S & M 2439

Motion State Recognition Focusing on a Person's Angle of Thigh for Supporting Evacuation Route Estimation in Disaster Relief

Hiroaki Morino[*] and Chisaki Takahashi

College of Engineering, Shibaura Institute of Technology, 3-7-5 Toyosu, Koto, Tokyo 135-8578, Japan

(Received July 14, 2020; accepted September 28, 2020)

Keywords: motion state recognition, pedestrian dead reckoning, smartphone sensing, rotation vector sensor

When evacuating a building after a disaster such as an earthquake, it would be useful to detect the trajectory of an evacuation path by pedestrian dead reckoning (PDR) and record it in a person's smartphone; this could help rescue staff to trace back to where an injured person remains in the building. Considering this application, we present a novel scheme to recognize motion states of a person, including walking, descending stairs, and ascending stairs, using the person's smartphone, focusing on the angle of the thigh detected by a rotation vector sensor and acceleration values. The proposed scheme mainly solves the existing problem of height estimation using barometers, in which sensors have a time lag of their output, giving inaccurate estimation. Also, the main benefit of the proposed scheme compared with related works is that it requires only these sensor values obtained when the person is walking or running on one floor of a building as reference values, which are used to recognize other motion states, where the scheme requires training data of only a part of the target motion states for recognition. A performance evaluation with ten experimental participants shows that the proposed scheme achieves a recall rate of each motion state of over 80% and an F-value of around 0.8.

1. Introduction

One of the prominent issues in relief activities after an earthquake occurs is the rescue of injured persons in buildings. There are often damaged places in a building where people cannot pass, especially at stairs or in corridors, and the rescue staff outside the building cannot obtain sufficient information about such places in advance. Here, we consider scenarios in which evacuating persons receive location information of injured persons and record the trajectory of evacuation paths in their smartphones. This information is passed to rescue staff when they exit the building, which could help the rescue staff reach the injured persons. In this scenario, localization of their evacuation paths will be needed. The first intuitive approach for this issue is to employ existing smartphone localization functions. Moayeri *et al.* evaluated the indoor location accuracy of the existing smartphone localization framework for several movement scenarios such as walking, sidestepping, walking backward and crawling.(1) The localization

[*]Corresponding author: e-mail: morino@shibaura-it.ac.jp

https://doi.org/10.18494/SAM.2021.2977

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

application programming interface (API) used in this study was the fused location provider (FLP)[2] in Android phones and Core location[3] in iPhones. These localization schemes mainly rely on WiFi or cellular signals in the indoor environment, assisted by sensors such as magnetometers and barometers. The benefit of this approach is clearly that localization APIs are available for all devices without prior installation. However, it was shown that localization error is as large as tens of meters particularly in an area with weak WiFi signals.[1] Since cellular or WiFi reception is unlikely to be available after disasters owing to severe damage or heavily concentrated traffic in many cases, we would need indoor pedestrian dead reckoning (PDR) using smartphones without the help of wireless signals.

PDR in indoor environments estimates the moving direction, displacement, and height from an initial position using multiple sensors of a smartphone.[4–7] Movement across different floors is typically detected from a change in average atmospheric pressure. However, barometers embedded in smartphones are known to have a time lag on the order of seconds when the height changes, and the time lag varies among smartphones. Thus, in some cases where there are paths of multiple persons moving to different floors and the path each person used is to be identified, this time lag could lead to estimation error.

To solve this problem, motion recognition on the basis of steps will be effective. Motion recognition presented in the literature basically relies on a recognition model constructed by using training data for all target motion states in advance.[8–11] The drawback of this is the cost and time needed to collect training data.

We present a novel scheme for recognizing motion states of a person, including walking, walking or running downstairs, and going upstairs, by focusing on the angle of the thigh during movement. The benefit of this scheme is that it employs training data only for horizontal movement on one floor of a building and not for movement on stairs. It exploits the relative difference in sensor values compared with those obtained in movement on one floor to recognize the motion state.

2. Background

2.1 Scenario and problem statement

Figure 1 illustrates an overview of the scenario we assume in this work. An evacuating person receives attribute information of injured persons he or she encounters during evacuation through mobile peer-to-peer (P2P) communication while the location information and evacuation trajectory are recorded on his or her smartphone. These data are passed to rescue staff outside the building. After that, motion recognition is assumed to be performed offline.

In general, indoor PDR employs an acceleration sensor and gyro sensor for movement on one floor of a building, while using a barometer to estimate movement on stairs. It is often supported by map matching to reduce the accumulated estimation error. However, when tracking movement on stairs, some of the barometers embedded in smartphones have a time lag before outputting a stable value when the height changes, which makes accurate recognition difficult. Figure 2 shows an example of a time sequence of barometer values obtained in our

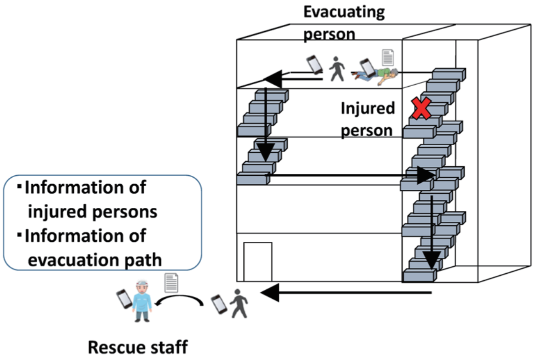

Evacuating person
Injured person
·Information of injured persons
·Information of evacuation path
Rescue staff

Fig. 1. (Color online) Scenario of evacuation trajectory recording in a building.

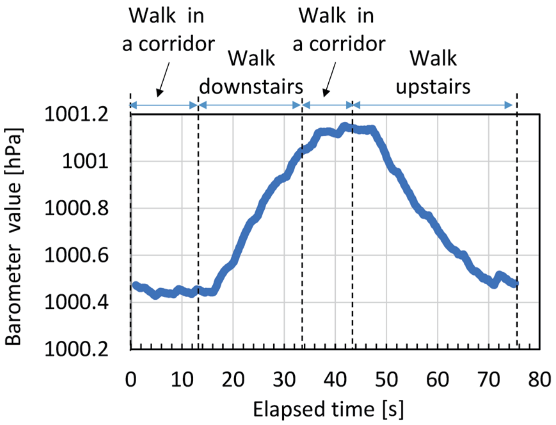


Fig. 2. (Color online) Example of barometer output time sequence when a person walks downstairs and upstairs.

preliminary experiment. The device model used in this experiment is Nexus 6, which was released in 2014. Let us focus on three timings: when the person starts going downstairs, finishes going downstairs and starts walking along a corridor, and starts going upstairs. The sensor time lags at each timing are approximately 2, 3, and 4 s, respectively. Furthermore, we confirmed that there is individual variability, for example, another device of the same model has no time lag. The same experiment was also performed using four different devices models, iPhone6s, iPhone7, Galaxy S9, and Pixel 3a XL. Maximum time lag values of them were approximately 3, 2, 0, and 0 s, respectively. These models were released in 2015, 2016, 2018, and 2019 respectively. From these results, as an overall trend, it can be said that longer time lag is seen in older models. Consequently, estimating whether the person is on the stairs from only an instantaneous barometer value inevitably will yield inaccurate results during time lag.

For example, let us consider the scenario shown in Fig. 3, where a person has to leave the emergency stairs owing to damage of the stairs. Then, possibly, he or she stays there for several seconds and moves to another set of stairs on the same floor by running and restarts going downstairs. There are three candidates for the next set of stairs: an escalator that stops in an emergency, and normal stairs A and B. This floor layout exists in the building of Toyosu campus, Shibaura Institute of Technology. We denote the shortest paths on the floor for moving from the emergency stairs to each of the candidates by paths (a), (b), and (c). As mentioned in Introduction, our system is aimed at assisting the estimation of the trajectory of evacuation paths, and in this case, it is fundamentally important to estimate to which set of stairs the evacuator moved. For this objective, it would be effective to identify which path a person took using the length of time for which the person continuously ran on this floor as one of the key values. The length of each path is about 21, 27, and 36 m. Assuming that a person runs along either of these paths, we can roughly estimate the required time length. When the height of the person is 170 cm, the stride of a single step is estimated to be about 0.6 times the person's height, as is widely known, and the time required to take two steps is set to around 0.6 s on the basis of our preliminary experiment, the velocity of running is calculated to be 3.4 m/s. Then

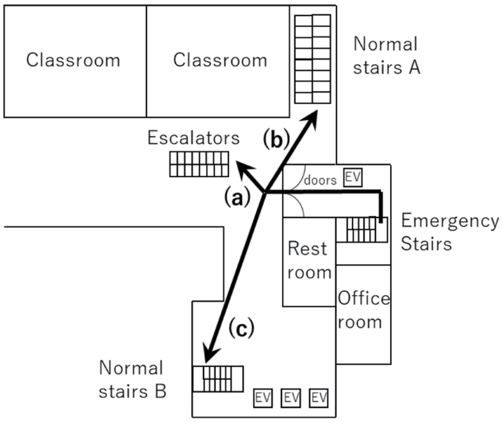


Fig. 3. Example of movement scenario in which barometer time lag affects recognition accuracy.

the time required to travel each path is about 6.1, 8.0, and 10.7 s, respectively. If the barometer time lag is negligible, we can derive these values of time from time periods for which the barometer value is continuously constant. However, considering a case where a person having a device with a barometer time lag of 4 s goes along path (a) and another case where he or she has a device with no barometer time lag goes along path (c), the time period for which the barometer value is continuously constant will be around 10 s in both cases, making it difficult to distinguish between them. This problem must be avoided.

Clearly there is large individual variability of the time lag among devices of the same model, and it is not known in advance. It is not realistic to measure the time lag of all barometers in smartphones in advance because of the individual variability. Thus, correctly identifying the motion state of each step requires not only instantaneous barometer values but also those of other sensors.

Among the five smartphone models used in our preliminary experiment of the barometer as described above, the model for which this time lag problem will arise the most is Nexus 6, whose sensor has a maximum time lag of 4 s. iPhone6s and iPhone7 also have time lags that, although smaller than the time lag of Nexus 6, will degrade the route estimation accuracy to a certain extent. Overall, the problem described here arises more often in old models than in recent models. Since many people continue to use smartphones of older models in their daily lives because they are familiar with their usability, it is important to cope with the aforementioned problem caused by barometer performance.

2.2 Related works

Indoor 3D PDR, which is particularly suited to estimating the period of time in which a person moves with a change in height, has been widely studied.(8–13) The schemes presented in Refs. 8 and 9 are used to detect whether a person moves on the same floor or moves to different floors from the change in the instantaneous atmospheric pressure, although they do not deal with the delay of the barometer output when the height changes. Kourogi *et al.*, presented a scheme for recognizing whether a person is walking, going upstairs or downstairs, or using elevators from the acceleration sensor value using a Kalman filter framework,(10) and it achieved a recognition accuracy of over 90% in experiments. It does not show the number of subjects in the experiment, nor describe how to cope with the difference in movement behavior among different persons in various motion states and therefore the accuracy is degraded. Yi *et al.* presented a scheme for recognizing the motion mode of a person, whether the person is walking, going upstairs or downstairs, or using elevators, from the acceleration sensor value obtained using a support vector machine (SVM) or a hidden Markov model (HMM).(11) Schemes were presented for recognizing the motion mode of a person as well as the state of the device, such as whether it is being held, used for calling, or kept in a pocket, using an SVM and k Nearest Neighors (kNN) in Ref. 12 and using only an SVM in Ref. 13. These schemes have the common feature that they use sensors other than a barometer to detect moving to different floors. However, the use of the SVM inevitably requires a training dataset for each motion mode (and the state of the device in Refs. 12 and 13) in advance. Furthermore, the motion state of running downstairs was not considered. This motion state often arises in evacuations and is an important target of recognition.

3. Principle of Motion State Recognition from Angle of Thigh

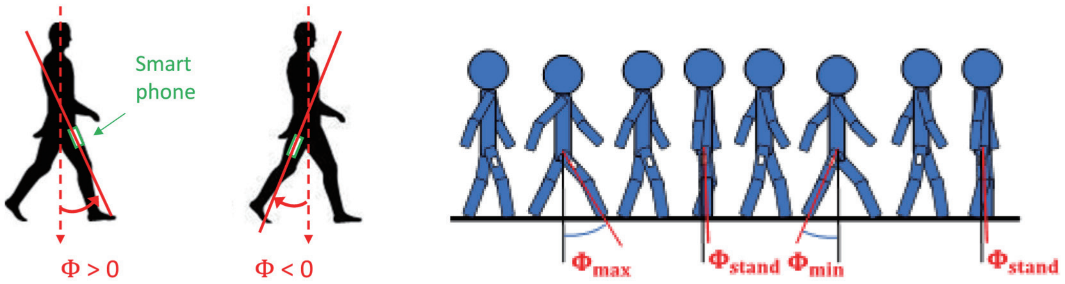
We present a scheme for recognizing the motion state of a person. This scheme can cope with the time lag of the barometer and only requires training data of part of the target motion states in advance. The target motion states for recognition are walking, running, walking downstairs, running downstairs, and walking upstairs. In this section, we briefly outline the principle of the scheme.

3.1 Characteristics of angle of thigh when person is in motion

When a person moves, both legs swing with behavior that varies depending on the motion state. In this section, we assume a person puts a smartphone in a front pocket of the pants with the screen vertical when standing straight, and we consider how the angle of the thigh on which the smartphone is resting varies during the period of a motion state.

The angle of the thigh Φ is defined as zero when a person stands with the knees straight and increases in the counterclockwise direction. Thus, Φ is positive when the leg is lifted or put forward while it is negative when the leg is behind the upper body, as shown in Fig. 4(a).

We focus on the values of Φ in the three leg states during the period of a motion state, as shown in Fig. 4(b). Φ_{max} is the angle when the person takes a step forward on the leg with the smartphone. Φ_{stand} is the angle when the leg is temporarily straight in the vertical direction during the motion state. Φ_{min} is the angle when the person takes a step forward with the leg that does not hold the smartphone. In addition, we define two values, $\Phi_{var\text{-}t} = \Phi_{max} - \Phi_{min}$ and $\Phi_{var\text{-}b} = \Phi_{stand} - \Phi_{min}$, and we examine how they depend on the motion state.



(a) (b)

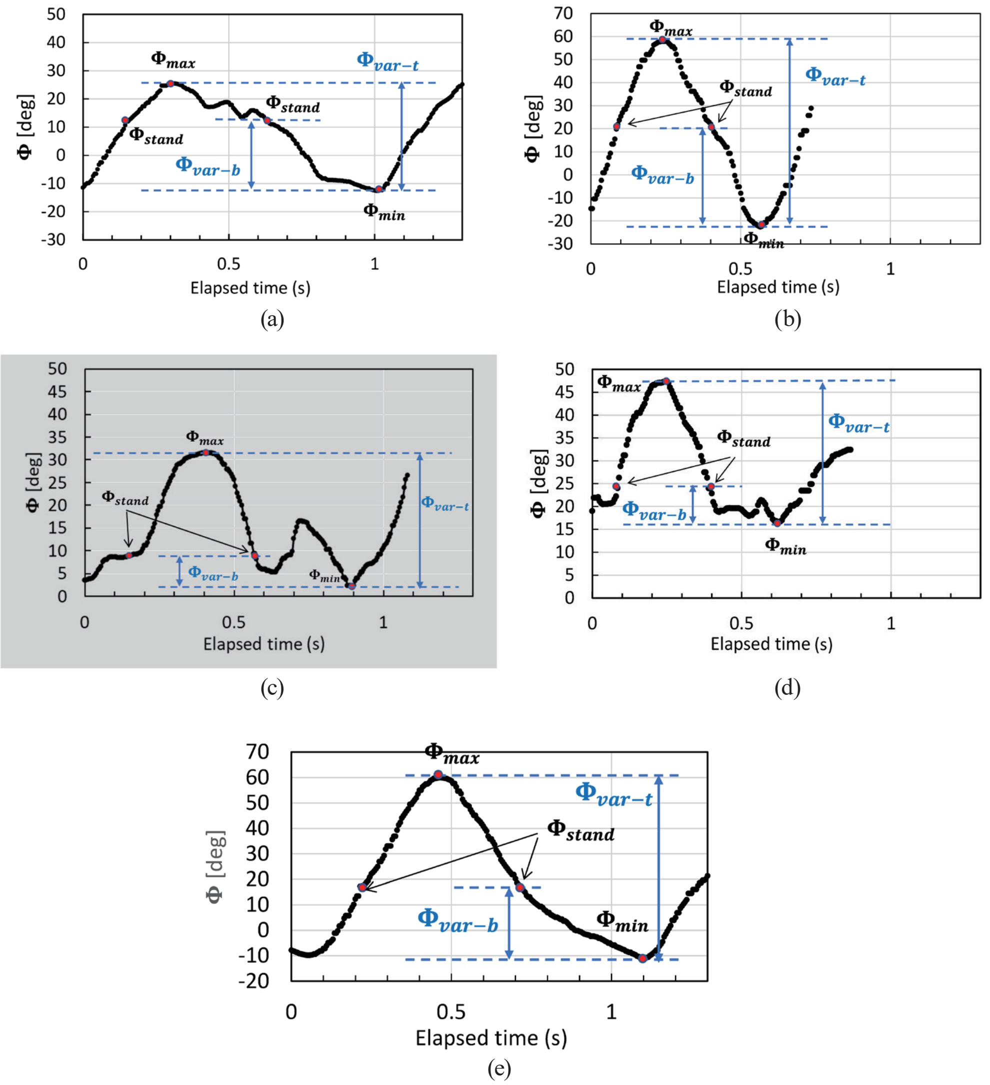
Fig. 4. (Color online) Definition of angle of thigh Φ and notations of some values during a motion state comprising two steps.

Figures 5(a)–5(e) show the time sequence of Φ during two steps for five motion states of the same person, obtained in our preliminary experiment. These results in Fig. 6 depict an approximate comparison of $\Phi_{var\text{-}t}$, $\Phi_{var\text{-}b}$, and Φ_{min} for each motion state in descending order. In general, the value of $\Phi_{var\text{-}t}$ roughly corresponds to the stride length or the degree of thigh lift when the person is ascending or descending stairs. When running, a person tends to take longer strides than in other motion states. Also, the person lifts the thighs higher when walking upstairs than in other states, leading to larger $\Phi_{var\text{-}t}$ values. In contrast, when going downstairs, a person generally takes shorter steps and the thigh lift is smaller, thus $\Phi_{var\text{-}t}$ is smaller than that in other states.

$\Phi_{var\text{-}b}$ corresponds to the degree by which the thigh is behind the upper body during steps. It is larger when the person moves on one floor than when moving on stairs, and it becomes larger when the stride is longer. Therefore, $\Phi_{var\text{-}b}$ tends to be largest when running. When descending stairs and the leg on which the smartphone is resting lags behind the upper body, a person bends the knee of this leg to follow the other leg, and $\Phi_{var\text{-}b}$ tends to be smaller than in other motion states. Finally, Φ_{min} has similar characteristics to $\Phi_{var\text{-}b}$. The difference is that Φ_{min} is negative and smaller (its absolute value is larger) when the leg is further behind the upper body, while $\Phi_{var\text{-}b}$ is positive and larger in accordance with its definition. Therefore, the descending order of Φ_{min} among the motion states is the opposite of that of $\Phi_{var\text{-}b}$.

On the basis of these characteristics, if $\Phi_{var\text{-}t}$, $\Phi_{var\text{-}b}$, and Φ_{min} are obtained for the walking state of a person, other motion states are likely to be recognized by comparing the values of these parameters with those of the walking state.

Besides these five states, the state of running upstairs would also arise in some evacuation scenarios. We decided not to include this state as a recognition target, because of the finding in our experiments that the time sequence of Φ during two steps in this state has similar features to those of walking or running and key feature values are not available at present for this state. Investigating key sensor values corresponding to this state is one aim of our future work.



(a)

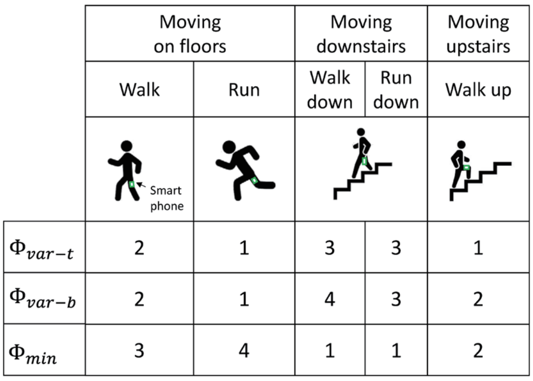
(b)

(c)

(d)

(e)

Fig. 5. (Color online) Example of time sequence of Φ during two steps in each motion state. (a) Walk. (b) Run. (c) Walk downstairs. (d) Run downstairs. (e) Walk upstairs.



	Moving on floors		Moving downstairs		Moving upstairs
	Walk	Run	Walk down	Run down	Walk up
	Smart phone				
Φ_{var-t}	2	1	3	3	1
Φ_{var-b}	2	1	4	3	2
Φ_{min}	3	4	1	1	2

Fig. 6. (Color online) Approximate order of range of Φ in each motion state in descending order of values.

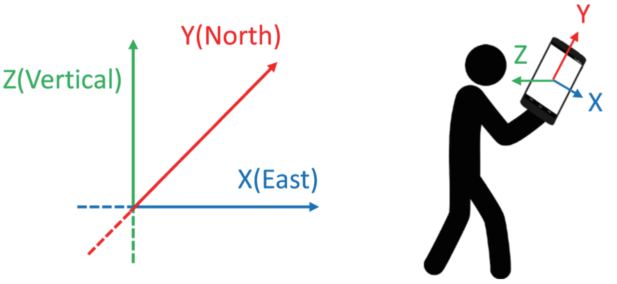
3.2 Characteristics of vertical acceleration value and time length of steps in each motion state

The characteristics of the vertical acceleration value and the time length of two steps in each motion state can also be exploited for motion state recognition. In general, vertical acceleration values are larger when running than when walking, or when running downstairs than when walking downstairs. Also, the time length of two steps tends to be shorter in the running state than in the walking state, and shorter when running downstairs than when walking downstairs. These values should be employed as feature values for recognition.

3.3 Estimation of angle of thigh using rotation vector sensor

A change in the angle of the thigh on which the smartphone is resting directly changes the pose of the smartphone. We employ the rotation vector sensor, which has been adopted in recent Android smartphones, to obtain this angle. It is a virtual sensor comprising gyro and acceleration sensors and outputs a value representing the rotation required from the predefined initial pose to attain the current pose in the world coordinate system. As depicted in Fig. 7(a), *X*, *Y*, and *Z* axes of the world coordinate system always face toward the east, the north and the vertical direction respectively. This is different from the device coordinate system used by most existing smartphone inertia sensors, in which *X*, *Y*, and *Z* axes are defined on the screen of the device, as illustrated in Fig. 7(b). Consequently, directions of axes in the world coordinate system do not change regardless of the device pose, whereas those in the device coordinate system change depending on the device pose.

The initial device pose in the rotation vector sensor is defined so that *X*, *Y*, and *Z* axes in the device coordinate system face towards the same directions as the *X*, *Y*, and *Z* axes in the world coordinate system [Fig. 8(a)]. Then, when the pose of the device changes, the sensor outputs rotation axis $\vec{v}$ and rotation angle r. This means the current device pose is obtained by a one-time rotation from the initial pose around rotation axis $\vec{v}$ by rotation angle r in the world coordinate system, as shown in Fig. 8(a). The sensor value is actually expressed by a four-dimensional number called a quaternion q_p, which consists of a scalar and a three-dimensional vector.



(a) (b)

Fig. 7. (Color online) (a) World coordinate system. (b) Device coordinate system.

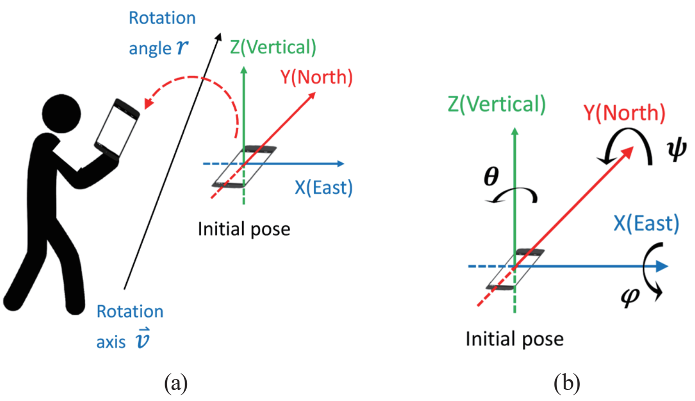


Fig. 8. (Color online) Characteristics of a rotation vector sensor. (a) Initial pose and output value of sensor. (b) Notations of rotation angles around *X*, *Y*, and *Z* axes.

$$q_p = \begin{pmatrix} \cos\frac{r}{2} \\ \sin\frac{r}{2}\cdot\vec{v} \end{pmatrix} \tag{1}$$

Furthermore, it is known that q_p can be decomposed as a product of three quaternions that correspond to rotation around the *X* axis, *Y* axis, and *Z* axis in the world coordinate system. Let us consider the case where a device changes its pose from “a” to “b” and then to “c”. Also, we look at the change from “a” to “c” directly. When we name the quaternions representing these three changes as q_{ab}, q_{ab}, and q_{ac}, respectively, the formula $q_{ac} = q_{bc} \cdot q_{ab}$ holds, where the operator “ · ” stands for the Hamilton product. When the device is stored in the front pocket so that the screen becomes vertical when standing straight, a rotation from the initial pose to the current pose represented by q_p is equivalent to performing the three rotations around the *X*, *Y*, or *Z* axis in the world coordinate system in sequence. Here, the angles of rotation performed around each axis is named φ, ψ, and θ, as illustrated in Fig. 8(b) including the direction of rotation. There are two types of rotation sequence.

Type 1 rotation sequence: In this sequence, a device is first rotated from the initial pose by angle φ around the *X* axis, then rotated by angle ψ around the *Y* axis, and finally rotated by angle θ around the *Z* axis. Thus, q_p can be written as follows.

$$q_p = \begin{pmatrix} \cos\frac{\theta}{2} \\ \sin\frac{\theta}{2}\cdot\begin{pmatrix}0\\0\\1\end{pmatrix} \end{pmatrix} \begin{pmatrix} \cos\frac{\psi}{2} \\ \sin\frac{\psi}{2}\cdot\begin{pmatrix}0\\1\\0\end{pmatrix} \end{pmatrix} \begin{pmatrix} \cos\frac{\varphi}{2} \\ \sin\frac{\varphi}{2}\cdot\begin{pmatrix}1\\0\\0\end{pmatrix} \end{pmatrix} \tag{2}$$

Type 2 rotation sequence: In this sequence, a device is first rotated from the initial pose by angle ψ around the *Y* axis, then by angle φ around the *X* axis, and finally by angle θ around the *Z* axis. Thus, q_p can be written as follows.

$$q_p = \begin{pmatrix} \cos\frac{\theta}{2} \\ \sin\frac{\theta}{2}\cdot\begin{pmatrix}0\\0\\1\end{pmatrix} \end{pmatrix} \begin{pmatrix} \cos\frac{\varphi}{2} \\ \sin\frac{\varphi}{2}\cdot\begin{pmatrix}1\\0\\0\end{pmatrix} \end{pmatrix} \begin{pmatrix} \cos\frac{\psi}{2} \\ \sin\frac{\psi}{2}\cdot\begin{pmatrix}0\\1\\0\end{pmatrix} \end{pmatrix} \tag{3}$$

In both types of rotations, values of φ, ψ, and θ are not known in advance and must be derived from q_p by solving the equations.

The type of sequence that should be applied depends on the device pose in the front pocket when standing straight. Type 1 is applied in the case that the *Y* axis of the device coordinate system (simply referred to as the *Y* axis hereafter in this section) faces vertically upward and the screen faces backward (toward the opposite of the moving direction), or the case that the *Y* axis faces vertically downward and the screen faces forward. Likewise, type 2 is applied in the case that the *Y* axis faces vertically upward and the screen faces forward, or the case that *Y* axis faces vertically downward and the screen faces backward. It is not possible to know which type should be applied in advance since the current device pose is unknown. However, when the current pose is given, only one of Eq. (2) or (3) is solved or both equations give the same solution. Thus, the scheme attempts to solve both Eqs. (2) and (3), and adopts the result of the equation that is solved.

When a person lifts or lowers his/her legs while moving, the smartphone rotates around an axis parallel to the horizontal plane. The angle of the thigh Φ is expressed using φ as follows.

1) When the device is stored so that the *Y* axis faces vertically upward
   $\Phi = \varphi - 90\,[\mathrm{deg}]$.
2) When the device is stored so that the *Y* axis faces vertically downward
   $\Phi = \varphi + 90\,[\mathrm{deg}]$.

Since it is not possible to know in advance the direction of the *Y* axis when standing straight, motion recognition is performed by using φ instead of by calculating Φ.

θ corresponds to the angle of a person's turn around the *Z* axis, i.e., the vertical axis. ψ corresponds to the angle of rotation of the device around the *Y* axis or *Z* axis in the device coordinate system. In this paper, we assume the variance of ψ during a person's movement is negligible in accordance with the results of our preliminary experiments and that ψ is constant.

Currently, this approach has some limitations in terms of the range of applicable cases. There can be cases where the device is not stored in the clothes, i.e., a person holds the device or it is stored in a bag. In these cases, neither Eq. (2) nor (3) can be solved. Also, when the device is stored in places other than a front pocket of the pants, such as a back pocket of the pants or a shirt pocket, this scheme cannot detect the movement of the thigh. These cases are currently beyond the scope of this paper, and we aim to extend the range of applicable cases in the future.

Another issue is the effect of sensor errors on performance. When a rotation vector sensor value contains error either in the rotation angle or the rotation axis or both, it could affect recognition accuracies since the performance of the proposed scheme relies on the accuracies of φ, ψ, and θ obtained by Eqs. (2) and (3). Furthermore, in the case that errors are large, it is possible that the quaternion representing a sensor value cannot be decomposed using the equations, which means a failure of pose estimation. To cope with this problem, we must analyze the distribution of errors included in φ, ψ, and θ given the largest amount of error in the sensor value, and then calculate the probability distribution of recognition accuracy with the given distribution of errors. Such studies are the topics of our future work.

4. Proposed Method

Using the principles mentioned in Sect. 3, we present a scheme for detecting each motion state from the time sequence of the thigh angle and the acceleration value assuming a person puts the smartphone in a front pocket of the pants.

4.1 Overview of detection sequence

This scheme employs three types of sensors for recognition: a rotation vector sensor, an acceleration sensor, and a barometer. Motion recognition consists of the following sequence.

Procedure 1) Step detection using acceleration sensor
Procedure 2) Segmentation of rotation vector sensor data every two steps
Procedure 3) Recognition of walking and running on one floor
Procedure 4) Recognition of motion states for all segmented data

4.2 Step detection and segmenting sequence of rotation vector sensor values

For a sequence of rotation vector sensor data obtained over the entire evacuation time period, a time sequence of φ is obtained using Eqs. (2) or (3). Then, it is segmented every two steps by using step detection, whose time length is called a motion in this paper,

Step detection only using a smartphone has been extensively studied,(14–17) and previous methods can basically be applied in the proposed scheme. These methods have additional capabilities besides simply detecting step counts in normal movements; detecting steps when walking while swinging the hand holding the smartphone,(14,17) detecting both step length and step counts,(15,17) coping with falsely detected walking states where the user rocks the smartphone while stopping.(16) Consequently, these algorithms are too complex to be used for our purpose because, for example, motion state recognition is incorporated within them, or they require both acceleration sensor and gyrosensor values, or the algorithms are based on machine learning. Since we mainly aim to evaluate the principles of the proposed scheme presented in Sect. 3, a simple algorithm is sufficient and has benefits in terms of implementation. As another approach, we tested a step counter embedded in a smartphone. However, its accuracy was low in the model used in our experiment (Nexus 6) and was not acceptable. For these reasons, we adopted a simple step count algorithm based on the detection of the peak acceleration sensor value for reducing implementation complexity.

A smartphone acceleration sensor generally outputs sensor data represented in the device coordinate system. First, it is transformed into data represented in the world coordinate system to enable analysis of the data independent of the device pose. We denote acceleration sensor data represented in the device coordinates as $\overrightarrow{a_{dev}} = \left(x_{dev}, y_{dev}, z_{dev}\right)^t$ and denote transformed data represented in the world coordinates as $\overrightarrow{a_{enu}} = \left(x_{enu}, y_{enu}, z_{enu}\right)^t$. We focus on $z_{enu}' = z_{enu} - g$, the acceleration along the vertical axis minus gravitational acceleration, and perform the following step detection using this value.

1) Configure a threshold A_{step} and detect the time at which z_{enu}' becomes larger than A_{step}. Tentatively set the time as a step timing.
2) In a period of time later than the time detected in 1), find a sequence of data that begins at the time when z_{enu}' becomes smaller than A_{step} and ends at the time when z_{enu}' again becomes larger than A_{step}. The end time is set so that it is later by no less than the predefined time of t_s from the time z_{enu}' becomes smaller than A_{step}. Set this end time as the tentative timing of the next step.
3) Between the two times detected in 1) and 2), find the time at which z_{enu}' reaches a maximum and set it as the exact timing of a step.

Figure 9 depicts an example of step detection and sequence segmentation in the motion state of walking. The orange line shows a sequence of z_{enu}', vertical acceleration minus gravitational acceleration, mentioned in Sect. 4.2. The time $T(k)$ (k = 1, 2, ...), shown by the green broken lines, indicates the timing of the kth step detected by the above algorithm with A_{step} = 4.0 m/s^2 and t_s = 0.25 s. The time period between $T(k)$ and $T(k + 2)$ (k = 1, 3, 5 ...) is defined as the ith motion, where $i = [k/2] + 1$. In the time period of each motion, the maximum value of z_{enu}' denoted by z_{max}, the time length of the period denoted by t_{motion}, φ_{max}, and φ_{min} are detected.

Next, the scheme detects φ_{stand}, the value of φ at the moment the person is standing straight, in each motion. Although φ_{stand} should be either 90 or −90 deg depending on the device orientation in the pants, it often takes values that are largely different from these because there are cases where the legs of a person in motion do not become vertical, particularly on stairs. Subsequently, the scheme detects φ_{stand} by the following procedure.

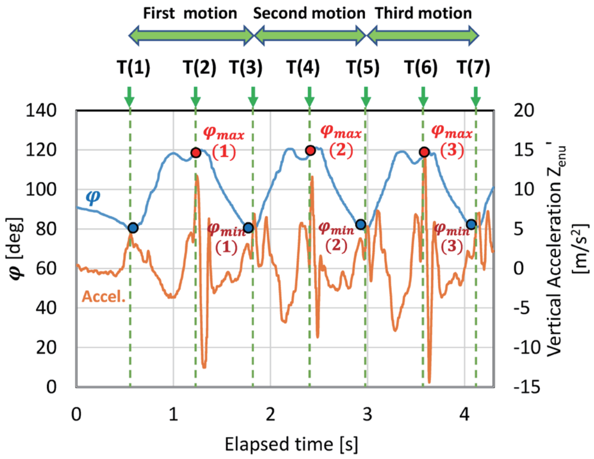


Fig. 9. (Color online) Example of sequence segmentation of φ.

1) Set a tentative value of φ_{step}, denoted by φ_{tent}. Find time t in the φ sequence of the motion such that $\varphi(t) = \varphi_{tent}$. Here, φ_{tent} is chosen so that there are no less than three different values of t satisfying this condition, and we represent these three values as t_1, t_2, and t_3 in ascending order.
2) Find φ_{step} by the following formula:

$$\varphi_{stand} = \underset{\varphi_{tent}}{\arg\min} \left\{ abs\left[\left(t_3 - t_2\right) - \left(t_2 - t_1\right) \right] \right\}. \tag{4}$$

Then, $\Phi_{var\text{-}t}$ and $\Phi_{var\text{-}b}$ are calculated as $\Phi_{var\text{-}t} = \varphi_{max} - \varphi_{min}$ and $\Phi_{var\text{-}b} = \varphi_{stand} - \varphi_{min}$ in accordance with the definitions in Sect. 3.

4.3 Recognizing walking and running states in continuous horizontal movements

Before detecting motion states in horizontal movements, the proposed scheme finds the time periods during which the person moves continuously on the same floor using a sequence of atmospheric pressure data. It finds a period of time during which atmospheric pressure values are stable, i.e., do not change by more than the predetermined $P_{th\text{-}fl}$ for T_{FL}, where $T_{FL} > (2 \times T_{RT})$.

T_{RT} is set as slightly larger than the maximum time lag of typical smartphone barometers. This period of time is used to judge whether the person walked on the same floor for the time duration of $T_{FL} - (2 \times T_{RT})$. The reason why 2 is multiplied with T_{RT} is to provide a margin and avoid using sensor values obtained during the time lag. Since the raw atmospheric sensor data include small spontaneous fluctuations even when moving on the same floor, we first apply a moving-average filter of a 2 s window to the raw data before applying this detection method.

Next, for each motion (time period of two steps), the scheme detects whether the person is walking or running on the basis of $\Phi_{var\text{-}t}$ and t_{motion} compared with predetermined thresholds $\Phi_{vthresh}$ and $t_{mthresh}$. To define $\Phi_{vthresh}$ and $t_{mthresh}$, the scheme requires datasets of $\Phi_{var\text{-}t}$ and t_{motion} for walking and running on the same floor collected in advance by a small number of people. In this study, we collected such data from four persons for this objective. Using the datasets, the scheme determines that the person is walking if $\Phi_{var\text{-}t} < \Phi_{vthresh}$ or $t_{motion} > t_{mthresh}$; otherwise the person is running.

After recognizing the motion states on the same floor, the scheme obtains a distribution of the following data for the time periods during which the person is recognized to be walking: $\Phi_{var\text{-}t}$, $\Phi_{var\text{-}b}$, Φ_{min}, t_{motion}, and z_{max}. They are used as reference values to detect the other motion states, as described in the next subsection. These distributions are denoted by $\hat{\Phi}_{var-t}\,(walk)$, $\hat{\Phi}_{var-b}\,(walk)$, $\hat{\Phi}_{min}\,(walk)$, $\hat{t}_{motion}\,(walk)$, and $\hat{z}_{max}\,(walk)$, respectively.

4.4 Recognizing motion states for all motions except continuous horizontal movements

By the procedure described in Sect. 4.3, the motion states for motions other than those on the same floor are recognized in accordance with the flowchart shown in Fig. 7. Here, P90 and P10 mean the 90th percentile and the 10th percentile of each distribution, respectively. Comparisons and conditions provided in this flowchart basically follow the observations on human motions described in Sects. 3.1 and 3.2.

For this scheme, the only training data that must be prepared in advance are time sequences of φ and acceleration values for the walking and running states on the same floor. This is a major benefit of this scheme.

On the other hand, a movement that is not continuous such as a single step walk is not detected by the proposed scheme since sensor data segmentation is performed with a unit of two steps. We believe this is not a serious issue and it is not associated with the barometer time lag problem stated in Sect. 2.1.

5. Performance Evaluation

5.1 Configuration

We performed evaluation experiments in the research building and the classroom building in Toyosu Campus of Shibaura Institute of Technology. The research building has 14 stories with two emergency staircases and the classroom building has seven stories with two normal staircases. These buildings are connected on each floor. The participants of the experiment started in the corridor on the 12th floor of the research building and moved in the following sequence.

The route map of the movement is illustrated in Fig. 11. The numbers in brackets in the figure correspond to those in the following descriptions.

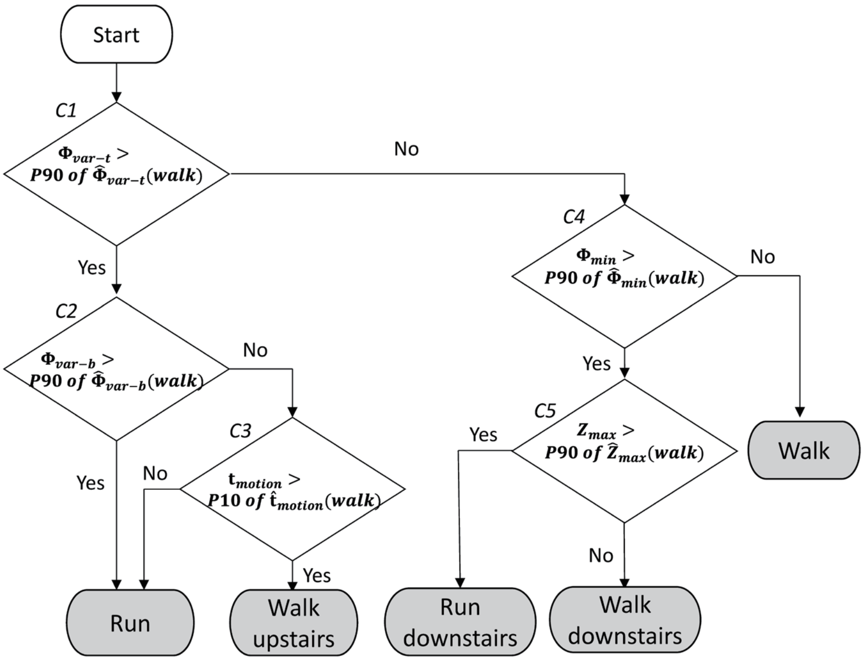

Start
C1
$\Phi_{var-t} >$ $P90\ of\ \hat{\Phi}_{var-t}(walk)$
No
Yes
C2
$\Phi_{var-b} >$ $P90\ of\ \hat{\Phi}_{var-b}(walk)$
No
Yes
C3
$t_{motion} >$ $P10\ of\ \hat{t}_{motion}(walk)$
No
Yes
C4
$\Phi_{min} >$ $P90\ of\ \hat{\Phi}_{min}(walk)$
No
Yes
C5
$Z_{max} >$ $P90\ of\ \hat{Z}_{max}(walk)$
Yes
No
Walk
Run
Walk upstairs
Run downstairs
Walk downstairs

Fig. 10. Flowchart of motion state recognition except for movements on same floor.

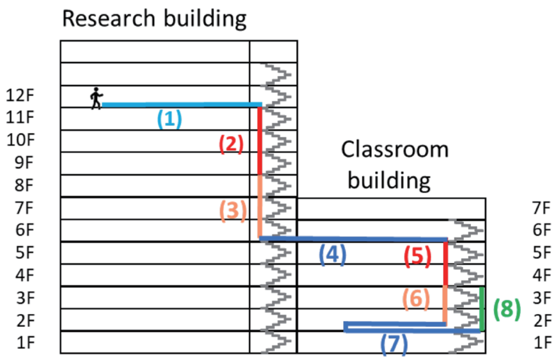


Fig. 11. (Color online) The route map of the movement in the experiments.

1) Run to the stairs on the same floor (1) and run downstairs to the 9th floor (2). Then walk downstairs to the 6th floor (3). The stairs include multiple landings.
2) Leave the stairs and walk to stairs in the classroom building on the same floor (4).
3) Run down the stairs to the 4th floor (5) and then walk down to the second floor (6). Walk on this floor for approximately 20 steps (7). Finally, walk up to the 4th floor (8).

There were 10 participants. They include four persons for which data of Φ were collected for walking and running in advance to determine $\Phi_{vthresh}$ and $t_{mthresh}$ by the process described in Sect. 4.3. A Nexus 6 smartphone was stored in the front left pocket of the pants throughout the experiment so that the Y axis in the device coordinate system faced vertically upward and the screen faced forward when standing straight.

The parameters of the scheme are configured as listed in Table 1. The sampling rate of sensors was set to 200 Hz.

5.2 Results for recognition accuracy

Table 2 shows the recognition accuracy of each motion state. The value in each cell is calculated as the number of times the motion state was correctly recognized divided by the number of true motion states indicated in the same row. Thus, the values in the cells marked in yellow are the recall rates of true motion states.

The recall rates of walk downstairs and run downstairs are below 70%. As recognition errors, walk downstairs is often recognized as run downstairs and vice versa. When we focus on solving problems caused by the barometer time lag such as described in Sect. 2.1, it would be sufficient to distinguish go downstairs from horizontal movement regardless of whether it is walk downstairs or run downstairs. In this viewpoint, it would be possible to consider the results in the light orange cells as correct recognition and the recall rates would exceed 80% for all motion states.

Table 3 shows F values of results in each motion state. Furthermore, as with Table 2, Table 4 shows F values where walk downstairs and run downstairs are considered to be the same motion state of go downstairs. In Table 4, the F value is minimum in the run state, 0.78, while it exceeds 0.8 in other motion states.

Table 1
Parameter configuration.

Symbol	Value	Meaning
A_{step}	4.0 m/s^2	Threshold of accelerator value to detect peak
t_s	0.25 s	Threshold of minimum time difference between a step and the next step
T_{RT}	4 s	Presumed maximum time lag of barometers
$P_{th\text{-}fl}$	0.03 hPa	Threshold of barometer value fluctuation to detect horizontal movement
$\Phi_{vthresh}$	50 deg	Threshold of angle of thigh difference to recognize walking and running in horizonal movement
$t_{mthresh}$	0.9 s	Threshold of time length of a motion to recognize walking and running in horizonal movement

Table 2
(Color online) Recall rate for each motion state.

			Ground truth				
			Walk	Run	Walk downstairs	Run downstairs	Walk upstairs
Estimation results		Walk	82.0 (%) (1337/1630)	6.8 (%) (37/544)	8.0 (%) (91/1140)	7.8 (%) (61/784)	3.7 (%) (43/1166)
		Run	3.9 (%) (64/1630)	89.5 (%) (487/544)	0.1 (%) (1/1140)	6.8 (%) (53/784)	9.2 (%) (107/1166)
	Go downstairs	Walk downstairs	6.1 (%) (99/1630)	0.4 (%) (2/544)	67.5 (%) (769/1140)	22.1 (%) (173/784)	1.5 (%) (18/1166)
		Run downstairs	2.0 (%) (33/1630)	1.5 (%) (8/544)	22.4 (%) (255/1140)	59.9 (%) (470/784)	0.1 (%) (1/1166)
		Walk upstairs	6.0 (%) (97/1630)	1.8 (%) (10/544)	2.1 (%) (24/1140)	3.4 (%) (27/784)	85.5 (%) (997/1166)

Table 3
F-values.

Walk	Run	Walk downstairs	Run downstairs	Walk upstairs
0.836	0.775	0.699	0.606	0.859

Table 4
F values when walk downstairs and run downstairs are considered to be the same motion state of go downstairs.

Walk	Run	Go downstairs	Walk upstairs
0.836	0.775	0.889	0.859

Figure 12 shows the range of $\Phi_{var\text{-}t}$ in each motion state for all participants as a box-whisker plot, together with $\hat{\Phi}_{var-t}$ (*walk*). The box of $\hat{\Phi}_{var-t}$ (*walk*) is lower than that for the states of running and walking upstairs, which means that it acts as a key metric for detecting these two motion states. Also, Fig. 13 shows Φ_{min} in each motion state together with $\hat{\Phi}_{min}$ (*walk*) for all participants. These results show that $\hat{\Phi}_{min}$ (*walk*) is an effective metric for detecting running downstairs and walking downstairs and distinguishes them from the walking state.

Table 5 shows the recall rate for each participant. The scheme gives a high recall rate for most of the participants, although the accuracy is very low in some motion states for participant No. 8. This is partly related to the results in Table 6, which shows the recognition accuracy of walking and running on floors obtained with procedures described in Sect. 4.3. The accuracy

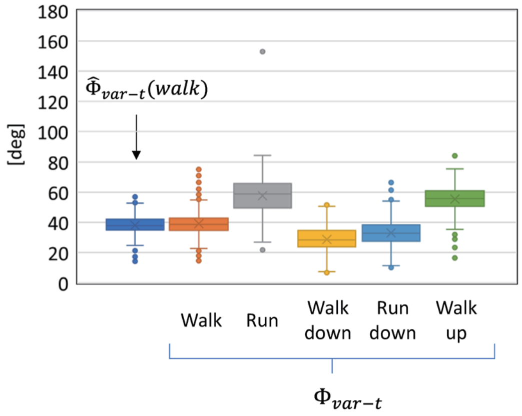


Fig. 12. (Color online) Distributions of $\Phi_{var\text{-}t}$ and $\hat{\Phi}_{var-t}$ (*walk*) for all participants.

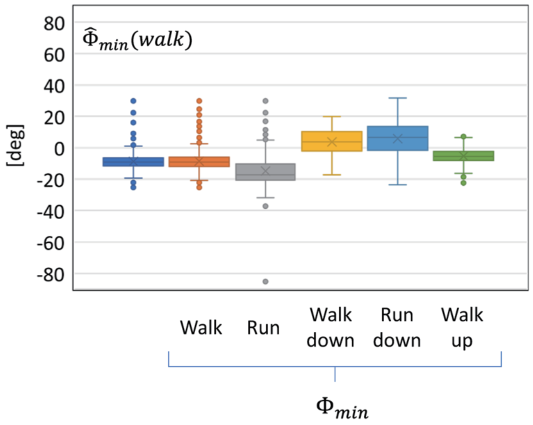


Fig. 13. (Color online) Distributions of Φ_{min} and $\hat{\Phi}_{min}$ (walk) for all participants.

Table 5
Recall rate for each participant.

		Recall rate for a participant (%)				
		Walk	Run	Walk downstairs	Run downstairs	Walk upstairs
Participant No.	1	88.5	91.2	76.7	62.0	89.2
	2	77.6	93.9	93.5	61.8	62.5
	3	80.4	89.8	92.9	52.9	82.2
	4	95.0	72.3	14.1	67.1	76.9
	5	84.3	91.5	92.1	86.2	91.2
	6	78.0	100.0	16.1	86.9	82.5
	7	80.2	82.0	61.5	89.0	93.8
	8	62.9	83.0	63.0	51.0	88.8
	9	84.6	95.4	92.7	2.7	94.2
	10	79.3	93.0	90.0	38.9	84.7

Table 6
Recall rate of walking and running states on the same floor in the recognition procedure.

Participant No.									
1	2	3	4	5	6	7	8	9	10
95% (63/66)	98% (78/80)	94% (49/52)	98% (92/94)	98% (44/45)	98% (60/61)	81% (56/69)	87% (41/47)	99% (66/67)	100% (50/50)

is relatively low for participants No. 7 and No. 8. The recognition accuracy when moving on the same floor varies among the participants, which affects $\hat{\Phi}$ (*walk*) and leads to variation of the recognition accuracy among the participants for other motion states. Thus, improving the recognition accuracy for motion on the same floor for some of the participants will be a key issue in improving the performance of the proposed scheme.

6. Discussion

In this section, we describe effect of errors. First, we consider what will happen when a person's movement does not correspond to any motion state that the proposed scheme can recognize. Among the motions seen in evacuation and evaluated in Ref. 1, let us take walking backward and sidestepping as examples. Regarding walking backward, assuming that step detection correctly detects steps, the scheme will recognize the movement as walk since the patten of leg swing in two steps resembles that of walking forward. This error will be solved by improving the recognition algorithm so that it detects whether Φ reaches Φ_{max} or Φ_{min} when vertical acceleration z_{enu}' reaches z_{max} in a period of two steps, hence distinguishing walking forward from walking backward on this basis of this result. Regarding sidestepping, the first issue is that the step detection adopted in the proposed scheme and described in Sect. 4.2 does not correspond to sidesteps. This error will be solved by introducing sidestep detection schemes reported in the past literature such as Ref. 18. The second issue is that a thigh swing that occurs when a person sidesteps causes device rotation around an axis directed from the back to the front of the body. Using Eq. (2), the angle of this rotation is obtained from a change of ψ, not φ. Thus, it is necessary to develop additional algorithms to recognize sidestepping (walk or run) using a time sequence of ψ by the analysis of movement patterns, similar to that described in Sect. 3.1.

7. Conclusions

We presented a scheme to detect five motion states of a person, including movement up and down stairs, in an indoor environment at every two steps using the angle of the thigh and the acceleration with only training data obtained from walking and running on the same floor in advance. The evaluation by experiments showed that the proposed scheme achieves an F-value of recognition of over 0.8 for motion states other than running, for which the F-value is 0.78. Owing to its satisfactory performance, we believe that the scheme will be useful for achieving our initial objective of avoiding the time lag problem in recognition using only a barometer.

For future work, we plan to investigate the relationship between data for walking and running on the same floor and those on stairs for each person, and explore the possibility of using different types of recognition algorithm in accordance with the characteristics of a person's movement to reduce the variation in recognition accuracy among persons.

References

1 N. Moayeri, C. Li, and L. Shi: Proc. IEEE Wireless Communications and Networking Conf. (WCNC), 2019. https://doi.org/10.1109/WCNC.2019.8885639

2 Google: Fused Location Provider API, https://developers.google.com/location-context/fused-location-provider (accessed 7 September 2020).

3 Apple: Core Location, https://www.apple.com/newsroom/2018/06/apple-ios-12-securely-and-automatically-shares-emergency-location-with-911/ (accessed 7 September 2020).

4 B. Shin, C. Kim, J. Kim, S. Lee, C. Kee, H. S. Km, and T. Lee: IEEE Sens. J. **16** (2016) 6977. https://doi.org/10.1109/JSEN.2016.2585655

5 K. Kaji and N. Kawaguchi: Proc. Int. Conf. Indoor Positioning and Indoor Navigation (IPIN) (2016). https://doi.org/10.1109/IPIN.2016.7743643

6 L. Zheng, W. Zhou, W. Tang, X. Zheng, A. Peng, and H. Zheng: Simul. Modell. Pract. Theory **65** (2016) 45. https://doi.org/10.1016/j.simpat.2016.01.003

7 B. Zhou, Q. Li, Q. Mao, W. Tu, and X. Zhang: IEEE Trans. Human-Mach. Syst. **45** (2015) 562. https://doi.org/10.1109/THMS.2014.2368092

8 S. Asano, Y. Wakuda, N. Koshiduka, and K. Sakamura: Proc. 2012 IEEE/ION Position, Location and Navigation Symp. (2012). https://doi.org/10.1109/PLANS.2012.6236898

9 M. Elhoushi, J. Georgy, A. Noureldin, and M. J. Korenberg: IEEE Trans. Instrum. Meas. **65** (2016) 208. https://doi.org/10.1109/TIM.2015.2477159

10 M. Kourogi and T. Kurata: Proc. IEEE and ACM Int. Symp. Mixed and Augmented Reality (ISMAR) (2003) 103. https://doi.org/10.1109/ISMAR.2003.1240693

11 S. Yi, P. Mirowski, T. K. Ho, and V. Pavlovic: Proc. 22nd Int. Conf. Pattern Recognition (2014) pp. 3505-3510, 2014. https://doi.org/10.1109/ICPR.2014.603

12 G. Wang, Q. Li, L. Wang, W. Wang, M. Wu, and T. Liu: Sensors **18** (2018). https://doi.org/10.3390/s18061965

13 B. Wang, X. Liu, B. Yu, R. Jia, and X. Gan: Sensors **18** (2018) 7577. https://doi.org/10.1109/JSEN.2018.2861395

14 M. Susi, V. Renaudin, and G. Lachapelle: Sensors **13** (2013) 1539. https://doi.org/10.3390/s130201539

15 N. Ho, P. H. Truong, and G. M. Jeong: Sensors **16** (2016). https://doi.org/10.3390/s16091423

16 F. Gu, K. Khoshelham, J. Shang, F. Yu, and Z. Wei: IEEE Sens. J. **17** (2017) 3453. https://doi.org//10.1109/JSEN.2017.2685999

17 L. Xu, Z. Xiong, J. Liu, Z. Wang, and Y. Ding: Remote Sens. **11** (2019). https://doi.org/10.3390/rs11030294

18 M. Kourogi, T. Ishikawa, and T. Kurata: Proc. IEEE/ION Position, Location and Navigation Symposium (PLANS) (2010) 85. https://doi.org/10.1109/PLANS.2010.5507239

About the Authors

Hiroaki Morino received B.E, M.E. and Ph.D. degrees from the University of Tokyo in 1994, 1996 and 1999, respectively. He is currently a professor of Shibaura Institute of Technology, Japan. His research interests include smartphone-based human motion recognition, vehicular networks and its applications, and mobile computing.

Chisaki Takahashi is an undergraduate student in the department of communications engineering, Shibaura Institute of Technology. Her research interest is smartphone-based human motion recognition.