, *running*, *cycling*, *descending stairs*, and *ascending stairs*. For "without filter", gesture recogni-

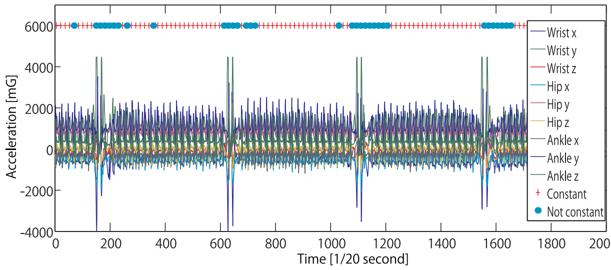

**Fig. 5** Results of constancy decision.

**Table 2** False positives of gesture detection during behavioral activities.

| Activity | Without filter (%) | With filter (%) |
|---|---|---|
| Walking | 6.8 | 0.43 |
| Running | 6.9 | 0.32 |
| Cycling | 9.0 | 0.69 |
| Ascending | 36 | 3.3 |
| Descending | 15 | 0.99 |

tion takes place as one inconstancy appears. For "with filter", the gesture recognition takes place as four consequent inconstancy appears.

For the results of "without filter", 6.8% to 36% of behaviors are classified as gesture even though the user is doing behavior only. This is because the user cannot keep a constant pace, and constancy breaks when the pace changed. In particular, the user takes the stairs two at a time or changes the pace due to fatigue while *ascending stairs* and *descending stairs*. On the other hand, the results dropped to 0.32% to 3.3% by applying the filter. We have examined with different sampling frequencies and confirmed the same tendency. With low sampling frequency ($< 10$ Hz), iterations of movement might not be captured due to coarse sampling. However, there is no problem since the performance of activity recognition would drop with such a low sampling rate [26] and activity recognition is generally carried out with higher sampling rate than 10 Hz.

### 4.2 Setup

#### 4.2.1 Data Collection

We evaluated our system for the data while *standing*, *sitting*, *walking*, *running*, and *cycling*. The training and testing data were taken from three male subjects aged 22 to 28 years, who wore five accelerometers [23] on their right wrist, left wrist, hip, right ankle, and left ankle. The sampling frequency was 20 Hz. Activities are listed in **Table 3**: four postures (*sitting*, *standing*, *lying*, and *kneeling*), five behaviors (*walking*, *running*, *cycling*, *descending stairs*, and *ascending stairs*), five hand gestures (*chop*, *throw*, *punch*, *draw a clockwise circle*, and *draw an counter-clockwise circle*), and two hand postures (*holding a mobile phone* and *raising a hand*). The subjects acted these hand gestures and hand postures while *sitting*, *standing*, *walking*, *running*, and *cycling*. Each gesture was recorded ten times for each global activity.

#### 4.2.2 Preprocessing

In SVM, raw data would generally be preprocessed by extracting the feature values to enable the meaning of sensed data to be understood. Let us assume time $t = T$, where SVM uses mean $\mu_i(T)$ and variance $\sigma_i^2(T)$ for 64 samples (3.2 seconds) of 3-dimensional sensed data $x_i(T)$ $(i = 1, 2, 3)$ retraced from time $t = T$. The mean and variance are used for behavior recognition, but only the mean is used for posture recognition. Feature vector $\mathbf{Z}(T)$ is normalized into a 6-dimensional vector $\mathbf{X}(T) = [\mu_1(T), \mu_2(T), \mu_3(T), \sigma_1^2(T), \sigma_2^2(T), \sigma_3^2(T)]$ by $\mathbf{Z}(T) = (\mathbf{X}(T) - \mathbf{M})/\mathbf{S}$, where $\mathbf{M}$ and $\mathbf{S}$ are the mean and the standard deviation of $\mathbf{X}$. The mean of $\mathbf{Z}(T)$ becomes 0 and its variance becomes 1 after this conversion.

**Table 3** Single activities.

| Kind | | Activity |
|---|---|---|
| Global | Posture | Sitting<br>Standing<br>Lying<br>Kneeling |
| | Behavior | Walking<br>Running<br>Cycling<br>Descending stairs<br>Ascending stairs |
| Local | Hand gesture | Chop<br>Throw<br>Punch<br>Draw a clockwise circle<br>Draw a counter-clockwise circle |
| | Hand posture | Hold a mobile phone<br>Raise a hand |

When recognizing gestures with DTW, 64-sample time series data is used. If the length of the gesture is less than 64 samples, the length of the gesture is used. In this paper, 3-dimensional data (1 sensor × 3 axes) were used for the DTW calculation and the label of the lowest sum of distances over all the axes becomes a result.

#### 4.2.3 Annotation and Training Data

The logged data were manually labelled, 10% of which became training data and the remaining 90% were used for testing. Training data for gestures are the gesture data recorded while standing. Note that SVM and DTW are supervised, but movement detection and constancy decision proposed in this paper are unsupervised which does not require training data nor ground truth.

#### 4.2.4 Comparison Methods

We measured recognition accuracies of combined activities by using three methods, which are SVM, DTW, and our method. As shown in **Table 4**, the first two methods are comparisons simply trained with all the possible combinations. The last one selectively uses SVM and DTW which learned single activities only and integrated the results. Most of the test data consists of global activity and gestures sporadically occur. Correct recognition results for gestures which is output in one second from the gesture ends are accounted for true positives.

### 4.3 Results

**Table 5** shows the recall and precision of the recognition for hand-gestures and hand-postures. "Null" in the column of local activity means that the subjects were performing a global activity only.

The recall and precision of gestures recognized by SVM were quite low. This is because the feature values have information on the orientation and exercise intensity but do not have information on the trajectory. Combined gestures of behavior with hand gesture or hand posture are almost misrecognized as a single behavior. Because of the same reason, the results for gestures while *standing* or *sitting* are also low. In addition, *sitting* is not correctly recognized. This is because the subjects leaned back in order not to hit the armrest when performing a gesture, resulting in the large difference to the training data for just *sitting*. Also, gesture motion while *sitting* is slower than that while *standing*, resulting in a small value of variance, and average body orientation during the gesture is similar to just *sitting*. Moreover, all of *hold mobile*

**Table 4** Proposed and comparison methods.

| Method | # of activities trained | Activities trained |
|---|---|---|
| Proposal | 16 | 4 postures<br>5 behaviors<br>5 hand gestures while standing<br>2 hand postures while standing |
| SVM | 63 | Combinations of 9 global × 7 local activities |
| DTW | 63 | Combinations of 9 global × 7 local activities |

**Table 5** Recall and precision of recognition.

| Activity | | Proposed Method | | SVM | | DTW | |
|---|---|---|---|---|---|---|---|
| Global | Local | Recall | Precision | Recall | Precision | Recall | Precision |
| Stand | Chop | 1.000 | 0.917 | 0.925 | 0.633 | 1.000 | 0.950 |
| | Throw | 1.000 | 1.000 | 0.839 | 0.942 | 0.996 | 1.000 |
| | Punch | 1.000 | 1.000 | 0.874 | 0.989 | 0.964 | 0.996 |
| | Clockwise | 1.000 | 1.000 | 0.397 | 0.426 | 1.000 | 1.000 |
| | Counter-clockwise | 1.000 | 0.882 | 0.938 | 0.661 | 1.000 | 0.920 |
| | Hold mobile | 0.996 | - | 1.000 | - | 1.000 | - |
| | Raise hand | 1.000 | - | 0.996 | - | 1.000 | - |
| | Null | 0.999 | - | 0.919 | - | 0.993 | - |
| Sit | Chop | 1.000 | 1.000 | 0.736 | 0.866 | 0.993 | 0.908 |
| | Throw | 0.955 | 1.000 | 0.840 | 0.296 | 0.842 | 0.983 |
| | Punch | 1.000 | 1.000 | 0.880 | 0.680 | 0.920 | 0.625 |
| | Clockwise | 1.000 | 1.000 | 0.831 | 0.503 | 0.805 | 0.914 |
| | Counter-clockwise | 1.000 | 0.955 | 0.855 | 0.473 | 1.000 | 0.471 |
| | Hold mobile | 1.000 | - | 1.000 | - | 1.000 | - |
| | Raise hand | 1.000 | - | 1.000 | - | 1.000 | - |
| | Null | 1.000 | - | 0.000 | - | 0.522 | - |
| Walk | Chop | 1.000 | 1.000 | 0.000 | 0.000 | 0.994 | 1.000 |
| | Throw | 1.000 | 1.000 | 0.000 | 0.000 | 0.946 | 1.000 |
| | Punch | 0.929 | 0.833 | 0.000 | 0.000 | 0.983 | 1.000 |
| | Clockwise | 0.944 | 0.944 | 0.000 | 0.000 | 0.993 | 0.995 |
| | Counter-clockwise | 1.000 | 0.944 | 0.000 | 0.000 | 0.990 | 0.990 |
| | Hold mobile | 0.705 | - | 0.975 | - | 1.000 | - |
| | Raise hand | 0.442 | - | 0.956 | - | 0.999 | - |
| | Null | 0.988 | - | 0.956 | - | 0.537 | - |
| Run | Chop | 0.917 | 0.826 | 0.000 | 0.000 | 0.986 | 1.000 |
| | Throw | 1.000 | 0.975 | 0.000 | 0.000 | 0.894 | 0.997 |
| | Punch | 0.429 | 0.429 | 0.000 | 0.000 | 0.966 | 1.000 |
| | Clockwise | 0.500 | 0.500 | 0.000 | 0.000 | 0.978 | 0.916 |
| | Counter-clockwise | 0.000 | 0.000 | 0.000 | 0.000 | 0.975 | 0.942 |
| | Hold mobile | 0.000 | - | 0.000 | - | 0.886 | - |
| | Raise hand | 0.000 | - | 0.000 | - | 0.734 | - |
| | Null | 0.981 | - | 0.875 | - | 0.309 | - |
| Bike | Chop | 0.900 | 1.000 | 0.000 | 0.000 | 0.729 | 0.956 |
| | Throw | 0.913 | 0.917 | 0.500 | 0.500 | 0.872 | 0.843 |
| | Punch | 1.000 | 0.742 | 0.000 | 0.000 | 0.930 | 0.711 |
| | Clockwise | 0.622 | 1.000 | 0.000 | 0.000 | 0.780 | 0.908 |
| | Counter-clockwise | 0.500 | 0.450 | 0.000 | 0.000 | 0.937 | 0.686 |
| | Hold mobile | 1.000 | - | 1.000 | - | 1.000 | - |
| | Raise hand | 0.999 | - | 1.000 | - | 1.000 | - |
| | Null | 0.991 | - | 0.997 | - | 0.878 | - |
| Average | | 0.843 | 0.857 | 0.510 | 0.279 | 0.908 | 0.908 |

*while running* and *raise hand while running* were misrecognized as *running* since the difference in hand orientation is absorbed by the vibration from running. From these results, it is hard for the feature-based recognition to identify a lot of combined activities.

DTW, on the contrary, had high recall and precision for all activities. It is remarkable that the performance of our proposal is comparable to that of DTW which has learned all the possible combined activities (9 global × 7 local = 63 activities), while our proposed method has learned single activities only (9 global + 7 local = 16 activities). In addition, *sitting* is correctly recognized with the proposed method even though SVM missed the recognition of *sitting* as stated above. This is because of the majority vote of each body part in the activity integration phase. In this case, the body posture as a whole was not close to the training data of *sitting*, but some body parts were close to the training data of *sitting*, resulting in the correct recognition with a majority vote.

However, accuracies of gestures while performing a behavior action still remain lower than those while *standing* since the waveform of gestures changed due to behaviors such as *running*. Filtering out the background activity from the sensor stream is one of the solutions to this problem. The detailed algorithm is our future work.

**Table 6** Processing time for comparison methods and the proposed method (millisecond).

| | SVM | DTW | Proposed Method | | |
|---|---|---|---|---|---|
| | | | Posture | Behavior | Gesture |
| Movement detection | - | - | 0.00141 | | |
| Constancy decision | - | - | - | 0.0452 | |
| Recognition with SVM | 0.0531 | - | 0.00203 | 0.0514 | - |
| Recognition with DTW | - | 137 | - | - | 34.9 |
| Total | 0.0531 | 137 | 0.00344 | 0.0980 | 34.9 |

The drawback of our proposed method can be seen from the results of *hold mobile* and *raise hand* while *running*. These low recall and precision were caused by the fact that the vibration of running is stronger than we assume. Therefore the hand is not classified into postures. The same tendency appeared as results of hand postures while *walking*. Though our method set the threshold according to the intensity of the leg as stated in Section 3.2.1 and it was set based on walking from our pilot study, the vibration was strong even while *walking* in some cases. Employing a flexible threshold is our future work. From our additional experiments, however, *holding mobile phone* and *raising hand* are correctly recognized if the hand is classified as a posture.

Moreover, if the gestures to be recognized includes a non-intensive gesture, $\alpha$ in Eq. (11) should be set to a large value. However, there is a trade-off between gesture intensity and accuracy of constancy decision. To handle small gestures, filtering out global activity from the sensor stream is needed as stated above. Installing a new sensor or a new architecture that detects human intention is another solution. If human intention is detected, small gestures can correctly be recognized with the current proposed system.

### 4.4 Processing Time

This section discusses the processing time of the proposed method. **Table 6** shows the processing time for the movement detection, constancy decision, and recognition with SVM or DTW. The computer used for the evaluation is a SONY VAIO VGN-US90PS (Inter CoreSolo Processor 1.2 GHz). The simulation program is implemented with Visual C++. The evaluation result is the time for each processing calculated based on the processing time for 100,000 trials. The reason the processing time for SVM and DTW differs in the proposed method and compared methods is that the number of activities learned is different. Since most of the recognition with SVM is occupied by feature extraction, the effect from the number of activities to be recognized is small. The recognition with SVM for the proposed method is smaller than that for the comparison method because the proposed method extracts only the mean as a feature value, while the comparison method extracts mean and variance. The number of templates for DTW is one for each activity, and the recognition with DTW takes processing time in response to the number of templates. For the results, the processing time for the movement detection and constancy decision is shorter than that for recognition and processing interval, therefore the proposed method can be applied to real-time applications.

## 5. Consideration of Complex Activities

The number of local activities performed at a time is limited to one and the local activities are confined to right-handed ones in this paper. This section shows problems and answers for extending the proposed system to handle more complex activities.

### 5.1 Local Activities on Every Body Part

To handle local activities of body parts other than the right hand, such as the left hand or leg, recognition models of each body part need to learn local activities. In addition, the activity classification phase has to classify each body part as a gesture. Each body part, however, is apt to be misclassified as a gesture, resulting in misrecognition. This problem can be solved by discarding recognition results whose confidence is lower than the threshold set to the DTW calculation.

### 5.2 Simultaneous Local Activities

For simultaneous different local activities at a time such as *standing while holding a mobile phone while the right hand and with holding on to a strap with the left hand*, recognition results of each body part can simply be integrated. Global activity, however, cannot be obtained when all body parts are recognized as local activities. To contend with the problem, the latest global activity can be used as the current global activity, or the current global activity can be output as “null” or “unknown”. Moreover, the activity integration phase does not simply combine activities of each body part, but considers the relationship of activities and deselects impossible combinations of activities. Since simultaneously performing different local activities is not easy and barely occurs, confining the number of simultaneous local activities to the most confident one would also be a realistic solution.

### 5.3 Local Activities Consisting of Multiple Body Parts

Local activities consisting of multiple body parts such as *folding arms* and *pitching a ball like a baseball pitcher* are also complex activities. Such activities are out of our scope since these are not “local” activity. To extend our system to handle these activities, an intelligent integration algorithm is required. For example, for the *folding arms* posture, the system can simply integrate the recognition results of both arms to the *folding arms* posture when both arms are recognized as *folding arms*. However, rules have to be made for the case that either arm is recognized as *folding arms* and the other arm is misrecognized as another activity, which would depend on applications. One rule would be that the system outputs *folding arms* posture only when both arms are correctly recognized as *folding arms*.

### 5.4 Activities Consisting of a Sequence of Local Activities

Local activities consisting of a sequence of local activities for each body part such as *dancing* also exist. Such activities need not be handled in the activity recognition layer, and the application layer should handle and interpret the sequence of activities.

### 5.5 Summary of Consideration

Summarizing this section, our system can be applied to local activities with body parts other than the right hand. To handle simultaneous different local activities or activities consisting of multiple body parts, our system need not be modified but additional integration rules are required.

## 6. Conclusion

We constructed an activity recognition mechanism for combined activities that classifies each part of the body as posture, behavior, and gesture, then recognizes individual activities and integrates them. Evaluation results confirmed that our proposed method achieved 0.84 recall and 0.86 precision, which is comparable to the method for learning all the combined activities: 0.91 recall and 0.91 precision. As future work, we plan to separate and integrate activities as a more primitive level. That is to say to delete a gesture component from the raw data of a combined activity and to extract an entirely global activity.

### References

[1] Bao, L. and Intille, S.S.: Activity recognition from user-annotated acceleration data, *International Conference on Pervasive Computing* (*Pervasive 2004*), pp.1–17 (2004).

[2] Pärkkäa, J., Ermes, M., Korpipää, P., Mäntyjärvi, J., Peltola, J. and Korhonen, I.: Activity classification using realistic data from wearable sensors, *Trans. Information Technology in Biomedicine*, Vol.10, pp.119–128 (2006).

[3] Lester, J., Choudhury, T. and Borriello, G.: A practical approach to recognize physical activities, *International Conference on Pervasive Computing* (*Pervasive 2006*), pp.1–16 (2006).

[4] Chambers, G.S., Venkatesh, S., West, G.A.W. and Bui, H.H.: Hierarchical recognition of intentional human gestures for sports video annotation, *International Conference on Pattern Recognition* (*ICPR ’02*), pp.1082–1085 (2002).

[5] Naya, F., Ohmura, R., Takayanagi, F., Noma, H. and Kogure, K.: Workers’ Routine Activity Recognition using Body Movement and Location Information, *International Symposium on Wearable Computers* (*ISWC 2006*), pp.105–108 (2006).

[6] Murao, K. and Terada, T.: A Motion Recognition Method by Constancy-Decision, *International Symposium on Wearable Computers* (*ISWC 2010*), pp.69–72 (2010).

[7] Ravi, N., Dandekar, N., Mysore, P. and Littman, M.L.: Activity Recognition from Accelerometer Data, *National Conference on Artificial Intelligence* (*AAAI 2005*), pp.1541–1546 (2005).

[8] Westeyn, T., Brashear, H., Atrash, A. and Starner, T.: Georgia tech gesture toolkit: Supporting experiments in gesture recognition, *International Conference on Multimodal Interfaces* (*ICMI 2003*), pp.85–92 (2003).

[9] HTK Hidden Markov Model Toolkit home page, available from ⟨http://htk.eng.cam.ac.uk/⟩.

[10] Junker, H., Amft, O., Lukowicz, P. and Tröster, G.: Gesture spotting with body-worn inertial sensors to detect user activities, *Journal of Pattern Recognition Society*, pp.2010–2024 (2008).

[11] Keogh, E., Chu, S., Hart, D. and Pazzani, M.: An online algorithm for segmenting time series, *International Conference on Data Mining* (*ICDM 2001*), pp.289–296 (2001).

[12] Liu, J., Wang, Z., Zhong, L., Wiekramasuriya, J. and Vasudevan, V.: Uwave: Accelerometer-based personalized gesture recognition and its applications, *International Conference on Pervasive Computing and Communications* (*PerCom 2009*), pp.1–9 (2009).

[13] Myers, C.S. and Rabiner, L.R.: A comparative study of several dynamic time-warping algorithms for connected word recognition, *The Bell System Technical Journal*, Vol.60, No.7, pp.1389–1409 (1981).

[14] Rabiner, L.R. and Juang, B.H.: A tutorial on hidden markov models and selected applications in speech recognition, *IEEE*, pp.257–286 (1989).

[15] Yoshida, G., Murao, K., Terada, T. and Tsukamoto, M.: Method of Determining Training Data for Gesture Recognition considering Decay of Gesture Movements, *Workshop on Context Modeling and Reasoning 2013* (*CoMoRea 2013*), pp.13–18 (2013).

[16] Toda, M., Akita, J., Sakurazawa, S., Yanagihara, K., Kunita, M. and Iwata, K.: Wearable Biomedical Monitoring System Using TextileNet, *International Symposium on Wearable Computers* (*ISWC 2006*), pp.119–120 (2006).

[17] Shen, C.L., Kao, T., Huang, C.T. and Lee, J.H.: Wearable Band Using a Fabric-Based Sensor for Exercise ECG Monitoring, *International Symposium on Wearable Computers* (*ISWC 2006*), pp.143–144 (2006).

[18] Ouchi, K., Suzuki, T. and Doi, M.: LifeMinder: A wearable Health-

care Support System Using User's Context, *International Workshop on Smart Appliances and Wearable Computing* (*IWSAWC 2002*), pp.791–792 (2002).

[19] Laerhoven, K.V. and Gellersen, H.W.: Spine versus Porcupine: A Study in Distributed Wearable Activity Recognition, *International Symposium on Wearable Computers* (*ISWC 2004*), pp.142–149 (2004).

[20] Stiefmeier, T., Ogris, G., Junker, H., Lukowics, P. and Tröster, G.: Combining Motion Sensors and Ultrasonic Hands Tracking for Continuous Activity Recognition in a Maintenance Scenario, *International Symposium on Wearable Computers* (*ISWC 2006*), pp.97–104 (2006).

[21] Fujinami, K., Jin, C. and Kouchi, S.: Tracking On-body Location of a Mobile Phone, *International Symposium on Wearable Computers* (*ISWC 2010*), *Late Breaking Results - Cutting Edge Technologies on Wearable Computing*, pp.190–197 (2010).

[22] Junker, H., Lukowicz, P. and Tröster, G.: Sampling Frequency, Signal Resolution and the Accuracy of Wearable Context Recognition Systems, *International Symposium on Wearable Computers* (*ISWC 2004*), pp.176–177 (2004).

[23] Wireless Technologies Inc., available from ⟨http://www.wireless-t.jp/⟩.

[24] Intille, S.S., Bao, L., Tapia, E.M. and Rondoni, J.: Acquiring in Situ Training Data for Context-aware Ubiquitous Computing Application, *Conference on Human Factors in Computing Systems* (*CHI 2004*), pp.1–9 (2004).

[25] Vapnik, V.: The Nature of Statistical Learning Theory, Springer (1995).

[26] Junker, H., Lukowicz, P. and Tröster, G.: Sampling Frequency, Signal Resolution and the Accuracy of Wearable Context Recognition Systems, *International Symposium on Wearable Computers* (*ISWC2004*), pp.176–177 (2004).

[27] Park, T., Lee, J., Hwang, I., Yoo, C., Nachman, L. and Song, J.: E-gesture: A collaborative architecture for energy efficient gesture recognition with hand-worn sensor and mobile devices, *Conference on Embedded Networked Sensor Systems* (*SenSys 2011*), pp.260–273 (2011).

[28] Korpela, J., Takase, K., Hirashima, T., Maekawa, T., Eberle, J., Chakraborty, D. and Aberer, K.: An Energy-Aware Method for the Joint Recognition of Activities and Gestures Using Wearable Sensors, *International Symposium on Wearable Computers* (*ISWC 2015*), pp.101–108 (2015).

**Kazuya Murao** is an Assistant Professor at College of Information Science and Engineering, Ritsumeikan University, Japan. He received his B.Eng, M.Info.Sci., and Ph.D. degrees from Osaka University in 2006, 2008, and 2010, respectively. Prof. Murao is working on wearable computing, ubiquitous computing, and context aware systems. He is a member of IEEE, ACM and IPSJ.

**Tsutomu Terada** is an Associate Professor at the Graduate School of Engineering, Kobe University, Japan. He is also a PRESTO researcher at the Japan Science and Technology Agency, Japan. He received his B.Eng., M.Eng., and Ph.D. degrees from Osaka University in 1997, 1999, and 2003, respectively. Prof. Terada is working on wearable computing, ubiquitous computing, and entertainment computing. He is a member of IEEE, ACM, IPSJ, and IEICE.