Sensors and Materials, Vol. 34, No. 8 (2022) 3001–3015
MYU Tokyo

S & M 3019

High-F1-score Recognition of Input Gestures while Holding Smartphone by Reducing False Detection Due to Walking Noise

Ryo Katsuma* and Tomoaki Amiya

Osaka Metropolitan University, 1-1 Gakuencho, Nakaku, Sakai, Osaka 599-8531, Japan

(Received April 10, 2022; accepted July 4, 2022)

Keywords: gesture recognition, walking noise, accelerometer, holding smartphone

Gesture input using the acceleration sensor of a smartphone is a promising new input method. The target input gestures in this paper are movements of a user's hand holding a smartphone. However, if parameter tuning is performed to improve the recognition accuracy of input gestures while stationary, erroneous detection at the start of walking will increase. On the other hand, if parameter tuning is performed to reduce false detection at the start of walking, the recognition accuracy of input gestures while stationary is lowered. Thus, there is a trade-off problem. In this paper, we propose a gesture recognition method to reduce erroneous recognition by combining a gesture detection method that uses similarity based on dynamic time warping (DTW) (TD) and a gesture classification method that also includes walking data as a candidate (CD). We conducted evaluation experiments with nine subjects. As a result, we confirmed that false detection at the start of walking can be eliminated using the proposed method. By verification using *t*-test, we confirmed that the F1-score of the proposed method was significantly higher than that of CD.

1. Introduction

In recent years, computers such as smartphones have become smaller, and their convenience has improved because they can be easily carried and worn. Various smartphone operation methods are being researched, such as voice operation, eye tracking operation, and gesture operation. In this paper, for gesture operation, we focus on gesture input with the smartphone held in the user's hand. Gesture input using the acceleration sensor of a smartphone is a promising new input method without the need to look at the screen. Although gesture input does not require users to look at the screen, it is desirable for the user to be stationary during input to avoid input while walking.

However, when the input is completed and the user starts walking, many false detections occur due to the large noise generated by walking. To prevent this false detection, parameters such as the acceleration value threshold for gesture detection can be adjusted so that a certain amount of acceleration can be ignored. However, this parameter adjustment causes many gesture

*Corresponding author: e-mail: katsuma@omu.ac.jp
https://doi.org/10.18494/SAM3938

ISSN 0914-4935

input failures. Therefore, improving the recognition accuracy of gesture input and reducing false detections have a trade-off relationship.

In gesture input, it is necessary to determine which gesture was performed and when on the basis of the acceleration measured by the smartphone. In other words, it is necessary to first solve the decision problem of gesture existence, and then, if a gesture has been made, solve the classification problem of which gesture it is. Therefore, in this paper, gesture recognition is divided into the following two problems, as overviewed in Fig. 1.

• Gesture detection: detects when a gesture is performed
• Gesture classification: classifies which gesture is performed

In gesture input, the following three types of error are important.

• False positive (for gesture detection): a detected gesture is output when no gesture is performed
• False negative (for gesture detection): no result is output when a gesture is performed
• False classification (for gesture classification): an incorrect gesture is output

Generally, in gesture detection, the existence of a gesture is determined by setting a threshold value and detecting whether the acceleration value exceeds the threshold.[1,2] Setting a threshold to make gestures easier to detect reduces false negatives but increases false positives due to noise. On the other hand, setting a threshold to make gestures harder to detect reduces false positives but increases false negatives. This results in the above-mentioned trade-off relationship between recognition accuracy and false detection.

In gesture classification, chronological acceleration data of candidate gestures assumed as inputs is measured in advance.[3,4] Then, by various classification methods, the gesture most similar to the input acceleration data is output as the solution. When there are candidate gestures that behave similarly to other candidates, they tend to be misclassified. Furthermore, if a part of the walking noise resembles a candidate gesture, false positives will occur.

In this paper, we propose a method to detect and classify gestures at the same time by adding typical walking noise to candidate gestures. In the proposed method, the threshold value of the acceleration, which has often been used in previous studies, is not used in the gesture detection. Instead, the proposed method detects and classifies gestures when they are significantly similar to a single candidate gesture. A threshold is set in this “significantly” part. If the input data is classified as walking noise, it means that no gesture is detected.

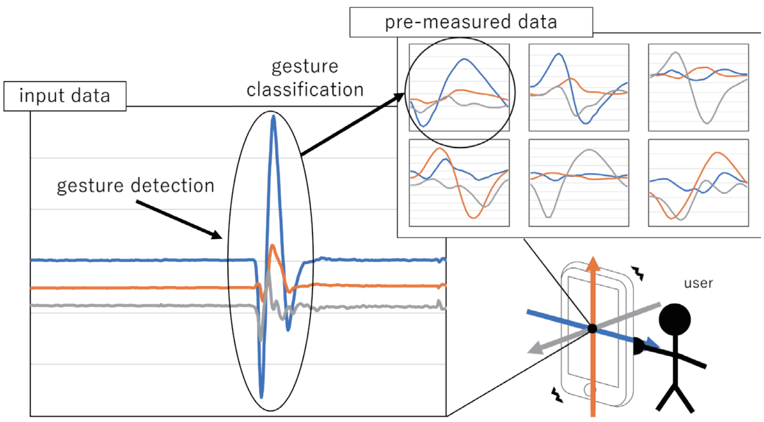


Fig. 1. (Color online) Overview of gesture recognition.

2. Related Works

2.1 Walking detection

False positives can be eliminated by setting the smartphone so that it cannot be operated while walking. However, most of the previous studies on walking detection did not require immediacy, and it is difficult to detect walking at the moment a user starts walking. Therefore, false positives occur immediately after the start of walking. There is also known to be an error of about 5 s in movement detection using GPS.[5]

2.2 Gesture detection by accelerometer

Acceleration thresholds are used for anomaly detection in automotive airbags and railroad tracks. Impact can be detected by setting a threshold value for continuously input acceleration values and the velocity can be calculated from these values.[6] Similarly, in gesture detection, an input gesture is detected by setting a threshold value for the acceleration values acquired from an accelerometer.[1] There is a trade-off between false positives and false negatives, and it is difficult to reduce both at the same time. The F-measure can be used as an index to evaluate the recognition accuracy in such a case, and the F1-score has been used in many studies.[7] In this paper, we both output the correct gesture for the gesture-performed data and ignore the walking data, and the accuracy is evaluated using the F1-score.

2.3 Gesture classification by accelerometer

When chronological acceleration data is given, the basic method of gesture recognition for finding the correct gesture from many candidate gestures is to measure the chronological acceleration data of the candidate gestures assumed to be input in advance and to find the most similar candidate gesture.[2–4]

In such gesture classification, the similarity calculation of chronological data is performed. The typical algorithms are angular metric for shape similarity (AMSS),[8] multi-dimensional time-series approximation with the use of local features at thinned-out keypoints (A-LTK),[9] hidden Markov model (HMM),[10,11] and dynamic time warping (DTW).[2,12,13] Even if the same gesture is performed several times, the gesture speed will change each time. Since the number of sample values of input data and the acceleration value vary, it is difficult to obtain the degree of similarity with premeasured data. Regarding this problem, DTW can be used to calculate the similarity degree by eliminating the difference in the number of samples and acceleration values by expanding and contracting the time for two chronological data. In this paper, DTW is used to calculate the similarity degree between the premeasured data and the input data.

2.4 Comprehensive method of gesture recognition

Izuta *et al.* used a DTW similarity calculation for gesture detection.[3] They found that a gesture can be recognized without waiting for its completion by repeatedly comparing the similarity between the input data and the premeasured data. In our previous study,[14] we proposed a more accurate DTW-based method in which the user is allowed to perform two gestures. This method achieved an F1-score of 98.7%. In this paper, we propose a method to further improve this highly accurate result while using only one gesture by adding walking noise to the premeasured data.

However, in our previous study,[14] we also showed that increasing the number of candidate gestures caused the accuracy to decrease. Figure 2 shows the F1-scores when considering 2 to 13 of the candidate gestures shown in Table 1. The F1-score exceeded 99% for a total of two candidate gestures but was only 96% when there were seven candidate gestures and 92% when there were 13 candidate gestures. Therefore, a concern of the proposed method is that increasing the number of candidate gestures will cause a slight decrease in the F1-score.

Many methods of gesture recognition use feature-based machine learning to improve the accuracy of gesture classification.[15–17] Typical methods are the k-nearest neighbor algorithm (k-NN), support vector machine (SVM), and random forest (RF). The proposed method was compared with these methods as reported in Sect. 6.

2.5 Main related works and position of our study

There have been many studies of gesture recognition, the more relevant ones being on accelerometer-based hand gesture recognition. Murao and Terada proposed a method to recognize whole-body gestures including punch motion by using three types of wearable sensor.[18] This study achieved high recognition accuracy in several noisy environments. Zhang *et al.* proposed a framework for hand gesture recognition based on wrist-mounted sensors through the information fusion of a three-axis accelerometer and multichannel electromyography (EMG).[19] However, these studies assumed the use of sensors not commonly used in ordinary life. Our research targets only a smartphone and does not use EMG, hand gloves with sensors, or any extra sensors.

Gupta *et al.* proposed a method to recognize continuous hand gestures by using accelerometer and gyroscope sensors for human–machine interaction.[20] Chu *et al.* proposed an accelerometer-based method to recognize 11 types of hand gesture to control household electric appliances.[21]

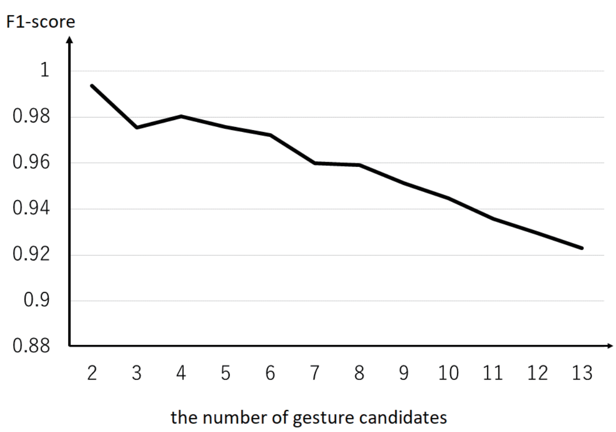
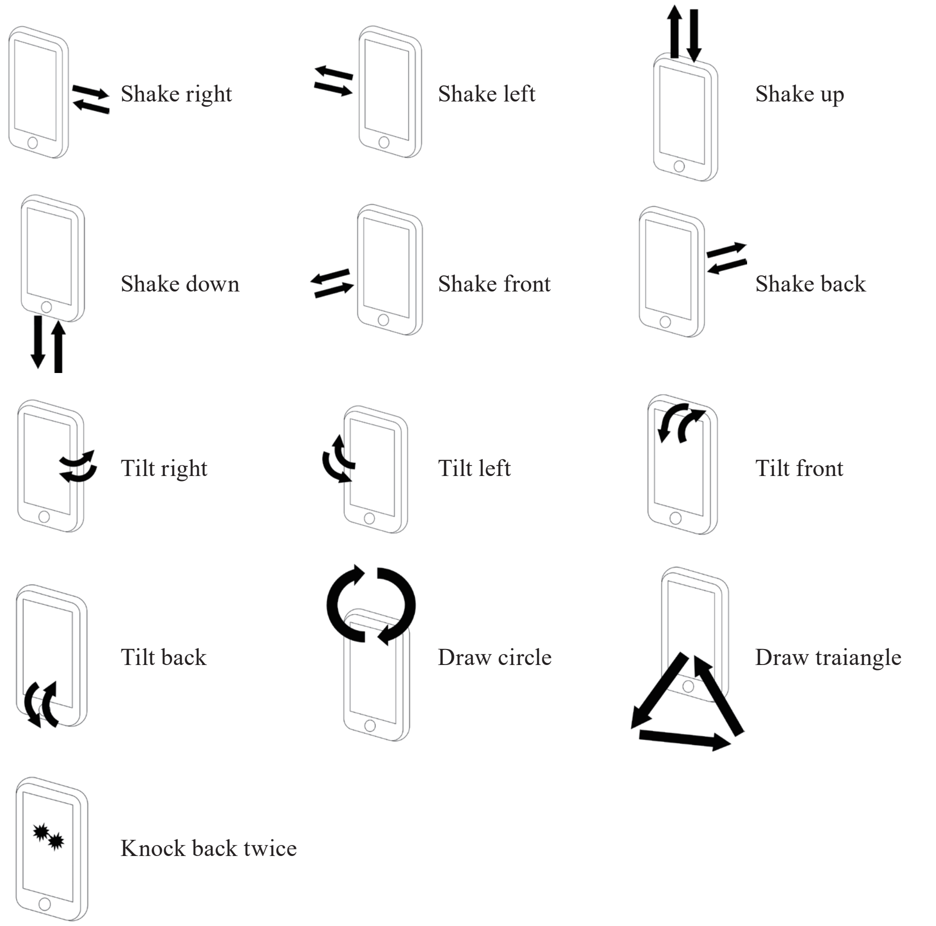


Fig. 2. F1-scores for different numbers of candidate gestures.

Table 1
Thirteen target candidate gestures.



Garcia-Ceja *et al.* proposed hand gesture recognition while holding a smartphone.(22) This method achieved 93.8% accuracy with 10 types of gesture. Agrawal *et al.* proposed a method to recognize letters drawn by a user's hand while holding a mobile phone.(23) However, these studies did not focus on changes in environmental noise.

The environment in which gestures are performed can be roughly divided into noiseless and noisy environments. Many studies have focused on gesture recognition in both environments. It is possible that the best performance can be achieved by switching between the most appropriate method depending on the environment. To realize this switching, it is necessary to know when the environment has changed. In our research, we are attempting to judge the switch from a noiseless environment to a noisy environment by adding this switch into the gestures. In this paper, we focused on the start of walking as a typical example of a switch in the environment. Walking is a very important behavior because it often causes switches to noisy environments such as when climbing stairs and getting on a train or a bus.

3. Problem Definition

Our goal is to recognize gestures from the acceleration values acquired by the accelerometer of the smartphone in the user's hand and use them as input operations for applications such as music players. In this section, we describe assumptions and the input and output of this problem.

The user holds the smartphone with one hand and performs a gesture while not walking. When the user starts walking, they hold the smartphone with one hand and do not input any gestures.

The input of this problem is the premeasured data and the data extracted with window size w from the continuous acceleration data. Here, w represents the number of samples and is sufficiently large for each gesture to be performed. The data is periodically extracted (e.g., at each sampling). Let $A = (a_{t0}, ..., a_{tw})$ denote the data extracted from time t_0 to time t_w.

The output is one of the gestures in the premeasured data or no gesture. Let N denote the number of gesture types and G_i ($i = 1, 2, 3, ..., N$) represent the gestures.

Premeasured data B_i measured in advance is constructed from chronological acceleration data $B_i = (b_0, ..., b_{ui})$. Here, u_i represents the number of samples, and all u_i ($i = 1, 2, 3, ..., N$) are smaller than the window size w. A single sample of extracted data a_i and the premeasured data b_i are vectors with three parameters, because acceleration values are obtained in the x, y, and z directions. Let (a_{ix}, a_{iy}, a_{iz}) and (b_{ix}, b_{iy}, b_{iz}) denote the contents of a_i and b_i, respectively.

We target the types of gesture that can be performed while holding a smartphone with one hand, with the hand returning to the original position after completing the gesture. Complicated gestures are undesirable because they increase the memory load on the user. Finally, we adopted 13 gestures that are relatively easy to perform (Table 1), assuming that the target application will need no more than 13 types of gestures. For example, in the case of a general music player, five types of input are required for the operations of play/stop, back, next, volume up, and volume down. In addition, three types of auxiliary operations are assumed, such as mute on/off, random on/off, and repeat on/off. In this case, a total of eight input types are required, so the 13 target input types are sufficient.

4. DTW-based Gesture Classification Method

4.1 DTW

DTW quantifies the similarity between two chronological data. The input is two acceleration chronological data P and Q. The output is $D(P, Q)$, which represents the distance between P and Q. If the two data are similar, then $D(P, Q)$ is small, and if they are different, it is large. Since DTW is a well-known algorithm used for gesture classification in many studies, details are omitted here.[2,12,13]

4.2 Baseline gesture classification method

In many studies, DTW is mainly used to calculate which gesture was performed on the basis of the assumption that a certain gesture has been performed. Here, we describe the basic method of gesture classification by DTW. Note that nongesture acceleration data (e.g., walking noise data) is not expected as the input. Hereafter, we focus on how to detect and classify the extraction data, and we refer to a single extraction data as the input data.

$$\text{OUTPUT: } B_I$$
$$\text{s.t. } D\left(A, B_I\right) = \min\left\{D\left(A, B_i\right)\right\}\left(i = 1, 2, 3, \ldots, N\right) \tag{1}$$

Similarities with the input data A are calculated for all premeasured data of candidate gestures. The distance $D(A, B_i)$ calculated using DTW is used as the similarity, where the smaller the distance, the greater the similarity of the two data. Therefore, B_I, which is the premeasured data nearest to the input data, is output as G_i.

5. Proposed Method

Conducting gesture detection and gesture classification at the same time without assuming that the input data includes gestures is called gesture recognition. First, two gesture recognition methods, which are the basis of the proposed method, are outlined in this section. Both methods have the common feature that they output the most similar premeasured gesture.

Threshold-based detection method (TD)

When the condition that only one gesture is significantly more similar than the other gestures is satisfied, the most similar gesture is output. Gestures are not detected unless this condition is satisfied.

Classification-based detection method (CD)

Premeasured nongesture data is added to the baseline method, and the most similar gesture from among the data is output. If no gesture is output, it means that no gesture has been detected.

5.1 TD method

In the TD method, gestures are detected by applying a threshold value in the comparison of similarity. When the distance from the premeasured data B_I, which has the smallest distance from the input data A, is significantly smaller than the distance from the premeasured data B_{II}, which has the second shortest distance, then TD detects the gesture and outputs B_I. If the distances $D(A, B_I)$ and $D(A, B_{II})$ are close, then no gesture is detected. The TD method for gesture recognition is as follows, where θ is the threshold for gesture detection.

$$\text{If } \theta D\left(A, B_I\right) < D\left(A, B_{II}\right) \text{ OUTPUT: } B_I$$
$$\text{s.t. } D\left(A, B_I\right) = \min\left\{D\left(A, B_i\right)\right\}\left(i = 1, 2, 3, \ldots, N\right) \tag{2}$$

Smaller values of θ make gestures easier to detect and reduce false negatives, i.e., no gesture is detected even though a gesture is input. However, false positives increase for nongesture input, because even though there is no clear difference in similarity, the output is forced. On the other hand, larger values of θ make gestures harder to detect and reduce false positives. However, false negatives increase. Since it is difficult to detect gestures, the user may have to repeat the gesture many times.

5.2 CD method

In the CD method, walking noise data B_z is added to the premeasured data. The CD method also classifies gestures from both premeasured gestures and walking noise data. The difference from the DTW-based method mentioned in Sect. 4 is the addition of walking noise data to the premeasured data. When the input data is classified into the premeasured gesture data, the gesture is detected and output at the same time. When the distance from the walking noise data is shorter than the distance from all other premeasured data, no gesture is detected. The CD method for gesture recognition is as follows.

$$\begin{gathered}\text{If } B_I \neq B_z \text{ OUTPUT: } B_I \\ \text{s.t. } D\left(A, B_I\right) = \min\left\{D\left(A, B_i\right)\right\}\left(i = 1, 2, 3, \ldots, N, z\right)\end{gathered} \tag{3}$$

5.3 Proposed TCD method

We propose a gesture recognition method that combines the TD and CD methods, called the threshold and classification-based detection method (TCD). The walking noise data B_z is added to the premeasured data in advance. First, the distances between the input data A and all of the premeasured data B_i (i = 1, 2, 3, ..., N, z) are calculated. Here, B_I and B_{II} represent the gestures in the premeasured data with the shortest and second shortest distances from A, respectively. If B_I = B_z, then no gesture is detected. Otherwise, B_I is output only if the distance $D(A, B_I)$ is significantly shorter than the distance $D(A, B_{II})$. The TCD method for gesture recognition is as follows.

$$\begin{gathered}\text{If } \theta D\left(A, B_I\right) < D\left(A, B_{II}\right) \;\&\&\; B_I \neq B_z \text{ OUTPUT: } B_I \\ \text{s.t. } D\left(A, B_I\right) = \min\left\{D\left(A, B_i\right)\right\}\left(i = 1, 2, 3, \ldots, N, z\right)\end{gathered} \tag{4}$$

Comparison of proposed method and TD

Consider the case where the characteristics of walking noise resemble one of the premeasured gesture data. The TD method simply outputs the most similar gesture. On the other hand, the proposed TCD method correctly classifies it as walking noise data due to the addition of walking noise data to the premeasured data.

Comparison of proposed method and CD

In CD, walking noise data is also prepared as premeasured data. However, since there are various patterns of walking noise, the input walking noise data may be completely different from the premeasured walking noise. In this case, CD incorrectly outputs the closest gesture. On the other hand, even if walking noise that is completely different from the premeasured data is input, the proposed method does not detect a gesture unless there is premeasured data that is clearly similar to the input data.

6. Evaluation

In our experiment, we assumed that each gesture is performed while the user is stationary, i.e., not walking. First, the appropriate values of the threshold θ used in TD and TCD were evaluated, and then they were compared with the F1-score of each comparison method.

6.1 Environment

In this experiment, we assumed that the user is using a music player outdoors. As mentioned in Sect. 3, a general music player requires five types of essential operation and three types of auxiliary operation. Here, 13 types of gesture are set as input candidates, which is a sufficient number of gestures for our assumption. Nine subjects participated in this experiment, in which an iPhone8 with a 100 Hz three-axis accelerometer was used. Each subject acquired data 15 times for each gesture. In addition, each subject also acquired walking noise data while holding the smartphone. We evaluated gesture detection and gesture classification at the same time.

We compared TD and CD with the proposed method. TD, CD, and TCD require premeasured gesture data. Among the 15 data acquired by one subject for one type of gesture, one data was used as premeasured data and the remaining 14 data were used as test data. Cross-validation was performed for all 15 premeasured data selections for each gesture. Furthermore, classification algorithms based on machine learning were also used for comparison. We used k-NN, SVM, and RF, which are typical feature-based learning models. In these methods, 14 data were used as training data and one data was used as test data. Cross-validation was used for the evaluation.

The window size of premeasured walking noise data was set to the same as the average time of the premeasured gesture data. The data could contain either a one-step wave or a two-step wave. Since the recognition accuracy was high when the data contained two steps, we adopted two-step data. The window size of the test data was set to the largest size of all the gesture data obtained from all subjects. Since we target the start of walking, the data in which the step impact starts in the middle of the data cannot be used as test data. For this reason, the test data of walking was extracted so that it did not start in the middle of the step impact. The number of walking noise data for each subject was set to 2730 samples, the same as the total number of evaluations for each gesture in the cross-validation. The test data were extracted so that there was no time overlap.

6.2 Investigation of appropriate threshold θ

In TD and TCD, the threshold θ affects the detectability of gestures. Therefore, we focus on the change in the F1-score when θ is changed. The change in the F1-score when θ is changed by increments of 0.01 is shown for TD and TCD in Figs. 3 and 4, respectively. With the change in the value of θ, false positives and false negatives, which have a trade-off relationship, increase and decrease, respectively. It is considered that θ is an appropriate value when the F1-score takes the maximum value. We confirmed that the value of θ with the largest average F1-score for each user was 1.16 and 1.02 for TD and TCD, respectively.

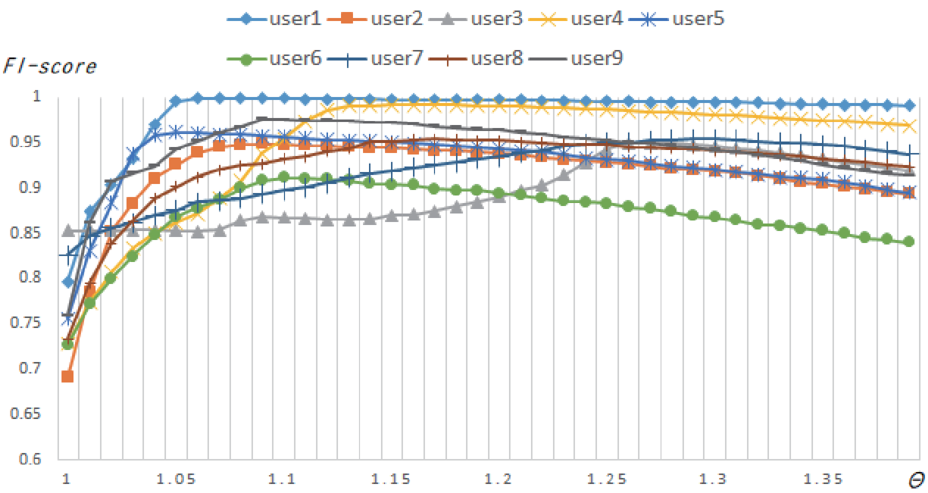


Fig. 3. (Color online) F1-score for TD.

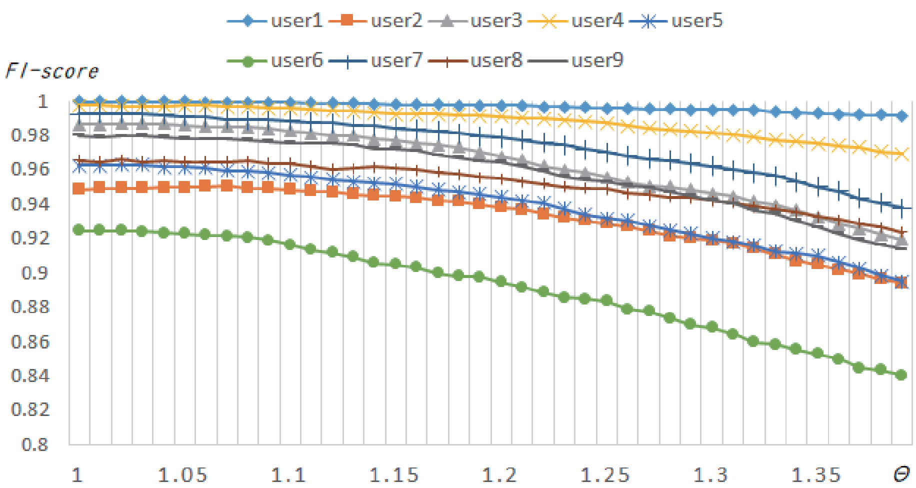


Fig. 4. (Color online) F1-score for TCD.

6.3 Performance comparison

For TD and TCD, we used the best θ values obtained as described in Sect. 6.2. A total of 18 features were used in k-NN, SVM, and RF, which are classification algorithms based on machine learning. These are the maximum, median, minimum, root mean square, mean, and variance in the x, y, and z directions. Parameters were optimally tuned by Optuna[24] for each machine learning method.

Table 2 shows the F1-score for each user for the methods of k-NN, SVM, RF, TD, CD, and TCD. We confirmed that the proposed method TCD was able to recognize gestures with higher F1-scores than TD, k-NN, SVM, and RF. TCD has better performance than CD, although the difference is small. Therefore, we investigated whether the difference is significant. We first confirmed that the hypothesis that the results of CD and TCD both follow a normal distribution could not be rejected by the Shapiro–Wilk test. Their scores were p = 0.39 for CD and p = 0.37 for TCD. As a result of a t-test based on homoscedasticity investigation, a significant difference (p = 0.019) was found in TCD compared with CD. In this experiment, we use a θ value common to all users. If the optimum θ value can be set for each user, the performance of the proposed method will be further improved.

Figure 5 shows the precision, recall, and F1-score of TCD with θ = 1.02 for each gesture, where the total result for all nine subjects is shown. The recall of the walking noise was 1. No false positive occurred in the walking noise in TCD. The highest F1-score was 99.29% for the gesture of tilt back, and the lowest F1-score was 95.19% for tilt front. We confirmed that the recall values of shake front, shake right, and shake up were markedly low, making it easy for them to be falsely detected. Since the recall of shake back was 99.95%, the "shake" gesture itself does not tend to be falsely detected.

Table 2
F1-score for each method.

	k-NN	SVM	RF	TD	CD	TCD
User1	98.30	96.60	98.98	99.79	99.97	99.97
User2	90.73	95.41	92.01	94.37	94.85	95.04
User3	94.47	93.54	93.88	87.54	98.63	98.69
User4	92.52	95.24	93.95	99.20	99.73	99.74
User5	95.99	96.94	97.01	94.82	96.24	96.31
User6	90.14	94.22	95.24	89.96	92.45	92.47
User7	97.69	97.69	98.37	92.88	99.25	99.30
User8	93.62	92.26	95.58	95.40	96.56	96.59
User9	92.60	95.94	97.28	96.71	97.98	97.98
Average	94.01	95.43	95.81	94.51	97.30	97.34

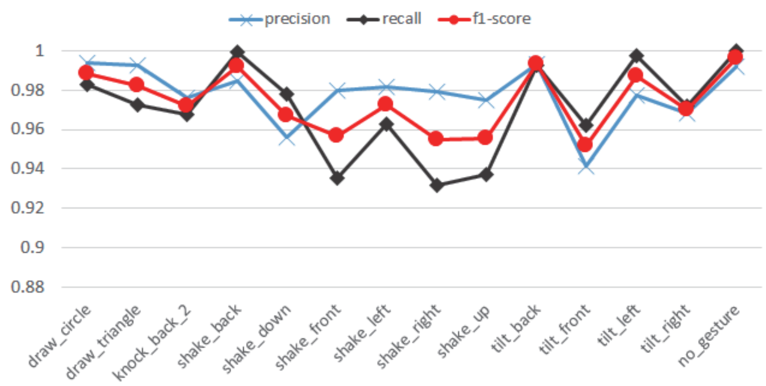


Fig. 5. (Color online) Precision, recall, and F1-score of TCD for each gesture.

7. Discussion

7.1 Consideration of experimental results and possibility of improvement

In Sect. 6.2, we described the relationship between the θ value and F1-score. If θ is small, each method forcibly detects some kind of gesture even if the data is not similar to any gesture. This is why the F1-score is extremely low when θ is small in TD, as shown in Fig. 3. Noise data that is not similar to any gesture can be stably classified as no gesture when θ is greater than 1.05. On the other hand, in TCD, walking noise is included in the premeasured gesture data; thus, walking noise can be discriminated even if θ is small.

In TD, the F1-score strongly depends on θ, whereas the dependence of the F1-score on θ is smaller in TCD. Furthermore, in TCD, the F1-score does not increase sharply only when a certain θ value is set, and it is stable and high. This shows the robustness of this method, i.e., the F1-score does not drop significantly even when θ is set to a slightly incorrect value. Thus, by conducting preliminary experiments with a sufficient number of subjects and calculating the optimum θ value in advance, a good θ value can be provided for new users. In addition, the performance of the method can be improved by introducing a mechanism that learns the user's movement and adjusts the θ value and premeasured data appropriately.

In Sect. 6.3, we described the performance of each method. The major difference between CD and TCD is whether walking noise can be recognized correctly when the walking noise is similar to other gestures. In our experiment, there were few such cases; thus, the performances of CD and TCD were similar. Depending on the type of gesture candidate, several gestures may be very similar to walking noise. In this case, the difference in the performance of CD and TCD may be large. The type of gesture that is selected as a candidate is an important issue.

As shown in Table 2, regardless of the θ value, the recognition accuracy of User 6 was clearly lower than that of the other users. Therefore, we focus on User 6. The confusion matrix for the best TCD performance of User 6 is shown in Table 3.

Table 3
Confusion matrix of User 6 in TCD.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A (shake right)	141	4	0	0	0	0	47	0	0	0	0	0	2	16
B (shake left)	6	196	0	0	0	0	0	2	0	0	0	0	0	6
C (shake up)	0	0	192	17	0	0	0	0	0	0	0	0	0	1
D (shake down)	0	0	3	206	0	1	0	0	0	0	0	0	0	0
E (shake front)	0	0	0	0	104	0	0	0	96	0	0	0	0	10
F (shake back)	0	0	0	0	0	210	0	0	0	0	0	0	0	0
G (tilt right)	0	0	0	0	0	0	210	0	0	0	0	0	0	0
H (tilt left)	0	1	0	0	0	0	0	209	0	0	0	0	0	0
I (tilt front)	0	0	0	0	7	0	0	0	200	0	0	0	0	3
J (tilt back)	0	0	0	0	0	1	0	0	0	208	0	0	0	1
K (draw circle)	0	0	0	0	0	0	0	0	0	0	210	0	0	0
L (draw triangle)	0	0	0	0	0	0	0	0	0	0	1	208	0	1
M (knock back twice)	0	0	0	0	0	0	0	0	0	0	0	0	210	0
N (no gesture)	0	0	0	0	0	0	0	0	0	0	0	0	0	2730

In Table 3, there are two noticeable misrecognitions: shake right and shake front gestures tend to be falsely recognized as tilt right and tilt front, respectively. Experimental results from other users do not show these extreme misrecognition biases. Therefore, this misrecognition can be concluded to be a characteristic of User 6. From this result, it was shown that there may be combinations of gestures that are difficult to recognize for individuals.

The confusion matrix corresponding to the sum of the results of all subjects is shown in Table 4. As a result of our effort in selecting the optimal premeasured walking data, it was possible to completely eliminate the case of walking being mistaken for another gesture. Our target gestures are actions that intentionally move a smartphone. On the other hand, walking with the device in the hand may often make users conscious of the need to keep their hand as stationary as possible. The impact of the step was measured as a major feature compared with the movement of the hand. For this reason, we consider that a clear difference was made between walking and other gestures. Table 4 also shows that no case was confirmed in which any gesture was mistaken for walking. Except for the extremely biased error of User 6, there are a relatively large number of misrecognition of shake up and shake down. Immediately before shaking up, the smartphone may have been slightly lowered to gain momentum.

7.2 Points to note and limitations of our paper

The subjects conducted all gestures and walking immediately after listening to the experimenter's explanation. The results presented in Sect. 6 used data obtained in this way. Note that we did not use data acquired in various environments, and different results may be obtained in different environments, such as on a rainy day or when holding a smartphone with the nondominant hand. By acquiring data for at least several months, it should be possible to acquire data more suitable for daily life.

Table 4
Confusion matrix of all subjects in TCD.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
A (shake right)	1759	21	13	1	0	0	57	0	0	0	3	2	2	32
B (shake left)	29	1820	0	5	0	4	0	3	0	1	1	0	9	18
C (shake up)	3	1	1776	53	0	3	0	0	8	0	4	0	12	30
D (shake down)	1	4	21	1846	1	1	0	1	0	0	0	0	5	10
E (shake front)	0	0	0	0	1768	1	0	0	104	1	0	0	2	14
F (shake back)	0	0	0	0	0	1890	0	0	0	0	0	0	0	0
G (tilt right)	0	1	0	0	0	0	1836	27	0	0	0	0	20	6
H (tilt left)	0	1	0	0	0	0	1	1886	0	0	0	0	0	2
I (tilt front)	0	0	0	0	22	0	3	17	1817	12	0	0	8	11
J (tilt back)	0	0	0	0	8	1	0	0	0	1876	0	0	0	5
K (draw circle)	1	0	9	0	0	0	0	0	0	0	1861	11	0	8
L (draw triangle)	4	11	0	14	0	0	0	0	0	0	3	1841	0	17
M (knock back twice)	0	3	3	10	6	21	0	0	2	0	1	1	1831	12
N (no gesture)	0	0	0	0	0	0	0	0	0	0	0	0	0	24570

In addition, our experiments revealed a non-negligible characteristic of User 6. It can be expected that other users will have similar characteristics. To investigate such characteristics, more experiments with a large number of subjects are required.

8. Conclusions

We proposed a gesture recognition method that improves the F1-score from that of the existing TD and CD methods. The features of the proposed method are to add walking noise into the premeasured data and set a threshold that affects the detectability of gestures. In our experiment, it was found that by adjusting the threshold, the proposed method achieved the best performance among six methods including machine-learning-based classification methods.

Our work currently targets environmental switches that involve the start of walking and does not cover other environments such as suddenly starting to run. As future work, we plan to consider many types of switch unrelated to walking.

References

1 J. Baek, J. Ik-Jin, and Y. Byoung-Ju: IEICE Trans. Inf. Syst. **89** (2006) 2957. https://doi.org/10.1093/ietisy/e89-d.12.2957
2 K. Murao, H. Yamada, T. Terada, and M. Tsukamoto: Sens. Mater. **33** (2021) 109. https://doi.org/10.18494/SAM.2021.2964
3 R. Izuta, K. Murao, T. Terada, and M. Tsukamoto: Int. J. Pervasive Comput. Commun. **11** (2015) 270. https://doi.org/10.1108/IJPCC-03-2015-0016
4 T. Marasović, V. Papić, and J. Marasović: Int. J. Adv. Rob. Syst. **12** (2015). https://doi.org/10.5772/60077
5 Y. Kanbetsunawa and R. Katsuma: IPSJ Kansai-Branch Convention 2021 (2021) (in Japanese).
6 H. T. Aung, K. Inoue, S. H. Pha, and W. Takeuchi: J. Disaster Res. **15** (2020) 353. https://doi.org/10.20965/jdr.2020.p0353
7 H. Guibing, Q. Xuesong, and L. Meng: Rtagcare: 18th Asia-Pacific Network Operations and Management Symposium (APNOMS) (2016). https://doi.org/10.1109/APNOMS.2016.7737258
8 T. Nakamura, K. Taki, H. Nomiya, K. Seki, and K. Uehara: Pattern Anal. Appl. **16** (2013) 535. https://doi.org/10.1007/s10044-011-0262-6
9 Y. Fang, D. Huy, H.-H. Huang, and K. Kawagoe: J. Comput. **10** (2015) 1. https://doi.org/10.17706/jcp.10.1.1-11
10 M. Zadghorban and M. Nahvi: Pattern Anal. Appl. **21** (2018) 323. https://doi.org/10.1007/s10044-016-0579-2
11 L. Yin, M. Dong, Y. Duan, W. Deng, K. Zhao, and J. Guo: Multimedia Tools Appl. **72** (2014) 843. https://doi.org/10.1007/s11042-013-1368-1
12 H. Sakoe and S. Chiba: IEEE Trans. Acoust. Speech, and Signal Process. **26** (1978) 43. https://doi.org/10.1109/TASSP.1978.1163055
13 T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh: Proc. 18th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD '12). (2012) 262–270. https://doi.org/10.1145/2339530.2339576
14 T. Amiya and R. Katsuma: 2021 IEEE Int. IOT, Electronics and Mechatronics Conference (IEMTRONICS) (2021) 1–7. https://doi.org/10.1109/IEMTRONICS52119.2021.9422516
15 L. Jing, Y. Zhou, Z. Cheng, and J. Wang: IEICE Trans. Inf. Syst. **94** (2011) 1062. https://doi.org/10.1587/transinf.E94.D.1062
16 Y. Xue, Y. Hu, and L. Jin: IEICE Trans. Inf. Syst. **97** (2014) 2182. https://doi.org/10.1587/transinf.E97.D.2182
17 F. Hong, S. You, M. Wei, Y. Zhang, and Z. Guo: Sensors **16** (2016) 530. https://doi.org/10.3390/s16040530
18 K. Murao and T. Terada: Int. Symp. Wearable Computers (ISWC) (2010) 1–4. https://doi.org/10.1109/ISWC.2010.5665870
19 X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang: IEEE Trans. Syst, Man Cybern. Part A Syst.Humans **41** (2011) 1064. https://doi.org/10.1109/TSMCA.2011.2116004
20 H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma: IEEE Sens. J. **16** (2016) 6425. https://doi.org/10.1109/JSEN.2016.2581023

21 Y. C. Chu, Y. J. Jhang, T. M. Tai, and W. J. Hwang: Appl. Sci. **10** (2020) 6507. https://doi.org/10.3390/app10186507
22 E. Garcia-Ceja, R. Brena, and C. E. Galván-Tejada: Springer Pattern Recognit. **8495** (2014). https://doi.org/10.1007/978-3-319-07491-7_13
23 S. Agrawal, I. Constandache, S. Gaonkar, R. R. Choudhury, K. Caves, and F. DeRuyter: ACM MobiSys'11 (2011). https://doi.org/10.1145/1999995.1999998
24 Optuna: https://optuna.readthedocs.io/en/stable/ (accessed March 2022).

About the Authors

Ryo Katsuma received his B.Ed. degree from Kyoto University of Education, Japan, in 2006 and his M.E. and Ph.D. degrees from Nara Institute of Science and Technology, Japan, in 2008 and 2011, respectively. From 2011 to 2018, he was an assistant professor at Osaka Prefecture University, Japan. Since 2018, he has been a junior associate professor at Osaka Prefecture University. His research interests are in mobile computing and sensing.
(katsuma@cs.osakafu-u.ac.jp)

Tomoaki Amiya received his B.E. degree in 2020 and his M.E. degree in 2022 from Osaka Prefecture University, Japan.