Design of a Low-false-positive Gesture for a Wearable Device

Ryo Kawahata[1], Atsushi Shimada[2], Takayoshi Yamashita[3], Hideaki Uchiyama[4] and Rin-ichiro Taniguchi[4]

[1]*Graduate School of Information Science and Electrical Engineering, Kyushu University, 744, Motooka, Nishi-ku, 819-0395, Fukuoka, Japan*

[2]*Faculty of Arts and Science, Kyushu University, 744 Motooka, Nishi-ku, 819-0395, Fukuoka, Japan*

[3]*Department of Computer Science, College of Engineering Chubu University, 1200, Matsumoto-cho, 487-8501, Kasugai, Aichi, Japan*

[4]*Faculty of Information Science and Electrical Engineering, Kyushu University, 744, Motooka, Nishi-ku, 819-0395, Fukuoka, Japan*

Keywords: Gesture Recognition, Wearable Device.

Abstract: As smartwatches are becoming more widely used in society, gesture recognition, as an important aspect of interaction with smartwatches, is attracting attention. An accelerometer that is incorporated in a device is often used to recognize gestures. However, a gesture is often detected falsely when a similar pattern of action occurs in daily life. In this paper, we present a novel method of designing a new gesture that reduces false detection. We refer to such a gesture as a low-false-positive (LFP) gesture. The proposed method enables a gesture design system to suggest LFP motion gestures automatically. The user of the system can design LFP gestures more easily and quickly than what has been possible in previous work. Our method combines primitive gestures to create an LFP gesture. The combination of primitive gestures is recognized quickly and accurately by a random forest algorithm using our method. We experimentally demonstrate the good recognition performance of our method for a designed gesture with a high recognition rate and without false detection.

1 INTRODUCTION

Wearable devices have become widespread in society. Various devices include eyeglass devices (e.g., Google Glass) and wristband devices (e.g., Nike+ FuelBand), and in particular, wrist-watch-type devices, called smartwatches, have become increasingly familiar in daily life.

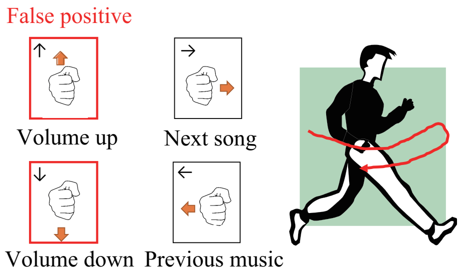
People can use many applications (e.g., email, map navigation and music player applications) on a smartwatch. Surface gestures (e.g., tapping, swiping, and flicking) are often used when manipulating the applications on a smartphone. However, in the case of the smartwatch, people are forced to manipulate the applications on a small touch screen. It has therefore become important to develop a new interaction method such as interaction by motion gesture for ease of use (Park et al., 2011).

Motion gesture enables more intuitive interaction than interaction with a keyboard or touch screen because people only need to perform a simple action like flicking a wrist. However, an interaction system that is based on motion gestures needs to recognize the gestures with a high recognition rate and low false positive (LFP) rate for users. To recognize gestures, sensors such as an accelerometer contained in a smartwatch are often used. An interaction system that is based on motion gestures and used in daily life faces the problem that the gesture recognizer will find it difficult to distinguish between gestures for operation of an application and everyday motions.

Figure 1 shows an example of the problem. There are four designed gestures for the operation of a music player on a smartwatch. The two gestures of "Volume up" and "Volume down" are detected falsely when the user is walking because the two gestures are almost the same as the everyday motion of walking.

There are two main solutions to the problem. One solution is for the user to press or touch a button before making gestures so as to segment gestures from everyday motions. This is an obstacle to intuitive interaction with a smartwatch because the solution requires the user to use both hands to push a button whenever the user operates applications by gestures.

Kawahata, R., Shimada, A., Yamashita, T., Uchiyama, H. and Taniguchi, R-i.
Design of a Low-false-positive Gesture for a Wearable Device.
DOI: 10.5220/0005701905810588
In *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2016)*, pages 581-588
ISBN: 978-989-758-173-1



(a) (b)

Figure 1: (a) Gestures for operation of an application, (b) everyday motion (walking).

The other approach is to use uncommon gestures; i.e., gestures with sensor patterns that do not appear frequently in daily motions. These gestures are referred to as LFP gestures. Specifically, a certain gesture is used to indicate the beginning and end of gestural input such as in the case of a delimiter (Ruiz and Li, 2011) or used as a gesture for operation of an application directly (Ashbrook and Starner, 2010). This approach does not require a user to press a button, but LFP gestures tend to be complex because simple actions are often part of daily motions. Conventionally, interaction designers carefully design LFP gestures by analyzing daily motions and by considering the situations in which motion gestures are used. In addition, gestural input depends on the applications. The design of LFP gestures thus remains difficult.

In this paper, we propose a method of suggesting LFP gestures automatically. Our method searches LFP patterns of simple gestures in daily motions and suggests LFP gestures to the system user. A simple action is referred to as a primitive gesture in this paper. The combination of simple actions reduces the LFP rate. Additionally, the LFP gesture suggested by our system does not restrict intuitive gesture interaction because of its use of simple actions. In fact, in Section 4.3, we experimentally demonstrate that one simple action happens more frequently than two successive primitive gestures in daily motions. The details of our method are given in Section 3.

There are two kinds of users of our system: an interaction designer and a gesture user. Interaction designers design the application interface and use gestures for the interface. They consider the situation of using an application and apply gestures to application commands on the basis that there are no false detections in long motions of the situation (lasting more than 1 week). Meanwhile, gesture users operate the application in practice using gestures. They apply gestures to application commands manually for ease of use on the basis that there are no false detections in daily motions (lasting about 1 day). We present experiments assuming a gesture user as our system user in Section 4.

2 RELATED WORK

Gesture recognition is an active area of research on human–computer interaction (Mitra and Acharya, 2007). In particular, the recognition of hand gestures has become more pervasive and has a wide range of applications such as the recognition of sign language (Zafrulla et al., 2011) and an interaction system for surgery (Ruppert et al., 2012).

There are two approaches for recognizing hand gestures: the use of vision-based methods and the use of sensor-based methods. A vision-based method recognizes hand gestures to detect hand motions or hand shapes using an RGB camera (Chen et al., 2007). This method is based on image processing that segments the hand area in the image. Segmentation of a hand gesture is easily affected by illumination variations and the positional relation between the camera and hand, which is a large limitation in the case of a mobile environment.

In contrast, a sensor-based method often uses an accelerometer to recognize gestures (Schlömer et al., 2008). Such methods have received much attention with the widespread use of smartphones and wearable devices that incorporate accelerometers and gyroscopes. In practice, a sensor-based method is applied to operate a smartphone (Ruiz et al., 2011) and smartwatch (Park et al., 2011). Conventional recognition methods using acceleration often focus on manually segmented gestures to avoid false gesture detection (Akl et al., 2011). However, considering the continuous gesture is important for real-time application. In handling this false-detection problem, previous research has required the user to press a button to notify the system of gesture input (Liu et al., 2009). In a wearable environment, pressing a button bothers the user because it requires the user to use both hands. Another method of solving the false-detection problem is improving the detector performance using a threshold. This method assumes that there is a difference between the gesture and daily movement, such as a difference in movement speed (Park et al., 2011). The start point of a gesture is the time at which the processed sensor value first exceeds the threshold. The method of using a threshold cannot deal with the problem that motion patterns that are similar to the gesture happen by chance during daily motion.

Another interesting method is to use an LFP gesture. An LFP gesture is designed on the basis that the gesture rarely appears in daily motions. This method allows gesture interaction without pressing a button or the false detection of gestures. Ruiz et al. designed LFP gestures for mobile interaction using a motion gesture delimiter called Doubleflip (Ruiz and Li, 2011). Doubleflip is user-friendly because the gesture consists of a combination of simple actions. Ruiz et al. evaluated the true positive rate and false positive rate of a gesture for 2100 hours of motion data. Considering the manipulation of an application, types of gesture depend on the application and situation. Therefore, designing an LFP gesture is a difficult task for the gesture designer, who frequently needs to create gestures for new applications and to determine the LFP rates of the gestures.

Ashbrook et al. proposed a design tool for the creation of LFP gestures (Ashbrook and Starner, 2010). The tool calculates the false positive rate of an input gesture from daily motion. The user of the tool can easily discriminate whether the input gesture will be detected falsely or not in daily motion. However, the user is required to repeat the design and input of gestures many times to find LFP gestures. Designing an LFP gesture thus remains difficult.

Kohlsdorf et al. proposed a new gesture design tool that facilitates the design of an LFP gesture. Their system suggests an LFP gesture automatically from input daily motion. Employing their method, daily motion is replaced by symbol sequences and a low-false-rate gesture is created by finding a symbol sequence that does not appear frequently in the input daily motion. Their system is limited to surface gestures, which are two-dimensional gestures on a touch pad, because of the restoration from the symbol sequence to gesture.

We propose a primitive-based gesture creation method for a gesture suggestion system. Our proposed method can suggest motion gestures for the system user using information of primitive gestures. A primitive-based method is used in the recognition of sign language (Bauer and Kraiss, 2002) and activity recognition (Zhang and Sawchuk, 2012).

3 PROPOSED METHOD BASED ON PRIMITIVE GESTURES

3.1 System Overview

We propose a method of searching and suggesting LFP motion patterns for a system that creates LFP gestures automatically. Figure 2 presents the system scenario. The system scenario of gesture creation is inspired by a system made by (Kohlsdorf and Starner, 2013) but differs in the way that LFP motion patterns

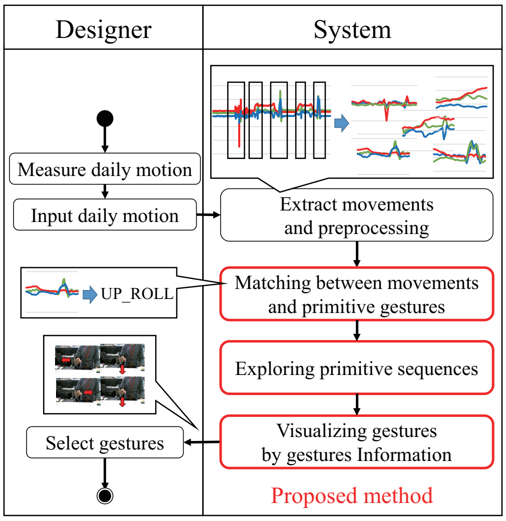


Figure 2: System scenario.

are searched for and suggested. While they uses symbol sequence for searching LFP motion patterns, our method searches for and suggests LFP motion patterns by considering the combination of primitive gestures.

There are a huge number of hand motion patterns in daily motion when we take into account all hand positions, directions, and movements. It is therefore difficult to find LFP patterns concretely because of the computational cost. To find LFP patterns, we make one assumption about the LFP gesture. The assumption is that the LFP gesture is a combination of primitive gestures that are rarely detected in input daily motion. Hand motion is represented by a limited number of motions and the system can explore LFP patterns according to the assumption.

We here introduce the flow of LFP gesture creation. First, a system user measures daily motions using sensors in a smartwatch and inputs the daily motions to our system. Our system runs a low-pass filter over input daily motions and extracts periods of high accelerometer values from the daily motions to eliminate periods in which there is no hand motion. Next, extracted data are matched with primitive gestures and a sequence of primitive gestures (i.e., the primitive sequence) is expressed. The proposed system counts the number of primitive sequences in the daily motions and finds primitive sequences that have low occurrence in the daily motions. Finally, the system gives primitive sequences and the user selects those that the user wants to use for application.

3.2 Design of a Primitive Gesture

Suggesting gestures to the system user requires the reconstruction of hand motions from sensor values,

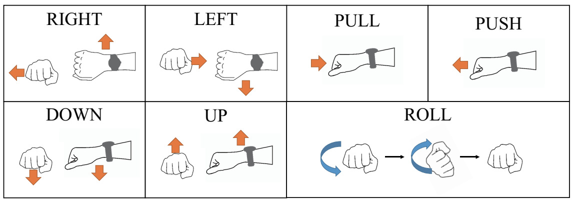


Figure 3: Primitive gesture.

which is difficult because sensor data such as accelerometer data lose motion information of the hand position and direction. Generally, multiple sensors such as those of a motion capture system are used in reconstruction and a complicated and sophisticated hand tracking method is required.

The proposed method uses information of primitive gestures for the reconstruction. Primitive gestures are components of motion gestures. In previous research, primitive gestures have been constructed employing an unsupervised clustering algorithm (Zhang and Sawchuk, 2012) (Bauer and Kraiss, 2002). First, sensor data are divided into a sequence of fixed-length-window cells (i.e., segments) and the feature vector for each segment of the sequence is calculated. Segments are then clustered according to their feature vectors and the center of a cluster is taken as a primitive gesture. As a result, vocabulary size of a primitive gesture depends on the cluster obtained from sensor data. It is inconvenient to suggest a certain LFP gesture because it cannot be expected to emerge from primitive gestures. Therefore, in our method, the primitive gesture is defined in advance by ourself. The use of predefined primitive gestures allows us to find certain motions from sensor data and we can thus represent sensor data with the predefined motions. As a result, the proposed method can reconstruct a sequence of predefined motions from sensor data. Furthermore, it can reconstruct hand motions more easily with only one accelerometer sensor than a motion capture system.

Figure 3 shows seven primitive gestures for our proposed method. These primitive gestures consist of simple and short movements so as to avoid motion complexity when primitives are combined. The sensors are oriented upwards because of visual feedback.

3.3 Preprocessing

Sensor data include much noise around high-frequency components, which is an obstacle to achieving high recognition performance. We adopt the weighted moving average to smooth the sensor data.

In our case, it is desirable only to handle data of hand movement (what we call the movement area) in daily motions. The recording of daily motion involves the collection of much data but no predefined movement area, and treating all data is thus a waste of computational time. We extract the movement area using threshold-based method. Let $A = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$ denote the time series of acceleration and $\mathbf{a} = (a_x, a_y, a_z)$ denote acceleration. We evaluate the amplitude of movement $\mathbf{G} = (G_x, G_y, G_z)$ by comparison between two observations; $\mathbf{a}_i$ and $\mathbf{a}_{i-N}$.

$$\mathbf{G} = |\mathbf{a}_i - \mathbf{a}_{i-N}| \tag{1}$$

The extraction of the movement starts when G_x, G_y or G_z is higher than the threshold at the start point, Th_s. The end point of the extraction is decided by two threshold; one is about the G and the other is about the time domain. In our method, we handle continuous gestures like the primitive sequences. Therefore, we set a temporal threshold T_e to ending point of the extraction not to split the continuous gestures. The extraction ends when G_x, G_y and G_z are smaller than Th_e for a period of T_e. This extracted area by the thresholds is called extracted period in this paper.

It is desirable to normalize the sensor data in handling the variability of gestures. Measured acceleration consists of two components: a dynamic component and gravitational component. The dynamic component relates to movement while the gravitational component relates to the change in tilt of the device. The variability of the sensor tilt affects recognition performance. The mean of the measured acceleration on each axis is the best estimate gravitational component value. We normalize the measurement data extracted via the threshold method by subtracting the mean from the data.

3.4 Feature Representation

The proposed method is similar to a bag-of-features method (Zhang and Sawchuk, 2012) when extracting features from the data of an accelerometer. The proposed method calculates the gradient of acceleration as a feature. The calculation flow is shown in Figure 4(a). First, as shown in Figure 4(a-1), the proposed method separates sensor data, extracted by a time window, into subsequences. The length of a subsequence is l_s and subsequences are extracted with shifting size l_t. Next, the gradient of accelerometer data is calculated for each subsequence and quantized into 5 levels as shown Figure 4(a-1). Then, a gradient histogram is made as shown in Figure 4(a-2). The proposed method divides a set of subsequences into n_h sub-windows and produces a histogram for each sub-window. Generally, a bag-of-features method ignores the order of observation, it causes confusion of

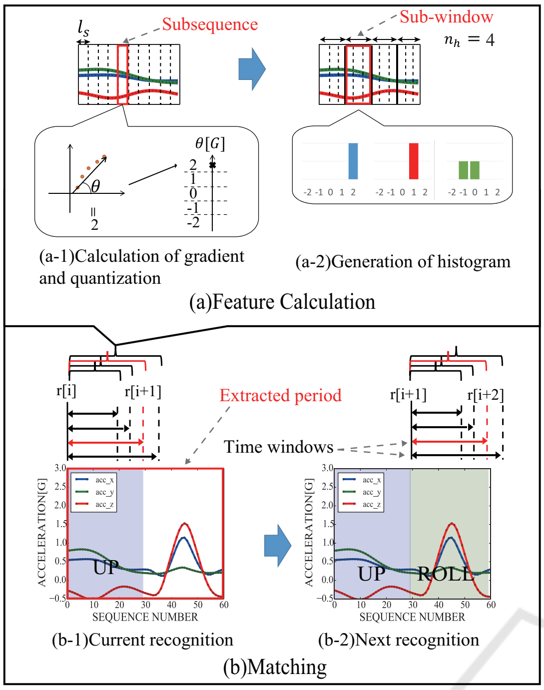


Figure 4: Feature calculation and matching between daily motion and primitive gestures.

movements such as LEFT and RIGHT. To solve this problem, the proposed method create a histogram in each sub-window. Finally, the histograms are concatenated to represent a feature vector.

3.5 Matching between Daily Motion and Primitive Gestures

The proposed method employs a time-series matching method for mapping between daily motion and primitive gestures. Dynamic time warping (DTW) is a general approach for time-series matching (Liu et al., 2009) (Akl et al., 2011) and allows us to calculate the distance between two temporal sequences, which may differ in length. DTW attempts to match all training samples one by one and has a high computational cost. It thus takes a long time to match daily motion measured over a long time and primitive gestures.

The proposed method uses the random forest algorithm (Liaw and Wiener, 2002) to reduce computational cost. The random forest is a method of ensemble learning for multiple classification. Multiple decision trees constitute a random forest and they are trained to control variance. In the testing phase, the random forest algorithm uses a discriminant function obtained in the training phase to map between daily motion and primitive gestures at high speed.

There are often variations between training and testing samples in the direction of the time axis. To handle these variations, we generate new training samples to expand, shrink and shift the original training samples along the time axis. Original training samples are expanded by linear interpolation and shrunk by decimating samples at regular intervals.

A matching between daily motion and primitive gestures is sequentially performed. The matching algorithm is shown in Figure 4(b). To handle the variation of gesture length, we set up several sizes of time windows for matching. A time window consists of subsequences defined in Section 3.4, so that a feature vector of each time window is represented by a concatenated histogram given in Figure 4(a). To simplify the explanation, we denote w_j as a time window, and its length as $|w_j|$. For each time window w_j, we firstly acquire a candidate of primitive gesture by the highest matching probability of class c. Then, we select a window $\hat{w}_j$ which has the highest matching probability in the all windows, and regards the class label of $\hat{w}_j$ as the recognition result. To achieve sequential recognition, we have to define the start point of recognition according to the previous recognition processing. Let $r[i]$ be the start point of current recognition, shown in Figure 4(b-1), and the issue is to set the start point of next recognition $r[i+1]$, given in Figure 4(b-2). As explained above, we acquire the recognition result for $r[i]$ as c recognized in $\hat{w}_j$, therefore, the time length of $\hat{w}_j$ is simply added to $r[i]$ to start the next recognition.

$$r[i+1] = r[i] + |\hat{w}_j| \quad (2)$$

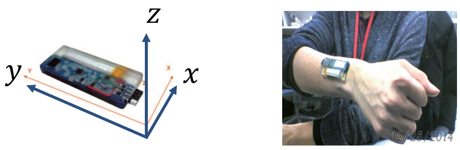
We repeat this sequential recognition processing l_m times by updating $r[i]$. For instance, if we would like to recognize two successive primitive gestures, we have to set the l_m to be 2.

4 EXPERIMENT

In this section, we report two experiments for evaluation of recognition performance and true positive and false positive rates of gestures created by our system. First, we investigate primitive patterns searched for by our system from daily motions in our laboratory and discuss characteristics of the gestures. Next we compare the proposed method with the DTW method in terms of their performance in recognizing gestures obtained in the first experiment.

4.1 Dataset and Parameters

In this experiments, we measured daily motions in our laboratory. These daily motions included hand mo-



(a) (b)

Figure 5: (a) Accelerometer and sensor axis, (b) sensor position.

tions made while, for example, using a computer, eating a meal, reading and writing, and walking. The major activity of the daily motion was the use of the computer.

We used the accelerometer shown in Figure 5, which made measurements at 50 Hz. This wireless sensor can record sensor data in internal memory and work continuously for 4 hours. As shown in Figure 5, we attached this sensor to the forearm, as if we were using a smartwatch.

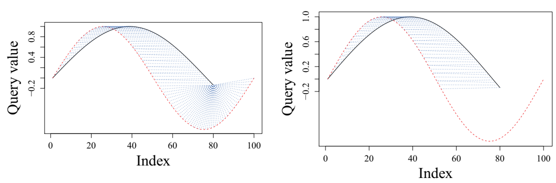
The daily motion in the laboratory was measured for one subject on separate days. The subject was instructed not to use primitive gestures deliberately. The total measurement time was 24 hours. In terms of the primitive gestures, we collected 20 samples per gesture for training the random forest algorithm.

The parameters for preprocessing was set $Th_s = 0.1G$, $Th_e = 0.05G$, $T_e = 0.7s$, $N = 5$ in our experiment. In terms of the matching parameters, we empirically set $l_s = 6$, $l_t = 1$, $n_h = 4$, $l_m = 2$, $\{|w_1|, |w_2|, \ldots\} = \{10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50\}$ in this paper.

4.2 Comparative Approach

The proposed method replaces measurement data with primitive sequences to search for LFP patterns. Additionally, gestures created by our system are recognized by matching between measurement data and primitive gestures. From the above, the searched patterns and performance of recognition of the created gestures depend on the matching method.

We used two kinds of DTW method for matching. The first method is conventional DTW.



(a) (b)

Figure 6: Data mapping between two time series of data: (a) conventional DTW, (b) open-end DTW.

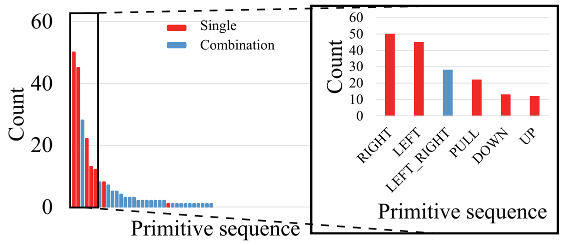


Figure 7: Top six primitive sequences.

This method fixes the start and end points of the calculated distance. We used the matching path and local distance from previous research (Liu et al., 2009) for a comparative approach. When matching between extracted period and primitive sequences, a sliding window is used. This window size is estimated by the mean of the length of primitive gestures as training samples and strides at intervals whose size is half the window size.

The second method is open-end DTW. This method can perform partial matching because the end point is flexible, and it is thus often used for continuous word recognition. See (Mori et al., 2006) (Oka, 1998) for more information. We used the matching path from previous research (Mori et al., 2006) and the same local distance as previously used.

An example of the difference in matching between the two methods described above is shown in Figure 6. The conventional DTW method maps one time series of data to the other overall under a constraint. In contrast, open-end DTW can find one in the other more suitable.

4.3 Characteristics of Created Gestures

We investigate the primitive sequences searched for by our system and count the primitive gestures in daily motions in the laboratory. The maximum length of a primitive sequence l_m is set to two primitive gestures because the duration of one primitive gesture is about 0.8 seconds and a duration longer than three lengths of a primitive gesture (longer than 2.4 seconds) is a burden on the user.

The top six gestures in terms of the count are shown in Figure 7. We rejected a pattern if the pattern was dissimilar to all primitive gestures by a thresh-

Table 1: Primitive sequences not appearing in daily motion.

DOWN_DOWN,	RIGHT_**ROLL**,	**ROLL**_RIGHT
LEFT_**ROLL**,	**ROLL**_DOWN,	**ROLL**_**ROLL**
PULL_**ROLL**,	**ROLL**_LEFT,	**ROLL**_UP
PULL_UP,	**ROLL**_PULL,	UP_PUSH
PUSH_**ROLL**,	**ROLL**_PUSH,	UP_**ROLL**

old amount. Therefore, the absolute number of occurrences of primitive gestures depended on the threshold. In this case, there was a large number of single primitive patterns. As a result, a simple primitive pattern was detected falsely more often than a combination of primitive patterns if we use such a primitives as an input gesture for a system.

Meanwhile, some primitive sequences did not appear in the daily motions. Table 1 gives primitive sequences that none of all of the three recognition methods observed. These primitive sequences often included the ROLL gesture. The ROLL gesture is thus resistant to false detection.

Table 2: Time required to search for primitive sequences and the LFP rate for daily motions over a period of 24 hours.

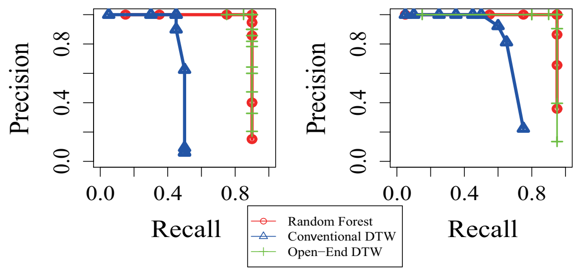
Conventional DTW	Open-End DTW	Random Forest
119[s]	809[s]	67[s]

The time required to search for primitive sequences on a laptop computer having an Intel Core i7 2.8-Hz CPU with 8 GB RAM is given in Table 2. The random forest algorithm is fastest and can search for LFP patterns 10 times as fast as the open-end method.

4.4 Accuracy

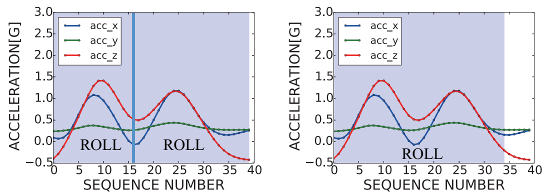
We evaluated the performance of recognition of primitive sequences for each recognition method. The recognition of primitive sequences by our system should be at a high recognition rate and LFP rate for users. We selected ROLL_ROLL gestures and UP_ROLL gestures for the evaluations on the basis of the results presented in Section 4.3. A precision–recall curve was used for this evaluation. We prepared other daily motions in our laboratory for a period of 4 hours for evaluation.

When recognizing a specified primitive sequence, the proposed method calculates the evaluation value (distance or similarity) for the extracted period and primitive gesture only correspond to the one. For example, when we specified the UP_ROLL gesture, the evaluation value is only calculated for the extracted period and UP gesture as a first primitive. The ROLL gesture is then used to calculate the evaluation value for a second primitive. The evaluation values are then summed and the mean of the evaluation value is estimated. Finally, the primitive sequence is recognized using a threshold of the mean value. In our case, the estimated primitive sequence does not always correspond to a specified sequence when the threshold is adjusted correctly because the lengths of primitive sequences are different when the recognizer erroneously finds primitive gestures in the primitive sequences. Therefore, recall is not always 1 when using the threshold.



(a)UP_ROLL (b)ROLL_ROLL

Figure 8: Precision–recall curve of UP_ROLL and ROLL_ROLL gesture.



(a) (b)

Figure 9: Example of false recognition: (a) ground truth, (b) estimated result.

The results of the recognition performance of ROLL_ROLL gestures and UP_ROLL gestures are shown in Figures 8(a) and 8(b). The random forest algorithm and open-end DTW had the best thresholds for the recognition of specified gestures at a high recognition rate with no false detection. Conventional DTW performed worse.

4.5 Discussion

Although our system often searched for LEFT and RIGHT gestures, which are simple actions, from daily motions in our laboratory, the ROLL gesture was not detected frequently. The daily motions include many actions relating to using a computer mouse. When using a mouse, a hand moves horizontally on a desk. As a result, primitive gestures such as LEFT and RIGHT, which include hand movement parallel to the ground, were often detected.

There were some cases that two successive gestures were mistakenly recognized as single gesture. We show an example of false recognition in Figure 9. In this example, the ROLL_ROLL gesture is recognized as a ROLL gesture falsely by the recognizer. Employing our method, the recognizer uses the gradient of acceleration. The outline of this ROLL_ROLL gesture is similar to that of the ROLL gesture in terms of this feature. This problem is solved if we adjust the size of window used to extract sensor data when matching.

The presented experiments demonstrate that the random forest algorithm has recognition performance similar to that of open-end DTW at high computational speed. The matching speed is important not only for intuitive interaction but also for the usability of our system. Practically, the dataset of daily motion will be longer than 24 hours in some cases. The random forest algorithm is suitable for our system designed to find optimal gestures for certain applications and situations quickly.

5 CONCLUSION AND FUTURE WORK

For intuitive interaction with wearable devices, gesture recognition has advantages over traditional methods such as gestures on a touch pad. In terms of recognizing gestures correctly for a smartwatch, the false positiveness of gestures is a big problem.

We proposed a primitive-based gesture recognition approach to solve the problem. This approach creates new gestures that are resistant against false detection in daily motions. We assume one system for LFP gesture creation. This system records daily motion data from users and searches for LFP patterns in the daily motions employing our proposed method. The system searches for and visualizes LFP motion gestures by focusing on primitive gestures.

In future work, we will continue to evaluate our proposed method for multiple people and investigate a way of visualizing primitive sequences though the evaluation. In addition, we will verify the validity of our method for seven primitive gestures.

REFERENCES

Akl, A., Feng, C., and Valaee, S. (2011). A novel accelerometer-based gesture recognition system. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 59(12):6197.

Ashbrook, D. and Starner, T. (2010). Magic: a motion gesture design tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2159–2168. ACM.

Bauer, B. and Kraiss, K.-F. (2002). Video-based sign recognition using self-organizing subunits. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, pages 434–437. IEEE.

Chen, Q., Georganas, N. D., and Petriu, E. M. (2007). Real-time vision-based hand gesture recognition using haar-like features. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1–6. IEEE.

Kohlsdorf, D. K. H. and Starner, T. E. (2013). Magic summoning: towards automatic suggesting and testing of gestures with low probability of false positives during use. *The Journal of Machine Learning Research*, 14(1):209–242.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

Liu, J., Zhong, L., Wickramasuriya, J., and Vasudevan, V. (2009). uwave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675.

Mitra, S. and Acharya, T. (2007). Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3):311–324.

Mori, A., Uchida, S., Kurazume, R., Taniguchi, R.-i., Hasegawa, T., and Sakoe, H. (2006). Early recognition and prediction of gestures. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 560–563. IEEE.

Oka, R. (1998). Spotting method for classification of real world data. *The Computer Journal*, 41(8):559–565.

Park, T., Lee, J., Hwang, I., Yoo, C., Nachman, L., and Song, J. (2011). E-gesture: a collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 260–273. ACM.

Ruiz, J. and Li, Y. (2011). Doubleflip: a motion gesture delimiter for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2717–2720. ACM.

Ruiz, J., Li, Y., and Lank, E. (2011). User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 197–206. ACM.

Ruppert, G. C. S., Reis, L. O., Amorim, P. H. J., de Moraes, T. F., and da Silva, J. V. L. (2012). Touchless gesture user interface for interactive image visualization in urological surgery. *World journal of urology*, 30(5):687–691.

Schlömer, T., Poppinga, B., Henze, N., and Boll, S. (2008). Gesture recognition with a wii controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 11–14. ACM.

Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., and Presti, P. (2011). American sign language recognition with the kinect. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 279–286. ACM.

Zhang, M. and Sawchuk, A. A. (2012). Motion primitive-based human activity recognition using a bag-of-features approach. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 631–640. ACM.