



Determining a Number of Training Data for Gesture Recognition Considering Decay in Gesture Movements

Murao, Kazuya
Yoshida, Gaku
Terada, Tsutomu
Tsukamoto, Masahiko

(Citation)

ヒューマンインタフェース学会論文誌, 17(2):213-222

(Issue Date)

2015-05-25

(Resource Type)

journal article

(Version)

Version of Record

(Rights)

© 2015 ヒューマンインタフェース学会

本著作物の著作権は特定非営利活動法人ヒューマンインタフェース学会に帰属します。本著作物は著作権者であるヒューマンインタフェース学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」に従うことをお願い致します。

(URL)

<https://hdl.handle.net/20.500.14094/0100492902>



Determining a Number of Training Data for Gesture Recognition Considering Decay in Gesture Movements

Kazuya Murao^{*1}, Gaku Yoshida^{*2}, Tsutomu Terada^{*2 *3} and Masahiko Tsukamoto^{*2}

Abstract – Mobile phones and video game controllers using gesture recognition technologies enable easy and intuitive operations, such as those in drawing objects. Gesture recognition systems generally require several samples of training data before recognition takes place. However, recognition accuracy deteriorates as time passes since the trajectory of the gestures changes due to fatigue or forgetfulness. We investigated the change in gestures and found that the first several samples of gestures were not suitable for training data. Therefore, we propose two methods of finding appropriate data for training for long-term use. We confirmed that the proposed methods found better training data than the conventional method from the viewpoints of the number of data collected and recognition accuracy.

Keywords : gesture recognition, training data selection, accelerometer

1. Introduction

Downsizing of computers has led to mobile and wearable computing that has recently attracted a great deal of attention. In particular, accelerometers are installed in most current devices, such as the iPhones and Android-powered devices, and the video game controllers for Wii or PS3, which enable easy and intuitive operations. In addition, life logs can be enriched by recording gesture movements in daily life with wearable sensors.

Gesture recognition systems generally have to be trained with users' gesture data before use, and recognize unknown gestures by comparing them with the training data. The simplest and most common way to train the systems is to use a few samples of gesture data before using the systems for the first time. Recognition accuracy, however, could deteriorate since the gestures to be recognized change due to user conditions, such as forgetfulness of the original gestures and day-to-day fatigue. As far as we know, no effective methods on training considering change in future gesture motions have yet been reported. Someone may think that employing robust gestures for decay with time solves the problem, however it is hard to find such gestures since uncomplicated gestures are not simply robust for decay. Even if robust

gestures were found, application developers cannot chose gestures as they like in many cases, such as actuating characters in video game and detecting actions in daily life, therefore decay in movement has to be considered for any kinds of gestures.

We investigated the change in gesture motions by repeating specific gestures 200 times a day for seven days, and confirmed that the first several samples were not suitable for training data which is commonly used in the conventional methods, since gesture motions changed by repetition. Therefore, we propose two methods of finding training data considering changes in future gesture motions. We confirmed that the proposed methods found training data in an early stage of data collection that were robust to temporal changes.

This paper is organized as follows. Section 2 introduces related work. Section 3 describes a preliminary experiment that we conducted to evaluate what effect would have on gesture motions. Section 4 presents our proposed methods. The experiment we conducted to evaluate the accuracy of our methods are described in Section 5. The summary and future work are mentioned in Section 6.

2. Related work

Many studies on gesture recognition using accelerometers have been reported. Murao et al.^[1] evaluated recognition accuracy for 27 kinds of gestures with nine accelerometers and nine gyroscopes on a board and demonstrated the differences in

^{*1}: College of Information Science and Engineering, Ritsumeikan University, Japan

^{*2}: Graduate School of Engineering, Kobe University, Japan

^{*3}: PRESTO, Japan Science and Technology Agency, Japan

recognition accuracy by changing the number, positions, and kinds of sensors and the number and kinds of gestures. Agrawl et al.^[2] proposed a system that recognized the alphabet written in the air with a cell phone. Acceleration data were converted to spatial motion. They achieved 83% recognition accuracy by adhering to some restrictions. Chambers et al.^[3] proposed a video annotation method by gesture recognition using one accelerometer mounted on wrist since video annotation is difficult only from analyzing video. They use Hidden Markov Model (HMM)^[4] for recognition, resulting in only one mistake for 30 trials of three Kung-Fu martial art movements; Cuts, Elbow, and Punch. The system proposed by Junker et al.^[5] recognizes ten daily short actions, such as pushing a button and drinking, and achieves approximately 80% precision and recall. Innovative point of this study is that it partitions the stream of sensor into several segments that represent atomic human movement by using the sliding-window and bottom-up (SWAB) algorithm^[6]. The training data in these studies were collected before recognition without taking changes in gesture motions caused by user conditions into consideration.

In contrast to these studies, Liu et al.^[7] focused on changes in daily gesture motions. They recognized eight kinds of gestures including those in drawing a line or a circle (recommended by Nokia Research Institute), with a 3-axis accelerometer. They captured more than 4,000 samples for eight subjects over a long period, using dynamic time-warping (DTW)^[8] as a recognition algorithm. An accuracy of 98.6% was achieved by successively renewing the training data. However, a ground truth was required when renewing the training data, which did not fundamentally solve the problem. In addition, it was not clear whether the system was effective against changes in gesture motions throughout the day since training data were renewed once a day. Rabiner et al.^[10] proposed an speaker-independent speech recognition with DTW. This method conducts clustering for training data and creates a set of word reference templates for each word by grouping the replications of each word into a suitable set of clusters. This method is off-line approach which is conducted after all the training data have been collected. In the contrary, our approach is on-line which finds training data for the future data as one training data

sample comes.

Motion sensor based approaches and image sensor based approaches are better to recognize gestures in the environment where camera can be installed and used. Since we assume that gesture recognition is used anywhere including outdoor, it is hard to install image sensor to the environment and user body in order to capture user gestures. Therefore, we employed accelerometer-based approach.

Yao et al.^[11] proposed a method of personalizing gesture recognition models. This method learns a portfolio of classifiers on the training data such that at least one classifier is well-suited to any given user. At run time, personalization data is used to evaluate all classifiers in the portfolio and to select the best for the target user. They tested the method with image data recorded by the Microsoft Kinect, however the method is so general that it can deal with any kind of sensor data. This method personalizes models based on the similarity of profile such as height and age, therefore decay of movement within individual cannot be considered since how the gestures decay is not known nor modelled beforehand. We propose a method to find the convergence point of gestures at run time. Training data obtained through our method with their personalization method would personalize model considering difference between individuals and decay with time.

3. Preliminary experiment

We conducted a preliminary experiment to investigate the change in gesture motions over a long period.

3.1 Setup and procedure

Data on two kinds of gestures: *throwing a ball* and *drawing a star in the air*, were captured continuously 200 times in 3-second intervals for seven days for five male subjects 22–23 years old who had a 3-axis accelerometer attached to their wrist. The reason for this setup was that gesture motions would change as they were repeated many times and the days pass. The trajectory of *throwing a ball* gesture is a pitching a ball and the trajectory of *drawing a star* gesture is five-stroke lines starting from top to bottom left. To reduce errors due to individual interpretation, the instructions were not given verbally. Instead, one of the authors demonstrated the actual movement of each gesture. The subjects were asked to remain

stationary between each gesture to separate the data into gesture and non-gesture with ease. A lot of studies on gesture segmentation have been done thus far. Real-time stream monitoring algorithm has been proposed^[12]. We also have proposed gesture spotting algorithm while periodic activity such as walking or running by calculating auto-correlation function^[13]. However, gesture segmentation is an independent topic of gesture recognition and has generally been done by pushing button or standing still in studies on gesture recognition. Therefore, we asked the subject to remain stationary between each gesture in order to obtain correct data samples for investigation.

The change in gesture motions was investigated with the following procedure. First, 200 sequences for seven days for each gesture $X_{i,j}$ were collected, where i is an index of the sequences on a day ($1 \leq i \leq 200$) and j is a day ($1 \leq j \leq 7$). Then, the distances between training data $X_{i,1}$ and testing data $X_{i',j'}$ were calculated with a dynamic time-warping (DTW) algorithm and averaged for $1 \leq i' \leq 200$

$$\bar{D}_{i,j'} = \frac{1}{200} \sum_{i'=1}^{200} DTW(X_{i,1}, X_{i',j'}), \quad (1)$$

iterating for $1 \leq i \leq 200$ and for $1 \leq j' \leq 7$. The function $DTW(X, Y)$ denotes DTW distance between gesture sequences X and Y . The value of $\bar{D}$ shows the average distance of 200 test data when one sample of data on the first day is used for training. Small $\bar{D}$ means that the training data is similar to the test data and is discriminative for the future data. The detailed algorithm of DTW is explained in the next.

The accelerometer used was a wireless 3-axis accelerometer, made by Wireless Technologies Inc.^[9]. The sampling frequency was 50 Hz.

3.2 Dynamic time-warping algorithm

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 perform data-mining. A simple approach to measure similarity is to use the Euclidean distance, but the results are susceptible to temporal distortion, and the number of samples in two data sequences must be equal. A better approach is to use dynamic time-warping (DTW). The DTW algorithm can be used to calculate the

temporal non-linear elastic distance between two sequences, which may vary in time, and the number of both samples need not be equal. For example, it can be used to find the similarity for *draw a circle in the air* when the rotating speeds differed.

Hidden Markov Model (HMM)^[4] is also famous for a method to treat time-series data and some studies use HMM to recognize gestures. However, as claimed in the literature^[7], HMM requires large amount of training data to be effective. For personalized gesture recognition using accelerometer, it is hard to collect large amount of training data, and we aim to reduce the number of training data. DTW is the mainstream in this field, therefore we adopt DTW in this paper.

In recognition systems for walking or standing activities, SVM or decision tree are commonly used after extracting feature values such as mean, variance, and FFT coefficient. However, converting raw data to feature values misses information on time and deteriorates gesture recognition accuracy. Gestures are generally recognized with time-series data and It is hard to find feature values that have both discriminability and robustness for decay with time, therefore we calculate similarity of time-series data in this paper.

The algorithm works as follows. For the sake of simplicity, we assume there is data for one axis. Testing data $X = (x_1, \dots, x_m)$ and training data $Y = (y_1, \dots, y_n)$, with length m and n , respectively, are compared, an $m \times n$ matrix d is defined: $d(i, j) = |x_i - y_j|$. Next, warping path $W = (w_1, \dots, w_k)$, which is a path of pairs of indices of X and Y , is found. W meets three conditions.

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

To find the path with the lowest cost that meets these conditions, the following steps are applied.

1. Initialization
 $dtw(0, 0) = 0$
 $dtw(i, 0) = \infty$ for $i = 1, \dots, m$
 $dtw(0, j) = \infty$ for $j = 1, \dots, n$

2. Calculation

Do for $i = 1$ to m Do for $j = 1$ to 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 $dtw(m, n)/n$ is $DTW(X, Y)$ in Eqn. 1. The returned $dtw(m, n)$ is divided by n since 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. DTW algorithm can be used for multiple axes of accelerometers and gyroscopes. The DTW calculation is carried out for three axes by utilizing $d(i, j) = \sum_{axis} |x_i - y_j|$.

3.3 Results

The results for the *throwing a ball* and *drawing a star in the air* gestures for a subject are plotted in Figures 1 and 2. The vertical axis indicates the average DTW distance obtained with Eqn. 1 and the horizontal axis indicates an index of the sequence for the training data, i in Eqn. 1.

For example, the line “day 2” in Figure 1 indicates the average DTW distances between the 200-sample data on the second day and the i th data on the first day $\bar{D}_{i,2}$, and i corresponds to the horizontal axis for the *throwing a ball* gesture. The distance between the same data was neglected in the calculations for day 1. The large value for the DTW distance indicates that the data at the point are different to the 200-sample data and that these are not suitable for training data. Moreover, the large fluctuations on the line indicates that the data around the point are not stable and that their quality varies vastly from point to point.

3.4 Discussion

From the results, the line graphs were categorized into two types: unvaried lines and decreasing lines. Figures 1 and 2 show examples of decreasing line and unvaried line, respectively. The DTW distances at the beginning of both lines are larger than those in the latter half, indicating that gestures

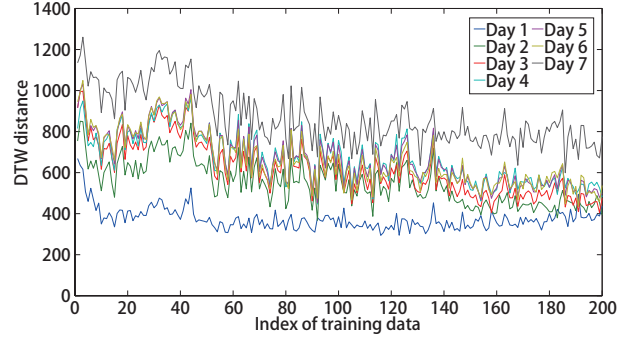


Fig. 1 Average DTW distance for *throwing a ball* gesture for subject 1.

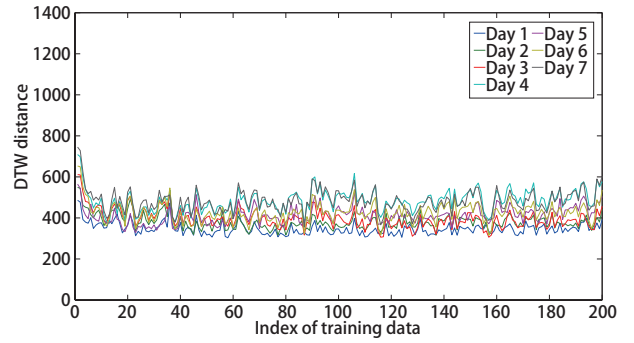


Fig. 2 Average DTW distance for *drawing a star* gesture for subject 1.

had changed through the experiment and that distances were larger when using the first few samples as training data as conventional method does. Eventually, the distance increases in relation to the other gestures, which leads to misrecognition. Moreover, the DTW distance converged as the number of trials increased, as shown in Figure 1. This is because the form of the subject’s gestures stabilized since the subject had become accustomed to the gestures or gesture forms that varied with those that had less fatigue. Most of the results for the *drawing a star in the air* gesture were unvaried lines, as shown in Figure 2. This is because *drawing a star* gesture is performed on a two-dimensional plane; therefore, the forms are stabilized due to small degree of freedom. The fluctuations in the lines in Figures 1 and 2 decreased as the index of training data increased, indicating the gesture motions were stabilized.

From the results, we assume that the distance converged and its fluctuation decreased at the best point of the data for the training. Accurate training data could be obtained by stopping data collection on the point where the line of the distance converges and stabilizes for decreasing lines and by stopping data

collection in the early phases for unvaried lines.

We calculated the recognition accuracy to conduct further investigations to see the effect of change on gesture motions. The subjects performed eight kinds of gestures that could be performed in video games and daily life to execute a command: *drawing a circle*, *drawing a triangle*, *drawing a square*, *punching*, *chopping*, *slapping*, *drawing a horizontal line*, and *drawing a vertical line* ten times each. These data and the data for the *throwing a ball* and *drawing a star* gestures for the first ten sequences of 200 sequences on the first day were used for training. Recognition took place for the 200 sequences for the *throwing a ball* and *drawing a star* gestures as follows. The DTW distances between testing data to be recognized and all the training data were calculated and data whose DTW distance is the shortest overall becomes a result. Tables 1 and 2 show the recognition results for the *throwing a ball* and *drawing a star* gestures, respectively, for 1,000 samples (200 samples $\times$ 5 subjects) per day. The rows indicate the number of outputs for the gestures.

The recognition accuracy for the *throwing a ball* gesture was over 99.7%. The recognition accuracy for the *drawing a star* gesture on the first day was 99.8%, while the recognition accuracy dropped to 78.9% on the seventh day. These results indicate that retaining the first few samples for training causes misrecognition as the conventional method does, which deteriorates interface usability or system accuracy using gesture recognition technology.

4. Proposed method

The results from the preliminary experiment clarified that the point where the average distance and its variance were the smallest was the most appropriate for training data. However, it is not possible to find the optimum point without future data. Though the quality of training data improves as the number of training data collected increases, the speed of improvement in quality would slow at a certain point. We propose two methods of finding the appropriate point for training data in real time by stopping collecting data when the training data are stabilized.

4.1 Proposed method 1

We confirmed that the point at which average DTW distance and its variance were the smallest is the most appropriate for training data. First, this

method collects gesture data n times, then calculates the average μ_n and variance σ_n of DTW distances between gesture data X_{n-T} and the latest T -sample gesture data X_{n-T+i} for $1 \leq i \leq T$ as follows.

$$\mu_n = \frac{1}{T} \sum_{i=1}^T DTW(X_{n-T}, X_{n-T+i}) \quad (2)$$

$$\sigma_n = \frac{1}{T} \sqrt{\sum_{i=1}^T \{DTW(X_{n-T}, x_{n-T+i}) - \mu_n\}^2} \quad (3)$$

The system determines that the gesture motion has converged and stops collecting training data when both μ_n and σ_n satisfy four conditions.

- $\mu_n < \mu_0$
- $\sigma_n < \sigma_0$
- $\mu_{n-i} < \mu_n \times (1 + \alpha)$ for $i = 1, \dots, l$
- $\sigma_{n-i} < \sigma_n \times (1 + \alpha)$ for $i = 1 \dots, l$

We set $\alpha = 0.01$, $T = 10$, and $l = 9$, and the proposed method 1 requires at least 20 gesture samples since 10 samples are used for calculating the average and variance and 10 samples are used for accessing the four conditions. These values were determined from our pilot study.

This method stops collecting training data around the point where gesture motions are stabilized and similar each other.

4.2 Proposed method 2

From the preliminary experiment, we confirmed that gesture motions were diverse in the first few samples and were going to remain unvaried as gestures were repeated. We assumed that gesture motions were meant to be stable and outlying motions barely appeared when the distances between the first few samples and the current sample were continuously close.

The algorithm is as follows. When the user collected gesture data n times, this method calculates the DTW distances between gesture data $X_{n-T'+i}$ ($0 \leq i \leq T'$) and the first T' -sample gesture data X_j for $1 \leq i \leq T'$ and finds the smallest distance among them

$$dist_{n-T'+i} = \min_{j=1}^{T'} DTW(X_{n-T'+i}, X_j), \quad (4)$$

then average μ'_n of $dist_{n-T'+i}$ for $1 \leq i \leq T'$ is calculated.

$$\mu'_n = \frac{1}{T'} \sum_{i=1}^{T'} dist_{n-T'+i} \quad (5)$$

Table 1 Recognition results for *throwing a ball* gesture.

Day	Throw	Star	Circle	Triangle	Square	Punch	Chop	Slap	Vertical	Horizontal	Accuracy
1	999	0	0	0	0	0	0	1	0	0	0.999
2	960	0	0	0	0	0	0	40	0	0	0.960
3	951	0	0	0	0	18	0	31	0	0	0.951
4	966	0	0	0	0	8	0	25	1	0	0.966
5	984	0	0	0	0	3	0	13	0	0	0.984
6	984	0	0	0	0	6	0	10	0	0	0.984
7	856	0	0	0	0	0	18	125	1	0	0.856

Table 2 Recognition results for *drawing a star* gesture.

Day	Throw	Star	Circle	Triangle	Square	Punch	Chop	Slap	Vertical	Horizontal	Accuracy
1	0	999	0	1	0	0	0	0	0	0	0.999
2	0	1000	0	0	0	0	0	0	0	0	1.000
3	0	991	0	9	0	0	0	0	0	0	0.991
4	0	997	0	3	0	0	0	0	0	0	0.997
5	0	1000	0	0	0	0	0	0	0	0	1.000
6	0	1000	0	0	0	0	0	0	0	0	1.000
7	0	848	0	152	0	0	0	0	0	0	0.848

The system stops collecting data for training when μ'_n meets the following condition.

- $|\mu'_{n-i} - \mu'_n| < \mu'_n \times \beta$ for $i = 1, \dots, l'$

We set $\beta = 0.1$, $T' = 5$, and $l' = 4$, and the proposed method 2 requires at least 10 gesture samples since five samples are used for calculating the average and five samples are used for determining the condition above. These values were determined from our pilot study.

This method stops collecting data around the point where gesture motions are stabilized in the range of the values of the first few samples, resulting in excluding outlying values.

5. Evaluation

5.1 Environment

We calculated the DTW distance between the 200-sample data for *throwing a ball* and *drawing a star* gestures captured in the preliminary experiment and the training data obtained with two proposed methods and three comparison methods to evaluate the effectiveness of our proposed methods: Comparison method 1 uses the first five samples on the first day for training, which is the most simple way commonly used in the conventional method and requires the least number of training data, therefore it would perform worst of all. Comparison method 2 uses the data at the optimum point for training which can be obtained after 200 samples had been captured. Comparison method 2 finds the optimum point by calculating the average DTW distance between n th data and 200-sample data $D_{comp2,n} = \frac{1}{200} \sum_{i=1}^{200} DTW(X_n, X_i)$. Then the point where

$D_{comp2,n}$ becomes less than 110% of $D_{comp2,200}$ at the first time is obtained as an optimum point. Comparison method 2 is the baseline method since it uses all the data. Comparison method 3 uses the the first five samples for training and updates the training data every day, which would perform well but forces tasks on the users.

5.2 Results

Tables 3 and 4 list the average DTW distance for *throwing a ball* and *drawing a star* gestures for the proposed methods denoted by P1 and P2 and the comparison methods denoted by C1, C2, and C3. The values in the top line for each subject indicate the number of training data obtained until each method stopped collecting data. The number of data for C1 is five for all the subjects since it uses the first five samples on the first day. The number of data for C3 is also five since it uses the first five samples on each day.

Seeing the results for each subject for *throwing a ball* gesture, for subjects 1, 2, and 5, the DTW distance for P1 was larger than that for C2, while the number of data collected is smaller. For subject 3 and 4, P2 stopped collecting data after the point obtained with C2, therefore the distance for P2 was smaller than that of C2. For the results of *drawing a star* gesture, the distance of P1 for subjects 2 and 5 were larger than those for C2. The results for subjects 1, 3, and 4 for P1 were comparable to those for the C2. The distance for P2 for *throwing a ball* gesture was larger than that for C2 for subjects 1, 2, 4, and 5. For subject 3, the distance for P2 was smaller

than that for C2 since P2 collected large number of samples than C2. For the results of *drawing a star* gesture, the distance for all the subjects for P2 was larger than that for C2. For all the subjects, the distances for C3 are equivalent to that for both P1 and P2 on the first few days, while the distances for C3 take advantage over both P1 and P2 since C3 updates training data daily.

Comparing the proposed methods (P1 and P2) and the comparison methods (C1, C2, and C3), the DTW distances for P1 and P2 were smaller than that for C1. However, the distances for P1 and P2 were larger than that for C2 and C3 in some cases. This is because the proposed methods accidentally stopped collecting data at the point where the data were continuously stable but had not converged. Moreover, the DTW distances for P1 and P2 get larger as day passes, while the distance for C3 is kept small, which means that updating the training data every day improves accuracy.

Comparing P1 and P2, the distance for P2 was equal to or larger than that for P1, while the number of training data collected was smaller. This is because similar forms of the gesture continuously appeared before real convergence; therefore, the system failed to collect data that appeared in the subsequent processes. P2 was able to reduce the number of training data collected compared with P1, however we have to consider the case where gesture changes with multiple convergence points.

At last, we calculated the recognition accuracy for *throwing a ball* gesture and *drawing a star* gesture with the training data found by P1 and P2. Ten samples including and before the found point were used for training. For *drawing a circle*, *drawing a triangle*, *drawing a square*, *punching*, *chopping*, *slapping*, *drawing a horizontal line*, and *drawing a vertical line* gestures, the first ten samples of 200 sequences on the first day were used for training. Recognition took place for the 200 sequences for the *throwing a ball* and *drawing a star* gestures as follows. Tables 5 through 8 show the recognition results for the *throwing a ball* and *drawing a star* gestures with P1 and P2 for 1,000 samples (200 samples $\times$ 5 subjects) per day. The rows indicate the number of outputs for the gestures.

Comparing the results for P1 and P2, P1 and P2 showed higher accuracy for *throwing a ball* gesture

Table 3 Average DTW distance and number of data collected for *throwing a ball* gesture.

Subject	P1	P2	C1	C2	C3
# of data	48	20	5	158	5*
Day 2	737	750	685	414	425
Day 3	802	890	776	474	465
Day 4	838	848	792	521	382
Day 5	852	872	841	487	434
Day 6	866	903	815	498	379
Day 7	1070	1106	1045	715	394
# of data	41	15	5	112	5*
Day 2	907	1024	1155	776	991
Day 3	1080	1053	1192	1204	758
Day 4	970	997	1083	1293	969
Day 5	1033	961	1063	1098	821
Day 6	1337	950	1011	1090	714
Day 7	1354	926	1012	996	703
# of data	123	165	5	68	5*
Day 2	550	756	573	678	449
Day 3	558	590	697	636	533
Day 4	514	632	660	715	510
Day 5	656	607	669	719	507
Day 6	648	593	740	841	587
Day 7	680	643	814	900	561
# of data	96	21	5	87	5*
Day 2	720	845	917	773	795
Day 3	731	846	957	859	585
Day 4	799	843	928	793	558
Day 5	851	920	1074	855	713
Day 6	873	950	1012	857	492
Day 7	842	915	1029	888	523
# of data	58	13	5	122	5*
Day 2	754	706	908	814	734
Day 3	699	802	972	763	651
Day 4	718	873	1066	882	874
Day 5	886	849	1062	801	765
Day 6	883	887	1129	1005	911
Day 7	806	896	1132	840	786

* First five samples per day.

from day 2 to day 5, while P2 showed higher accuracy than P1 for day 6 and day 7, which accords with the result of distance calculation in Table 3. For the results of *drawing a star* gesture, P1 is slightly better than P2 due to the bad training data of subject 5 which showed larger DTW distance as shown in Table 4. Comparing the results of C1 shown in Table 1 and 2, the proposed methods show equivalent accuracies until day 6. However, the accuracy for the comparison method vastly drops on the day 7, while the proposed methods maintain high degree of accuracy.

5.3 Summary

Summarizing the results, both P1 and P2 performed well. There is a trade-off between the quality of training data and the number of samples collected. The quality of training data for P1 was slightly bet-

Table 5 Recognition results for *throwing a ball* gesture with proposed method 1.

Day	Throw	Star	Circle	Triangle	Square	Punch	Chop	Slap	Vertical	Horizontal	Accuracy
2	997	0	0	0	0	0	0	3	0	0	0.997
3	977	0	0	0	0	18	0	5	0	0	0.977
4	985	0	0	0	0	12	0	3	0	0	0.985
5	964	0	0	0	0	9	0	27	0	0	0.964
6	889	0	0	0	0	101	0	10	0	0	0.889
7	804	0	0	0	0	37	20	138	1	0	0.804

Table 6 Recognition results for *drawing a star* gesture with proposed method 1.

Day	Throw	Star	Circle	Triangle	Square	Punch	Chop	Slap	Vertical	Horizontal	Accuracy
2	0	1000	0	0	0	0	0	0	0	0	1.000
3	0	1000	0	0	0	0	0	0	0	0	1.000
4	0	1000	0	0	0	0	0	0	0	0	1.000
5	0	1000	0	0	0	0	0	0	0	0	1.000
6	0	1000	0	0	0	0	0	0	0	0	1.000
7	0	985	0	15	0	0	0	0	0	0	0.985

Table 7 Recognition results for *throwing a ball* gesture with proposed method 2.

Day	Throw	Star	Circle	Triangle	Square	Punch	Chop	Slap	Vertical	Horizontal	Accuracy
2	996	0	0	0	0	0	0	4	0	0	0.996
3	988	0	0	0	0	10	0	2	0	0	0.988
4	987	0	0	0	0	9	0	4	0	0	0.987
5	995	0	0	0	0	1	0	4	0	0	0.995
6	997	0	0	0	0	2	0	1	0	0	0.997
7	903	0	0	0	0	0	7	90	0	0	0.903

Table 8 Recognition results for *drawing a star* gesture with proposed method 2.

Day	Throw	Star	Circle	Triangle	Square	Punch	Chop	Slap	Vertical	Horizontal	Accuracy
2	0	1000	0	0	0	0	0	0	0	0	1.000
3	0	936	0	64	0	0	0	0	0	0	0.936
4	0	944	0	56	0	0	0	0	0	0	0.944
5	0	999	0	1	0	0	0	0	0	0	0.999
6	0	996	0	4	0	0	0	0	0	0	0.996
7	0	998	0	2	0	0	0	0	0	0	0.998

ter than that for P2, while the data for the P2 was collected more rapidly than those for P1, however the performance depends on the user and gesture; therefore, it would be better to use both methods simultaneously and use the data point found first.

Gesture movement that has changed due to fatigue or forgetfulness may become similar to another gesture, e.g. distorted triangle assimilates circle. Both conventional method and the proposed methods cannot work well for the case. In case of incompatibility of gestures, gesture to be recognized or design of the application should be changed.

6. Conclusion

We proposed methods of finding appropriate points for training data by calculating the DTW distance between collected data in real time. Our method found training data that performs as well

as the method using all data including future data, while the number of samples collected could be reduced. We plan to evaluate other kinds of gestures and conduct further investigations into the effect of fatigue and forgetfulness in future work.

Acknowledgment

This research was supported in part by a Grant in aid for Precursory Research for Embryonic Science and Technology (PRESTO) from the Japan Science and Technology Agency.

Reference

- [1] K. Murao, T. Terada, A. Yano, and R. Matsukura: Evaluating Gesture Recognition by Multiple-Sensor-Containing Mobile Devices, *Proceedings The of International Symposium on Wearable Computers (ISWC 2011)*, pp. 55–58 (2011).
- [2] S. Agrawal, I. Constandache, S. Gaonkar, R.

Determining a Number of Training Data for Gesture Recognition Considering Decay in Gesture Movements

Table 4 Average DTW distance and number of data collected for *drawing a star* gesture.

Subject	P1	P2	C1	C2	C3
# of data	22	30	5	48	5*
1	Day 2	458	382	465	357
	Day 3	509	433	530	374
	Day 4	491	433	562	416
	Day 5	370	370	397	384
	Day 6	457	394	510	374
	Day 7	473	451	538	461
	Day 7	324			
2	# of data	26	26	5	48
	Day 2	743	743	787	707
	Day 3	943	943	894	739
	Day 4	987	987	983	919
	Day 5	828	828	873	779
	Day 6	852	852	927	871
	Day 7	786	786	862	786
3	# of data	53	48	5	69
	Day 2	701	799	984	643
	Day 3	701	827	1017	581
	Day 4	623	820	999	545
	Day 5	748	912	1085	720
	Day 6	730	986	1238	682
	Day 7	852	1093	1293	792
4	# of data	55	21	5	60
	Day 2	667	723	868	662
	Day 3	839	681	832	614
	Day 4	547	604	822	589
	Day 5	563	631	845	611
	Day 6	702	637	836	624
	Day 7	752	836	1026	768
5	# of data	96	15	5	103
	Day 2	617	1136	1047	661
	Day 3	952	1444	1425	678
	Day 4	939	1440	1446	598
	Day 5	757	1279	1215	585
	Day 6	759	1325	1256	580
	Day 7	692	1321	1178	679

* First five samples per day.

- Choudhury, K. Caves, and F. DeRuyter: Using Mobile Phones to Write in Air, *Proceedings of The International Conference on Mobile Systems, Applications, and Services (Mobisys 2011)*, pp. 15–28 (2011).
- [3] G.S. Chambers, S. Venkatesh, G.A.W. West, and H.H. Bui: Hierarchical Recognition of Intentional Human Gestures for Sports Video Annotation, *Proceedings of The International Conference on Pattern Recognition (ICPR 2002)*, pp. 1082–1085 (2002).
- [4] L.R. Rabiner and B.H. Juang: A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of The IEEE*, Vol. 72, No. 2, pp.257–286 (1989).
- [5] H. Junker, O. Amft, P. Lukowicz, and G. Tröster: Gesture Spotting with Body Worn Inertial Sensors to Detect User Activities, *Journal of Pattern Recognition*, Vol.41, No. 6, pp. 2010–2024 (2008).
- [6] E. Keogh, S. Chu, D. Hart, and M. Pazzani: An Online Algorithm for Segmenting Time Series, *Proceedings of The International Conference on Data Mining (ICDM 2001)*, pp.289–296 (2001).

- [7] J. Liu, Z. Wang, L. Zhong, J. Wiekramasuriya, and V. Vasudevan: uwave: Accelerometer Based Personalized Gesture Recognition and Its Applications, *Proceedings of The International Conference on Pervasive Computing and Communication (PerCom 2009)*, pp. 1–9 (2009).
- [8] C. Myers and L.R. Rabiner: A Comparative Study of Several Dynamic Time-warping Algorithms for Connected Word Recognition, *The Bell System Technical Journal*, Vol. 60, pp. 1389–1409 (1981).
- [9] Wireless Technologies, Inc.: <http://www.wireless-t.jp/>.
- [10] L.R. Rabiner and J.G. Wilpon: Speaker-Independent Isolated Word Recognition for a Moderate Size (54 word) Vocabulary, *IEEE Transaction on Acoustics, Speech, and Signal Processing*, Vol. ASSP-27, No. 6, pp. 583–587 (1979).
- [11] A. Yao, L.V. Gool, and P. Kohli: Gesture Recognition Portfolios for Personalization, *Proceedings of The International Conference on Computer Vision and Pattern Recognition (CVPR 2014)*, pp. 1923–1930 (2014).
- [12] Y. Sakurai, C. Faloutsos, and M. Yamamuro: Stream Monitoring under the Time Warping Distance, *Proceedings of IEEE International Conference on Data Engineering (ICDE 2007)*, pp. 1046–1055 (2007).
- [13] K. Murao and T. Terada: A Motion Recognition Method by Constancy-Decision, in *Proceedings of The International Symposium on Wearable Computers (ISWC 2010)*, pp. 69–72 (2010).

(received Sep. 29, 2014, revised Feb. 15, 2015)

Biography

Kazuya Murao (Member)



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

Gaku Yoshida



Gaku Yoshida is in West Japan Railway Company. He received B.Eng and M.Eng from Kobe University in 2011 and 2013, respectively. Mr. Yoshida had been interested in activity recognition.

Tsutomu Terada (Member)



Tsutomu Terada is an Associate Professor at Graduate School of Engineering, Kobe University, Japan. He received 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, IPSJ, and IEICE.

Masahiko Tsukamoto (Member)



Masahiko Tsukamoto is a Professor at Graduate School of Engineering, Kobe University, Japan. He received B.Eng., M.Eng., and Ph.D. degrees from Kyoto University in 1987, 1989, and 1994, respectively. Prof. Tsukamoto is working on wearable computing and ubiquitous computing. He is a member of ACM, IEEE, and IPSJ.