Choi, HS, Lee, GM and Rhee, WS

A Consecutive Motion and Situation Recognition Mechanism to Detect a Vulnerable Condition Based on Android Smartphone

ISSN:1738-6764, eISSN:2093-7504

A Consecutive Motion and Situation Recognition Mechanism to Detect a Vulnerable Condition Based on Android Smartphone

Hoan-Suk Choi [1], Gyu Myoung Lee [2] and Woo-Seop Rhee [3,*]

[1] Korea Advanced Institute of Science and Technology; Contract Research Scientist; hkrock7904@gmail.com
[2] Liverpool John Moores University; Professor; g.m.lee@ljmu.ac.uk
[3] Hanbat National University; Professor; wsrhee@hanbat.ac.kr
* Correspondence

https://doi.org/10.5392/IJoC.2020.16.3.001

Manuscript Received 10 July 2020; Received 23 September 2020; Accepted 23 September 2020

Abstract: *Human motion recognition is essential for user-centric services such as surveillance-based security, elderly condition monitoring, exercise tracking, daily calories expend analysis, etc. It is typically based on the movement data analysis such as the acceleration and angular velocity of a target user. The existing motion recognition studies are only intended to measure the basic information (e.g., user's stride, number of steps, speed) or to recognize single motion (e.g., sitting, running, walking). Thus, a new mechanism is required to identify the transition of single motions for assessing a user's consecutive motion more accurately as well as recognizing the user's body and surrounding situations arising from the motion. Thus, in this paper, we collect the human movement data through Android smartphones in real time for five targeting single motions and propose a mechanism to recognize a consecutive motion including transitions among various motions and an occurred situation, with the state transition model to check if a vulnerable (life-threatening) condition, especially for the elderly, has occurred or not. Through implementation and experiments, we demonstrate that the proposed mechanism recognizes a consecutive motion and a user's situation accurately and quickly. As a result of the recognition experiment about mix sequence likened to daily motion, the proposed adoptive weighting method showed 4% (Holding time=15 sec), 88% (30 sec), 6.5% (60 sec) improvements compared to static method.*

Keywords: motion classification; situation recognition; movement sensor; vulnerable condition; android smartphones

1. Introduction

Human motion recognition is one of the most interesting research topics for user-centric services such as surveillance-based security, elderly condition monitoring, exercise tracking, counting daily calories burned, etc [1]. It is required to recognize physical status of a target user such as what he or she is doing [2].

Human motion recognition has been classified into two type of approaches. The first approach is an infrastructure-based approach which captures the human motion using the video cameras installed in the target environment. However, this approach is not suitable for the real-time mobile environment because complicated calibration procedures are usually required [3]. Another approach is a user-centric approach which attaches various type of movement sensors onto the user's body. According to a recognition purpose, the research work on motion recognition has mainly focused on classifying a certain motion (e.g., sitting, walking, running and falling, etc.) which is required to a target application. Also, another important factor for motion recognition is situational information such as how the target's moving (by car, train or on foot) [4] or a vulnerable (life-threatening) condition for the elderly.

A single (atomic) motion could be recognized using current recognition methods. However, it's a big challenge to develop a recognition method specially with smartphones considering the nature of complex human activity. A human activity may composite several atomic motions concurrently occurred or interleaved. Also, they may have different patterns and temporal dependencies [5]. Therefore, we need a way to recognize a consecutive motion which consists of several atomic motions and its transition. However, a consecutive motion is very difficult to recognize with one type of information and its performance is not good. Therefore, much research has been conducted to make a comprehensive use of relevant context and environment information as well as sensor data. For example, the sensor fusion techniques have been studied to intelligently fuse data from various sensors attached to target users or objects [6]. It utilizes a large number of homogeneous/heterogeneous sensors to recognize complex situations.

Apart from human motion recognition, a typical complex situation such as a vulnerable condition of the elderly can be derived different results by a variety of factors including the target's health profile, recent activity history, and the before and after status of an event such as a fall or an accident. In particular, the elderly may have less ability to see and a sense of balance than the young or their athletic ability may be impaired due to the existing chronic disease. Thus, the probability of falls increases if the elderly didn't find the obstructions while walking or in case of severe bumps on the walking surface. In addition, in some cases, the elderly fall by heatstroke or a chronic disease. Therefore, a fall of the elderly can often be a life-changing event, which can lead to a decline in physical function or even loss of autonomy [7]. When a fall occurs, if they are able to get up or if routine movements are possible, then, it is no big problem. However, if they lose consciousness or they are unable to move normally due to an injury caused by a fall, then, it is very dangerous [8]. In such cases, they need an active and immediate response from the peoples who are medical personnel, family and someone around them.

To recognize a vulnerable condition, we need to observe target user's status and detect an accidental event. Based on this, we should make a comprehensive decision taking human motion recognition into consideration. However, the existing motion recognition studies are only intended to measure additional information such as a stride and a step or to recognize a single motion. Therefore, a consecutive motion and situation recognition mechanism is proposed. The proposed solution was implemented as Android Smartphone App for real-time movement data collection and vulnerable condition notification. Because a smartphone is an available sensor-rich and portable mobile device. Also, it helps to provide a personalized healthcare and wellbeing service with making a significant contribution to user [5].

Our solution first collects human movement data (i.e., acceleration, steps and speed) to recognize five targeting single motions (i.e., sitting, squat, slow walking, normal walking and fast walking) through the Android smartphones in real time. Then, it adopts a pre-processing method of movement data to remove overlap sections among different motions. Because the collected sensor data has some errors and too large deviation for reliable motion recognition. And it provides a way to calculate distribution area of each movement data to classify a target motion. To recognize a consecutive motion, an adaptive movement data weighting method is proposed which adjusts weights depending on the tendency of the aggregated movement data. Because the consecutive motion includes various motions which have different scale of movements. The proposed method can adjust features extracted from the movement data quickly and stably at the beginning and end of transition to avoid the degradation of recognition accuracy.

As a result, for detecting a vulnerable condition, we propose a situation recognition mechanism based on adaptively weighting values and past state information through the state transition model. The state transition model describes a target state and its entry and an escape condition. And when the proposed system detects the vulnerable condition, the system notifies this condition to user's family or medical personnel. Our experimental results for performance evaluation show that the proposed mechanism recognizes a consecutive motion and a user's situation correctly and quickly.

Existing studies have attached multiple sensor nodes to the body to recognize a motion based on detailed movement data. These methods have constraints that ensure correct attachment location and angle. An image-based recognition technology is also required in environments where subjects can be continuously filmed. However, the proposed method recognizes target motions based on a smartphone and is advantageous for application development. In addition, Machine Learning (ML) and Deep Learning based recognition techniques has a huge amount of operational overhead (i.e., learning time and calculation overhead) and disadvantages to real-time recognition on a mobile device. It is also difficult to extract the desired knowledge for recognizing target motions from the learning data properly. Therefore, in order to correctly extract an acceptable level of knowledge from the learning data, it is known that appropriate parameters should be entered empirically and that an approximate approach should be selected and applied in accordance with the nature of the learning data [9]. So, it is difficult to recognize complex situations with frequent changes according to historical condition.

However, the proposed method can easily define the condition required by the target application in the status transition model.

The rest of paper is organized as follows. In Section 2, we survey some related work on existing human motion recognition methods. In Section 3, we propose the consecutive motion and situation recognition mechanism. This section describes a Local Maximum Value (LMV) based real-time motion classification method and an adaptive LMV weighting method for consecutive motion recognition. In addition, we propose a situation recognition mechanism for detecting a vulnerable condition. Section 4 provides evaluation results about the proposed mechanism. Finally, we conclude with remarks on future work in Section 5.

2. Related Work

Human motion recognition contains two sub-types, the multi-sensors-based approach and the single sensor-based approach. The multi-sensors-based approach has advantages as it can generate a variety of contexts using relatively few computational resources based on data collected through many different types of sensors. However, such systems tend to be complex and cumbersome to attach many different sensors, thus, it is difficult to apply to mobile environments. In this regard, the single sensor-based approach is more appropriate for mobile environments [3].

[2] proposed activity and location recognition using wearable sensors. it recognizes a user's location, detects transitions between preselected locations, and recognizes and classifies sitting, standing, and walking behavior. Also, they find an optimal sensor position on the body to improve activity recognition by adding different types of sensors. The first sensor is a leg module that is attached at the trouser pocket to measure the acceleration and angle of the user's thigh. And the second sensor is a waist module that is attached to the middle of the user's waist to detect direction as the person moves. The proposed system consists of three function blocks; a sensing block, a unit motion recognizer and a location recognizer. The sensing block reads the sensing data and executes pre-processing. When the unit motion recognizer identifies predefined type of motions, the location recognizer calculates the current displacement vector. For evaluation, each subject walked 20 cycles of level behavior at three speeds, upstairs and downstairs with 24 steps. The proposed method calculated the standard deviation of the forward acceleration, upward acceleration, and the thigh angle, respectively. As a result, the proposed method has recognition ratios of 95.9% for level, 94.3% for upstairs and 92.8% for downstairs. Also, based on N given locations, they build N*(N-1) location transition vectors for the total number of paths recognition.

[5] proposed a method to recognize complex activities with common classifiers by a smartphone which contains activity sensing, features extraction, activity categorizing and activity classification. The activity data was collected about the subject of preparing a breakfast scenario. [5] defined four main activities: preparing for the breakfast, tea, sandwich, then clean up. Those activities are categorized into three hierarchical levels according to the activity complexity such as, high, middle and low level. This study was tested by two smartphones to collect data at a specific location. One smartphone was fixed in armband of the subject and the other in his waist mount. The collected data were pre-processed to extract the two features that is mean and standard deviation. And these features data were classified by seven representative classifiers (i.e., C4.5 (J48), logistic regression (LR), Neural Network (NN, multilayer perception), Naive Bayes (NB), k-nearest neighbor (KNN), Decision Table (DT), and Support Vector Machine (SVM)). Those classifiers are commonly used for motion recognition. The default settings of those classifiers is WEKA 3.7.10 [10]. As a result, the highest accuracy was got by KNN classifier. It has the highest accuracy in both positions of the three activity levels. The KNN classifier got 91.05% recognition accuracy for low level activities and 82.15% for high level activities.

[11] collected acceleration and rotation data of 16 participants on 13 activities through the iPod Touch. The target 13 activities were sitting, walking, jogging, and going upstairs and downstairs with different paces. They calculate mean and variance of acceleration and gyroscope on each axis, vector magnitude of acceleration. Also, to extract frequency feature they used fast Fourier transform for acceleration magnitude of each axis. This study presents a comparison result with different ML based classifiers (i.e., decision tree, multilayer perception, NB, LR, KNN, and meta-algorithms). As a result, the KNN classifier got highest accuracies (up and down stair walking: 52.3% – 79.4%, jogging: 91.7%, walking on level ground: 90.1%–94.1%, and sitting: 100%).

[4] proposed a method to detect the human motion type through sensors data collected by a smartphone. Indeed, while determining the current user's location and direction might be insignificant (e.g., by using global positioning system (GPS)). Thus, they argued that recognizing how the user is moving (e.g., by car or by train) is a more challenging issue. Also, if such detection is performed in an automatic way, considerable efforts are

required. To this aim, they have evaluated different feature extraction techniques and classification algorithms. And they have demonstrated that the joint utilization of sensors data from multiple sources (e.g., accelerometer and gyroscope) can improve the accuracy of the classifier. Moreover, they proposed an history-based recognition technique to reduce the occurrences of classification errors by exploiting the temporal correlation among consecutive sensor readings. The proposed recognition technique integrated the motion type recognition algorithm into an Android application. It utilizes a smartphone profile to associate with motion recognition, and can share recognized information with other context-aware applications.

Interface technologies that control various intelligent equipment and devices (e.g., robot, intelligent air conditioning, physical sense game machines) are evolving in a way that utilizes human motion as well as conventional input/output devices. Instead of the conventional human-computer interaction, a human motion recognition-based interaction technique has a broad prospect of application and achievements. Therefore, [12] proposed a new method of fast human motion recognition based on depth sensors of Kinect for human-computer intelligent interaction. Kinect can see the three-dimensional world from the depth image. On the basis of the depth image, the internal part of the body is segmented into 32 parts by ML theory to form bone data. And through the Kinect coordinate system, they can get the three-dimensional coordinates of each node. Also, they proposed action feature extraction which extracts four key points, both hands and elbows. Based on extracted features, they analyzed sequence of angle and distance between some pair of points. The proposed Fast Dynamic Time Warping (FDTW) algorithm-based recognition system has a good recognition result with different situations such as, complexity of the environmental background, the intensity of ambient light, the position change of the tester, the size of the tester, and the interference of multiple people. The proposed system got the successful recognition rate reached 96.2% for vector-based extraction and 97.5% for angle and distance-based extraction.

[13] proposed a novel algorithmic approach of an unobtrusive fall detection system, which is based on skeleton detection of a human figure for a real-time scenario with the Kinect sensors. It consists of the infrared projector with Infrared Radiation (IR) camera for accurate depth maps, a color camera, and a four-microphone array. The proposed fall detection methodology is based on skeleton extraction. They defined the fallen as follow; if the body motion includes changes of Y or Z coordinate, a person sitting or standing is in a situation of evidently fallen [13]. However, such an image-based approach usually needs a complicated calibration procedure and does not fit in the mobile environments [3].

[14] defined various fall types according to the normal activity of an elderly person's daily life (i.e., sitting downs and standing up from armchair, kitchen chair, toilet seat, car seat, bed and walking). They attached bi-axial gyroscope sensors at the sternum. In this environment, [14] proposed a threshold-based fall-detection algorithm based on the angular velocity. The gyroscope sensor measured pit and roll angular velocity signals. And the measured signals were low-pass filtered and calculated the resultant vector by taking the root-sum-of-squares. There are three thresholds for fall-detection, the first threshold is a lowest resultant angular velocity, the second threshold is a lowest angular acceleration, and the third threshold is a lowest recorded trunk angle signal. With these thresholds, the proposed algorithm removes no fall situations. Through analysis of the 240 recorded falls, they provided 100% fall detection accuracy.

[15] proposed a fall detection system using both accelerometers and gyroscopes. It recognized different kinds of falls (e.g., falling forward, backward, leftward, rightward, vertically). The three-axial accelerometers attached at different body locations chest and thigh. Linear acceleration and angular velocity which are measured by two accelerometers is utilized to determine whether motion transitions are intentional. Also, they proposed posture recognition about standing, bending, sitting and lying by different inclination angles of the trunk and thigh. For evaluation, they provided a special case study, sit down fast and fall on stairs through the analysis of angular rate, acceleration and inclination. Also, they provided continuous monitoring with 70 records which include fall-like motions. As a result, the 91% of fall event is detected.

[16] provided analysis on natural physical properties and temporal recurrent transformation possibilities of human motions. And they propose a Recurrent Transformation Prior Knowledge-based Decision Tree (RT-PKDT) model for recognition of specific human motions. In order to solve data inadequacy problem, RT-PKDT makes the most of sensor streaming data and human knowledge to compensate by using a temporal information and hierarchical classification method. They decided a target motion which most frequently appearing for medical care and emergency rescue scenarios including standing, lying, walking, running, walking upstairs/downstairs and elevator up/down. And they attached five sensor nodes which include accelerometer, gyroscope and barometer at the shoulder, waist, wrist, knee and ankle. The proposed RT-PKDT recognizes the

target motion according to transition schematic models which describe possible transferability between motions. Also, they provided comparisons with other methods such as DT, KNN, SVM, etc. As a result, the RT-PKDT had the highest classification accuracy 96.6%.

For better comparison, Table 1 describes the features of relative motion recognition methods. Most recent researches recognize target human motions by learning sensor data based on ML techniques such as [2, 4, 5, 11]. Therefore, it must process huge amount of data that is difficult to process in real-time through a constraint device such as a smartphone. According to the nature of the learning data, suitable ML solutions (e.g., DT, KNN, SVM, etc.) should be selected. And ML based techniques must enter appropriate environmental parameters that are finely analyzed/selected by data analysis experts to extract the desired knowledge (i.e., target motions) from the learning data and to add a new target motion for a particular purpose.

[4, 12] are an image-based motion recognition method. Therefore, it is necessary to install the camera where the target user can be filmed continuously. [13-15] proposed a fall detection method with various type of chairs or fall directions. [16] proposed a knowledge-based DT classification method with the transition model and 8 target motions. However, [16] installs 5 sensor nodes to collect motion data, so it is difficult to apply as a technique for general applications. Except [5], other methods [2, 4, 11-16] designed to recognize a single (atomic) motion. However, activities of real life style appear in a mixture of various motions.

Table 1. Comparisons with relative motion recognition method

	Target motions	Sensor Type	Sensor Location	Characteristics	Accuracy
[2]	3 motions: sitting, standing and walking	Accelerometer, compass, gyroscope	2 nodes: leg and waist	Atomic motion, Location transition, Fuzzy rules.	Atomic: 92.8% - 95.9%
[4]	Car, train and walking	Accelerometer, gyroscope and GPS	1 phone: anywhere	How user is moving, ML based (Random Forest, SVM and NB)	Atomic: 56.1% - 98.8%
[5]	4 motions: preparing breakfast, tea, sandwich and clean up.	Accelerometer	2 phones: armband and waist	Consecutive motion, ML based (J48, KNN, DT, SVM, etc.)	Consecutive /Low level: 91% /High level: 82.1%
[11]	13 motions: sitting, walking, jogging, and going upstairs and downstairs	Accelerometer and gyroscope	1 phone: anywhere	Atomic motion, ML based (DT, LR, KNN, etc.)	Atomic: 52% - 94.1%
[12]	6 motion: raise hand, applaud, swing leg, bend over and squat	Infrared projector and IR camera	1 Kinect: front	Atomic gesture, Depth image-based skeleton extraction	Atomic: vector based: 96.2% Angle and distance based: 97.5%
[13]	Fall	Infrared projector and IR camera	1 Kinect: front	Fall, Depth image-based skeleton extraction	Reliable (no number)
[14]	Fall with various chair (car seat, stool, armchair toilet seat)	Gyroscope	1 node: chest	Fall, Angular velocity threshold based	Atomic: 100%
[15]	Fall (forward, backward, leftward, rightward, vertically)	Accelerometer and gyroscope	2 nodes: chest and thigh	Fall, Acceleration and angular velocity threshold based	Atomic: 91%
[16]	8 motions: standing, lying, walking, running, up/downstairs and elevatorUp/Down	Accelerometer, gyroscope and barometer	5 nodes: shoulder, waist, wrist, knee and ankle	Atomic motion, transition model, Knowledge based DT	Atomic: 96.6%
This paper	8 motions: sit, squat, walk, driving, fall, etc.	Accelerometer	1 phone: anywhere	Consecutive motion, LMV threshold based, transition model	Atomic: 92.7% Consecutive /30 sec: 78.8% /60 sec: 92%

3. Consecutive Motion and Situation Recognition Mechanism

The activities in real life appear in a mixture of various motions. For example, you can keep walking at a normal speed for 10 seconds and then sit at the next moment, or switching to fast walking. Therefore, motions should be recognizable in a form of consecutive motions including many transitions.

Unlike motion recognition, a situation is difficult to recognize by only movement data because similar values of movement data can be collected, even in different situations. For example, driving situation is usually measured a very high speed than the walking, but depending on traffic conditions, it has very slow speed, and sometimes it can be stopped by signals. Therefore, it is difficult to distinguish between the walking motion and driving situations with only movement data such as acceleration and speed.

Similarly, to recognize a vulnerable condition, historical status analysis (e.g., before and after state information of the accidental event) as well as consecutive motion recognition are required. For example, if you are in a normal movement immediately after the fall, it's not a vulnerable condition. However, if there is no movement after a fall, it would be a vulnerable condition where the shock caused by the fall can lead to loss of consciousness or inability of a normal motion due to serious injury. Thus, a situation means that it has the same motion data but can be defined by a combination of states.

Therefore, a consecutive motion and situation recognition mechanism is proposed. As shown in the Figure 1, the proposed system was implemented as Android Smartphone Apps for real-time movement data collection and vulnerable condition notification. It consists of three sub parts, 1) movement data weighting function, 2) consecutive motion and situation recognition and 3) collected data repository to recognize vulnerable condition. First, in Section 3.1, we describe a preprocessing method of movement data to remove overlap sections among different motions and calculation of distribution area of movement data to classify target motions. In Section 3.2, we propose an adaptive movement data weighting method for consecutive motion recognition. It adjusts weights depending on the tendency of the aggregated movement data. In Section 3.3, we propose a situation recognition mechanism based on adaptively weighted values and past state information through the state transition model. When the proposed system detects the vulnerable condition, the system notifies this condition to user’s family or medical personnel.

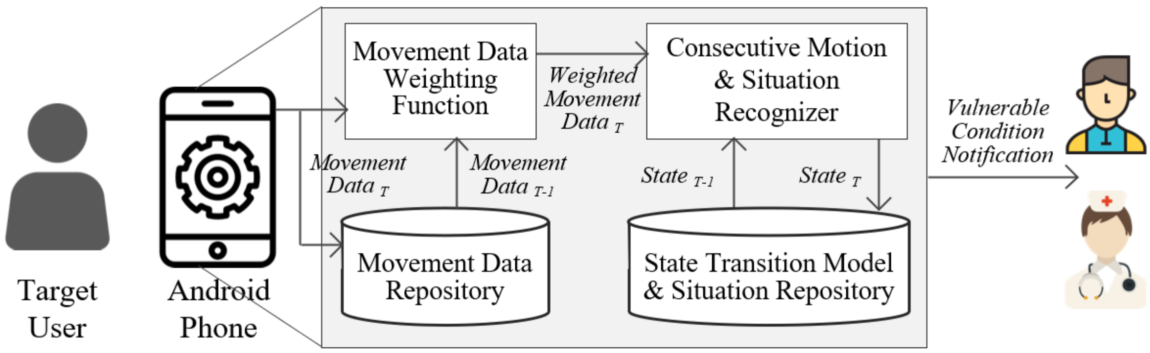


Figure 1. System model with a consecutive motion and situation recognition mechanism

3.1. LMV based real-time motion classification method

Here, we describe a preprocessing method of movement data to remove dependency of sensor attached position and the overlap section, and to reduce the deviation. First, we calculate the LMV based on Acceleration Vector Magnitude (AVM) among the various movement data analysis methods. Generally, movement sensors are used to collect movement data such as acceleration and angular velocity. However, acceleration sensors have different collected angular values depending on the position and direction of attachment of the sensor. So, the criteria of each axis are changed by various factors, such as posture, angle, etc., of the sensor which attached the target. We collect acceleration data for three axes of movement sensor included in Android smartphones every 0.1 second. Through these collected sensor data, we calculate an AVM to remove the dependence of axis. The AVM represents the magnitude of the acceleration through the Equation (1).

$$AVM = \sqrt{Acceleration\_X^2 + Acceleration\_Y^2 + Acceleration\_Z^2} \tag{1}$$

As a result, the absolute value of the movement (i.e., AVM) is extracted regardless of the sensor installation angle or position. Even though this method has limitations in the detailed motion recognition compared to the recognition method using each axis independently, we can use the movement sensor of the smartphone due to regardless of the attachment position and angle. However, AVM has a wide range of distributions depending on the size of motion. And, the distribution range of smaller motions is included in the distribution range of larger motions. As shown in Figure 2, the range of fast walking (larger motion) includes normal walking (smaller motion). So, it is difficult to distinguish between fast walking and normal walking based on AVM.

Therefore, to classify target motions, we utilize LMV to remove the overlap section among the target motions. A movement data shows consistently repeated certain patterns as large and small values. In this case, the LMV is the largest value in the local repeated area of movement data. The AVM data is collected every 0.1 second, and each time the LMV is calculated with Equation (2). The dots of normal walking graph are examples of LMV as shown in Figure 2. Through the LMV, we can get the upper boundary of AVM about a certain motion.

$$\boldsymbol{IF}(AVM_{t-1} - AVM_t \leq 0 \;\; \boldsymbol{AND} \;\; AVM_t - \;\; AVM_{t+1} > 0 \;\;) \quad \boldsymbol{THEN} \quad LMV_t = \;\; AVM_t \tag{2}$$

Figure 2. AVM and LMV of target motions

Figure 3 shows the LMV graphs for the five target motions. Each motion's LMV graph is distributed in different areas. So, we can distinguish LMV data of each motion. However, the LMV also has large deviation depending on the motion because it is a subset of the AVM. Therefore, there are still some overlap sections.

To classify target motions, we should remove the overlap section, reduce the deviation of motion data and calculate the motion's LMV distribution areas. The distribution area of a particular motion can be calculated by means of an intermediate value of other motions, distributed in the neighboring areas. Generally, the fast walking motion data is distributed higher than the normal walking motion data, and the slow walking motion data is distributed lower than the normal walking motion data. As a result, the distribution area is calculated as shown in Equation (3). The $avgLMV(LVn)$ is average LMV of the motion when movement level is n. The LMV level of five target motions describes in Table 2. Each motion's LMV distribution areas exist sequentially as shown in Figure 3.

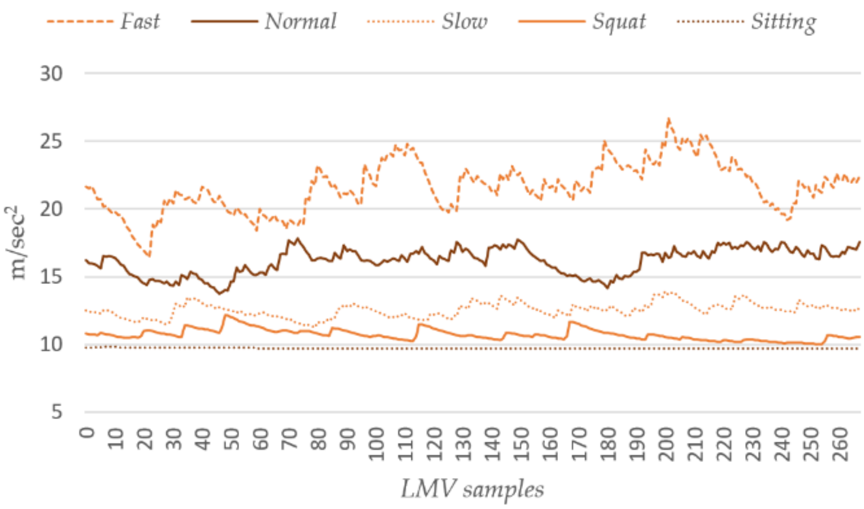


Figure 3. LMV for target motions

$$\frac{avgLMV(LVn-1)+avgLMV(LVn)}{2} < LMV(LVn) < \frac{avgLMV(LVn)+avgLMV(LVn+1)}{2} \quad (3)$$

Table 2. LMV levels of target motion

LMV LV	1	2	3	4	5	6	4 - 6
Motion	fall	siting	squat	slow	normal	fast	drive

However, collected LMV in real time have very large deviation. Thus, by updating the LMV according to the Equation (4) which is exponential weighted average, the deviation of the values can be reduced. Figure 4 shows the adjusted LMV by different weights. The tempLMV, real-time collected LMV shows a very large deviation. If the recognition is attempted using this tempLMV, the result is very unstable. When the weight(α) was 0.95 in the Equation (4), LMV data is adjusted very stable. When the weight was 0.4, the adjusted value has frequently changed according to the newly collected LMV. As a result, as a weight value is higher, the adjusted LMV has lower deviation and stable convergence, it is easy to separate the distribution areas. Finally, through the weighted LMV, current status is classified a certain motion according to LMV levels that was calculated with Equation (3).

$$LMV_t = \alpha \times LMV_{t-1} + (1-\alpha) \times tempLMV_t \quad (4)$$

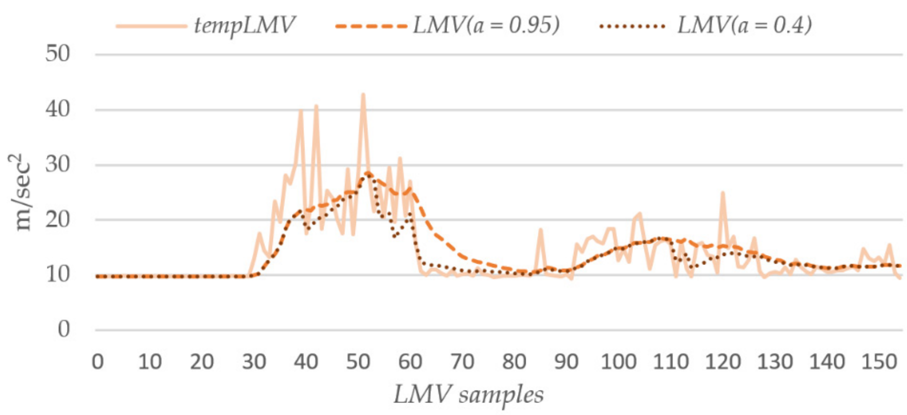
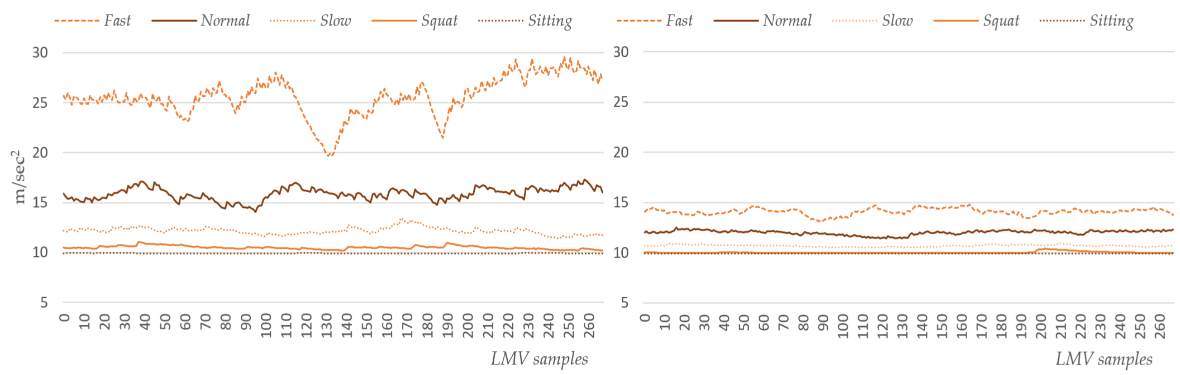


Figure 4. LMV adjusting results by different weights

To evaluate the proposed LMV based motion classification method, we provide experiments for the five target motions. And the experimental environment for the single motion classification is as follows:

The subjects consisted of three men and four women. Other detailed information is removed for privacy problems. The subjects performed five target motions (i.e., sitting, squat, slow walking, normal walking and fast walking) for 30 seconds. The experimental site is outdoors and wide area to aggregate stable motions and GPS signals. Also, the weight for LMV is 0.95. And each motion was performed 50 times over.

Figure 5 shows LMV of the five target motions of subject 1 and 2. In case of the sitting motion of subject 1, LMV was 9.8 m/sec^2 that equals to the acceleration of gravity in all ranges due to the almost no change in acceleration [17] as shown the Figure 5(a). The squat motion was measured at a slightly higher area than the sitting motion and the slow walking motion was distributed in the middle of the sitting and normal walking motion. For the fast walking, there was a large deviation compared to other motions. As a result, the LMV values of the five motions were distributed in their respective areas. Four motions had a 100% recognition rate about 258 samples. On the other hand, in case of the fast walking motion, since the LMV was rapidly reduced near the 131th sample, the recognition rate was 97.7%, however, it showed a good recognition result generally. Figure 5(b) shows LMV of the five target motions of subject 2. This result had a very narrow gap in the distribution area, and all motion data distributed in areas lower than subject 1. Because the subject 2 had smaller motions than subject 1. Nevertheless, the result had a narrow gap, subject 2 had 100% recognition rate except for the squat motion due to the collected data was very stable continuously. However, the Squat motion was often recognized as the sitting motion when the LMV didn't change at 9.8m/sec^2.



(a) (b)

Figure 5. LMV of target motions of subjects: **(a)** Subject 1; **(b)** Subject 2.

3.2 Adaptive LMV weighting method for consecutive motion recognition

An activity in real life is a consecutive motion which consists of various motions and its transitions which have a big difference. Also, there is an exceptional data generated by sensor errors. Therefore, an adaptive weighting method which is noise resistant as well as accommodates motion changes are needed.

Algorithm 1 describes the proposed adaptive LMV weighting method. In this method, if the newly measured LMV has decreased less than 80% compared to the previous value, the method changes the weight to a reduced rate and LMV adjusting ($\alpha = nLMV / pLMV$). With this way, newly collected LMV has a bigger weight when the incoming LMV is decreased rapidly to recognize transition from a higher motion to a very lower motion like a walking and sitting. If LMV variation is reduced even if the values have not changed significantly, the motion is been converted completely to another motion and movement data are considered to be collected reliably. Therefore, in this case, the weight is set to 0.7 to stabilize the weighted value. In other conditions, there are no status transition and the movement data maintained continuously, so the weight is set to 0.95 for stable weighting and noise-resistant.

Algorithm 1: Adaptive LMV weighting method

Input : nLMV, //newly measured LMV
pLMV, // previously measured LMV
ndiffLMV, // difference of nLMV
pdiffLMV // difference of pLMV
Output : α //weight

```
IF   nLMV / pLMV < 0.8   THEN
     RETURN α = nLMV / pLMV
ELSEIF   ndiffLMV / pdiffLMV < 0.5 THEN
     RETURN α = 0.7
ELSE
     RETURN α = 0.95
```

3.3 Situation recognition mechanism for detecting vulnerable condition

A vulnerable condition should be detected based on various sensor information and previous situations. Thus, in this paper, the situation recognition mechanism for detecting a vulnerable condition is proposed as shown in Figure 6. The proposed mechanism consists of three parts as follows:

- Calculate the distribution range of movement data: movement data is collected through the Android sensor interface in every 100ms. The collected data are speed, steps and AVM values which are calculated based on the acceleration x, y, z values. It obtains an LMV from the collected AVM and calculates the distribution range about each motion through Equation (1).
- Real-time movement data collection and adaptive LMV weighting: The collected AVM is clipped to remove abnormal values. And variation of SMV is calculated for LMV extraction and adaptive weighting. The extracted LMV is adjusted with the adaptive weighting method for stable recognition. For recognition of consecutive motions effectively, we change a weight adaptively as described in Algorithm 1 in Section 3.2.
- Motion classification and state transition model: Current state is recognized based on the weighted LMV, speed and number of steps. Depending on the previous state, the criteria for recognizing the current state will be different. It will be switched according to the proposed state transition model as shown in Figure 7. Based this proposed state transition model, we recognize abnormal conditions (e.g., driving, fall and vulnerable conditions). And the normal condition (i.e., sitting, squat, slow walking, normal walking and fast walking) is recognized by the motion classification algorithm. However, the Android device does not measure the correct step count immediately when the movement is not measured for a period of time. Therefore, the existence of valid step information determines the detailed motion classification algorithm. So, we propose two types of motion classification algorithms depending on the step information as described in Algorithm 2 in this section.

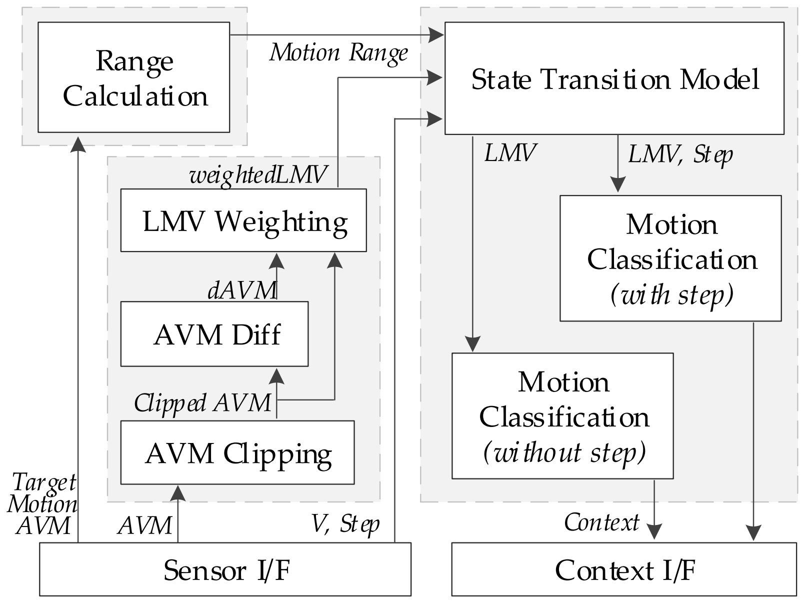


Figure 6. Situation recognition mechanism for vulnerable condition

Figure 7 shows a state transition model for the abnormal condition. It defines state and its entry and escape conditions (i.e., transition condition). Basically, it starts in a *Normal* state and if the transition condition is satisfied, it transits to another state. For example, in case of the *Fall*, it can be recognized if AVM drops drastically below 0.3 m/sec^2. The *Vulnerable* state can only be transited if the previous state is *Fall* and also the *FallCNT* is greater than 5. That means there is no movement for five seconds after the *Fall*. In case of the *Driving*, it can be transited from *Normal* when the *Step* is zero and the velocity(V) is faster than the half of user's stride (*Stride* * 0.5). The *Driving* state has higher speed than *Normal* state but also, does not measured steps. Therefore, the escape condition of *Driving* state is defined as if measured available step counter without an error. When satisfied this condition, the *Driving* state is transited to the *Normal* state. As a result, we can add new rules for other conditions at the state transition model for situation recognition.

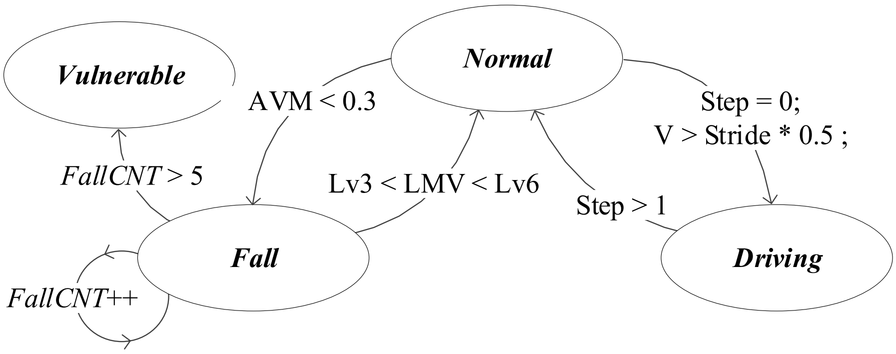


Figure 7. State transition model

Algorithm 2 is the proposed motion classification algorithm. The ordinary person's step count is measured within 0 to 3 times per second. However, if an Android device has a step count error, it cannot measure a step for a certain period of time, about two seconds. And after the steps measured an error, the counted step value during the error period is added to the next step value. So, the step count has a big value (over than four times per seconds) when an error occurs. Therefore, the proposed motion classification algorithm ignores the step data if there is a step error and checks the speed to transit *Driving* state as shown in Algorithm 2(a). Then, the motion is classified in order of small motion size according to an LMV level and distribution area which calculated with Equation (3). If the steps data is valid, the algorithm is divided two flows when the step is zero or not as shown in Algorithm 2(b). The *zeroStep* is a variable that increases per second until the step is zero continuously. When the *zeroStep* is greater than 1, it is actually *Squat, Sitting*, or *Driving* which is not measured a step actually. And the other cases are classified by the LMV value. If the step is continuously measured (if the *zeroStep* is zero), it is *Slow, Normal*, and *Fast* motion.

Algorithm 2(a): Motion classification algorithm with step counter error	**Algorithm 2(b):** Motion classification algorithm without step counter error

Input : speed, stride, LMV
Output : motion type
 	
IF speed >= stride*3 **THEN**
    RETURN *Driving*
ELSEIF lmvLV1 < LMV < lmvLV2 **THEN**
    RETURN *Sitting*
ELSEIF LMV < lmvLV3 **THEN**
    RETURN *Squat*
ELSEIF LMV < lmvLV4 **THEN**
    RETURN *Slow*
ELSEIF LMV < lmvLV5 **THEN**
    RETURN *Normal*
ELSE
    RETURN *Fast*
	Input : speed, stride, LMV, zeroStep
	Output : motion type
	
	IF zeroStep > 1 **THEN**
	// when the step counter data was not available.
	    IF speed >= stride*0.5 **THEN**
	        RETURN *Driving*
	    ELSE
	        IF lmvLV1 < LMV < lmvLV2 **THEN**
	            RETURN *Sitting*
	        ELSE
	            RETURN *Squat*
	ELSE
	// when the step counter data was available.
	    IF LMV < lmvLV4 **THEN**
	        RETURN *Slow*
	    ELSEIF LMV < lmvLV5 **THEN**
	        RETURN *Normal*
	    ELSE
	        RETURN *Fast*

4. Performance Evaluation Results

In this section, we describe the performance evaluation results of the proposed consecutive motion and vulnerable condition recognition. For performance evaluation, we implemented the proposed mechanism as the Android application. The subjects perform the consecutive motions with the smartphone in their trouser pocket. At first, we measured fall recognition as a vulnerable condition.

Figure 8 shows the LMV and AVM of a vulnerable condition. The upper side graph shows AVM and temporal LMV (*tempLMV*) and weighted LMV (*LMV*). The *tempLMV* is newly measured LMV in real time from AVM and the *LMV* is adjusted value by the adaptive weighting method. The bottom side graph shows the recognized results by eight states, (i.e., sitting, squat, slow walking, normal walking, fast walking, driving, fall and vulnerable). To describe detailed recognition result in every time when collect AVM, the horizontal axis of the graph is the order of the measured AVM data. The AVM is measured 10 times per second and calculate LMV.

The experimental results showed that a normal AVM was measured until 3 seconds. However, after 3 seconds, the AVM was measured close to zero. Usually, the AVM is almost 9.8 m/sec^2 even if there is no movement due to the acceleration of gravity. On the other hand, in this case, if the AVM is very close to zero, it can be considered a free fall state to the direction of gravity. So, the proposed mechanism detects the fall state when AVM is almost zero. This approach had a fall detection rate of 100 percent in more 50 fall experiments. After a fall is detected and if normal motion is measured immediately, it is considered no problem. However, after the fall, the normal movement is not detected for a period of time, it can be considered a very vulnerable condition in which fall damage can cause immobilization or loss of consciousness. So, the proposed mechanism is determined as a vulnerable condition if the motion is not continuously measured for 5 seconds. Therefore, it is recognized the vulnerable condition at 8 seconds which is 5 seconds after the fall was detected as shown the recognition result graph of Figure 8. If the normal movement is detected continuously after a vulnerable condition, it switches back to the normal state. As shown in Figure 8, the normal motion was continuously measured for about 5 seconds from around 18 seconds to 23 seconds. So, the state is changed to the squat at 23 seconds.

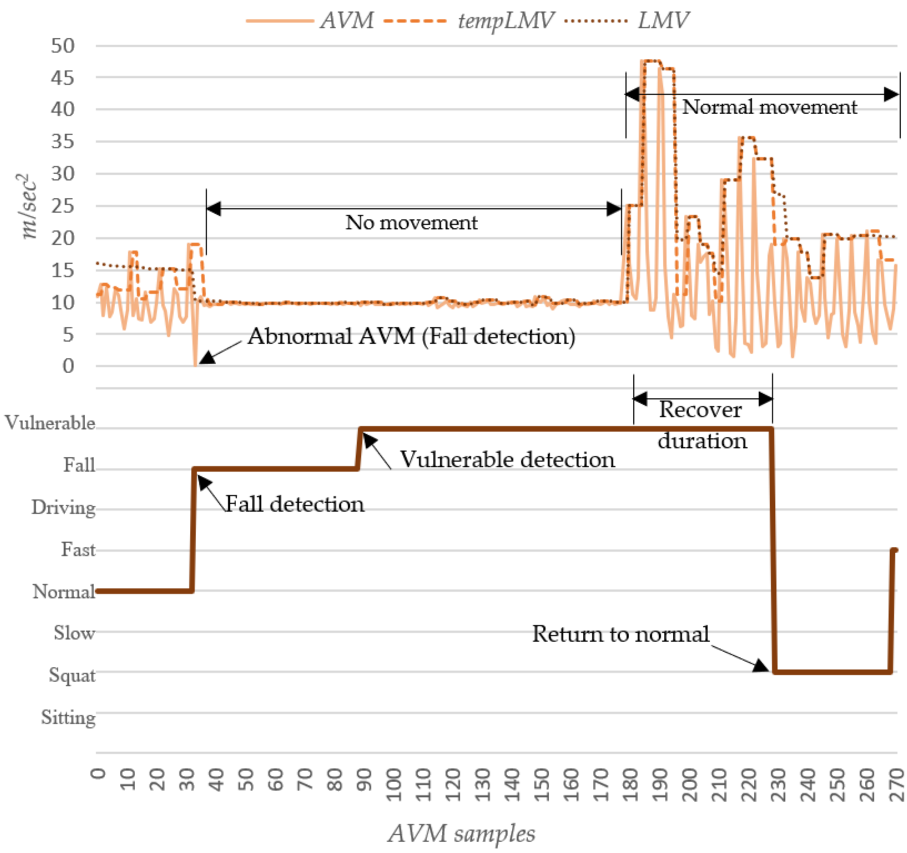


Figure 8. Vulnerable condition recognition results

We defined three types of consecutive motions to analyze the performance of the proposed mechanism. The subjects perform a specific motion during the holding time and then switches to another motion. The holding time was set to 15, 30, and 60 seconds. The defined three types of the consecutive motion sequences as follows:

- Down sequence: the sequential transition from the most active motion to the least active one. (Fast walking -> Normal walking -> Slow walking -> Squat -> Sitting)
- Up sequence: the sequential transition from the least active motion to the most active one. (Sitting -> Squat -> Slow walking -> Normal walking -> Fast walking)
- Mix sequence: the sequence of motion is shifted to Sitting -> Fast walking -> Square -> Normal walking -> Slow walking like a random mixture. It shows the consecutive motions that are not related to each other when the motion is switched.

If the holding time is very short, it is affected by the delay time required for the subject to recognize the next motion and to perform the actual motion. Therefore, we explain the experimental results when the holding time is 30 seconds firstly. Figure 9 shows the recognition results of the down sequence which is the first consecutive motion sequence type when a holding time is 30 seconds. The upper part of Figure 9 shows the collected data. The *tempLMV* is newly measured LMV and the *sLMV* are adjusted values by exponential averaging with static weight 0.95. The *aLMV* are adjusted LMV with the adaptive weight as proposed in Algorithm 1. The bottom of Figure 9 shows the recognition results of the proposed mechanism. The criteria in the bottom graph are the reference state which is the basis of the performance evaluation.

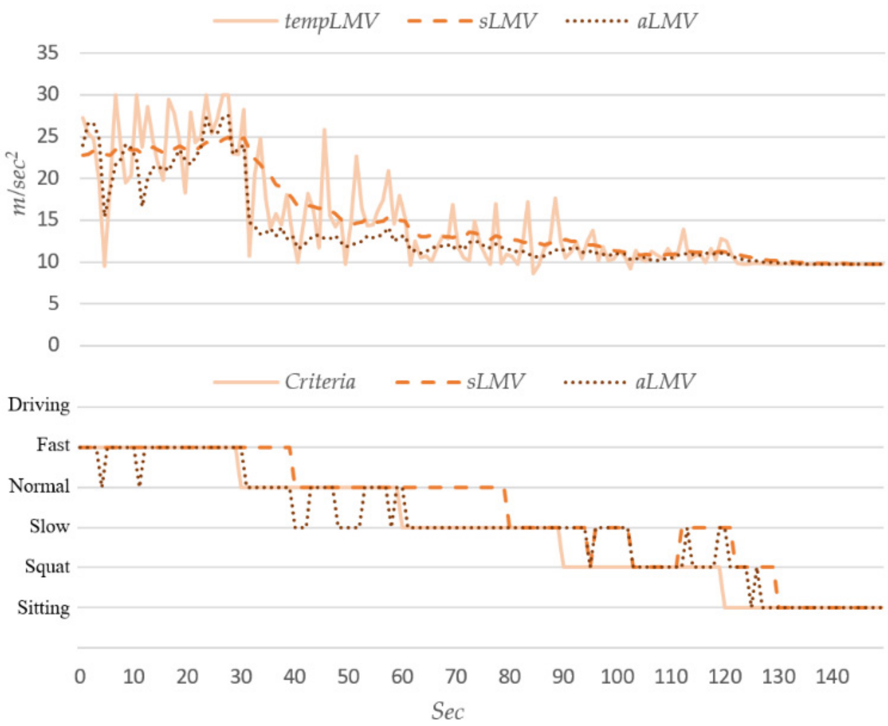


Figure 9. Recognition results of the down sequence (Holding time = 30 sec.)

The subjects keep the fast walking motion until the first 30 seconds and then shifts to the normal walking motion in the second area. The *tempLMV* values show a very large deviation. The static weighting method, *sLMV* have very little effect on the newly collected LMV because it is mostly concentrated on the collected previous LMVs (95%). So, it is recognized as the fast walking motion up to 40 seconds. This result means there is a recognition delay about 10 seconds that is switching time of the static weighting method. We have confirmed that the higher weight occurs recognition delay at the transition. At around 30 seconds, in case of the adaptive weighting method, the *aLMV* is changed to a lower value than the static weighting method rapidly. On the other hand, if *tempLMV* are collected at lower values, the adaptive weighting method can be changed sensitively compared to the static weighting method, resulting in recognition errors such as around 40 seconds and 50 seconds.

In this paper, the recognition rate is defined as the ratio of the number of samples which is correctly recognized with the target motions. As the overall recognition rate was 61.1% for the static weighting method and 78.2% for the adaptive weighting method, the adaptive weighting method showed the high performance of 17.1%. Because of the characteristic of the down sequence that changes to a smaller motion, the static weighting method has a very slow conversion to a new value. On the other hand, the proposed adaptive weighting method changes weights adaptively when the *tempLMV* rapidly decrease.

Figure 10 shows the recognition results of the up sequence which is the second consecutive motion sequence type with a holding time is 30 seconds. At the 30 seconds, 60 seconds, and 120 seconds, due to a transition of motion, we can see the little recognition delay. In this sequence, the static weighting method gets good recognition results because it adjusts the *sLMV* very stable. In case of the adaptive weighting method, the recognition results were slightly better than the static weighting method at the first transition (around 30 seconds). However, at near 75 seconds and 100 seconds, there were errors that recognized as a lower motion by a few lower *tempLMV* values. As the recognition rate was 84.6% for the static weighting method and 77.6% for the adaptive weighting method, the adaptive weighting method showed 7% lower performance.

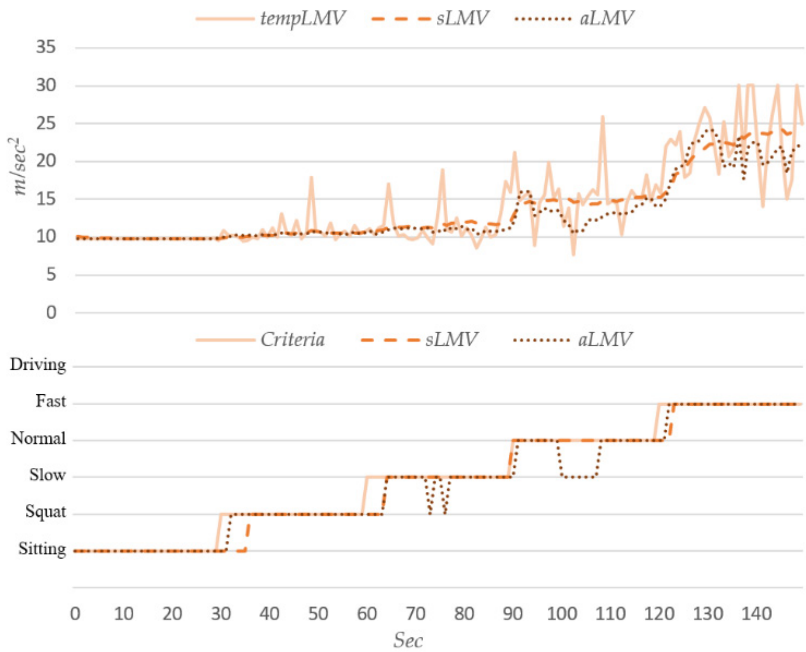


Figure 10. Recognition results of the up sequence (Holding time = 30 sec.)

Figure 11 shows the recognition results of the mix sequence which is the third consecutive motion sequence type with a holding time of 30 seconds. This sequence consists of different large motion transitions. At the 30 seconds, the first transition, very large *tempLMV* was measured (from the sitting to the fast walking motion). Thus, both methods had a recognition delay of about 5 seconds. However, at 35 seconds, the *aLMV* was measured 26 m/sec^2 and *sLMV* was 20 m/sec^2. This is because the proposed adaptive weighting method is capable of rapid renewal when switching to other motions.

At the 60 seconds, the second transitions, the sequence changed from a very large motion to small motion (from the fast walking to the squat motion). In case of the static weighting method, the *sLMV* is adjusted very slowly, even if collected a very low *tempLMV* constantly. Therefore, the recognition results in areas of 60 to 90 seconds are very poor. On the other hand, for the adaptive methods, the weight is highly adjusted when the low *tempLMV* is collected, so the transition to the new value is very quickly. It takes less than 5 seconds to change from 25 m/sec^2 to 10 m/sec^2. Therefore, the recognition rate is higher than the static weighting method. At the area of 120 to 130 seconds, the static weighting method has a recognition error (recognizes the slow walking to the normal walking), however, the adaptive weighting method is recognized correctly. As a result, the adaptive weighting method has a recognition rate of 78.8% and the static weighting method has 70% in the mix sequence which contains large difference motions like an actual daily life. The difference in the recognition rate of the two approaches is about 8%, but it is effective for real-time applications by reacting sensitively for switching motions in real time.

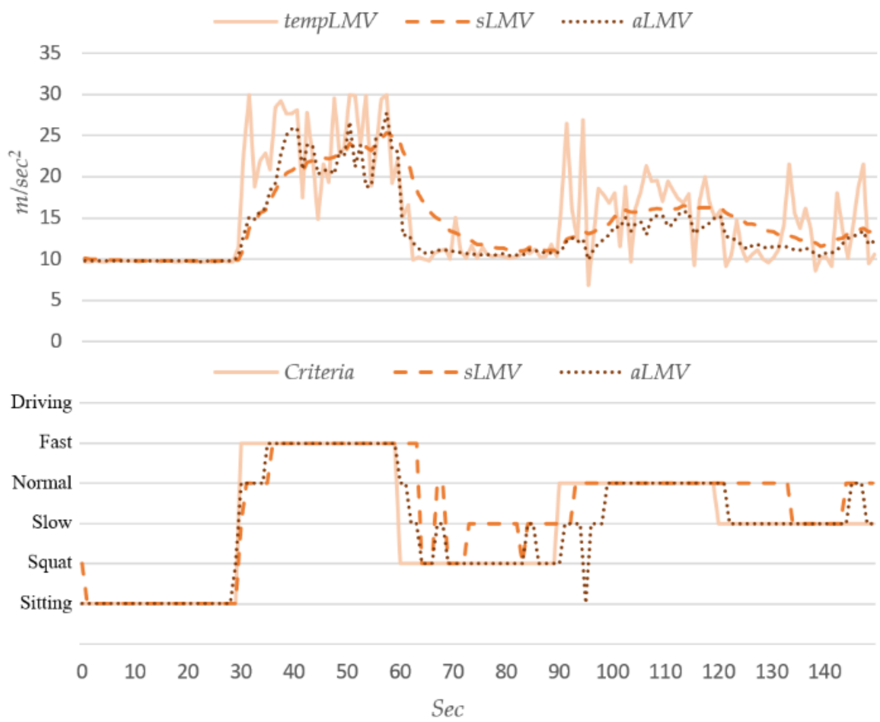


Figure 11. Recognition of the mix sequence (Holding time = 30 sec.)

To check the difference in recognition results according to the motion holding time, performance evaluation was performed based on the holding time is 15 seconds and 60 seconds. Figure 12 shows the recognition results of the mix sequence when a holding time is 15 and 60 seconds. As shown in the Figure 12(a), due to the very short holding time, the delay of the motion transition has a great impact. In addition, because of the limitation of LMV changing speed in recognizing converted movements, if the holding time is short, the recognition rate is very poor. As a result, the recognition rate was 60% for the static weighting method and 64% for the adaptive weighting method. Figure 12(b) shows the recognition results when a holding time is 60 seconds. In this case, there are more stable LMV areas because of the longer holding time. Therefore, the recognition rate has also improved. At 120 seconds, transitions from a very large motion to a small motion (from the fast walking to the squat motion), the adaptive weighting method adjusts LMV very quickly compared to the static weighting method. As a result, the recognition rate was 85% for the static weighting method and 92% for the adaptive weighting method. Due to the fast LMV updating in a transition point, the adaptive weighting method has a higher recognition rate of about 7% than the static weighting method.

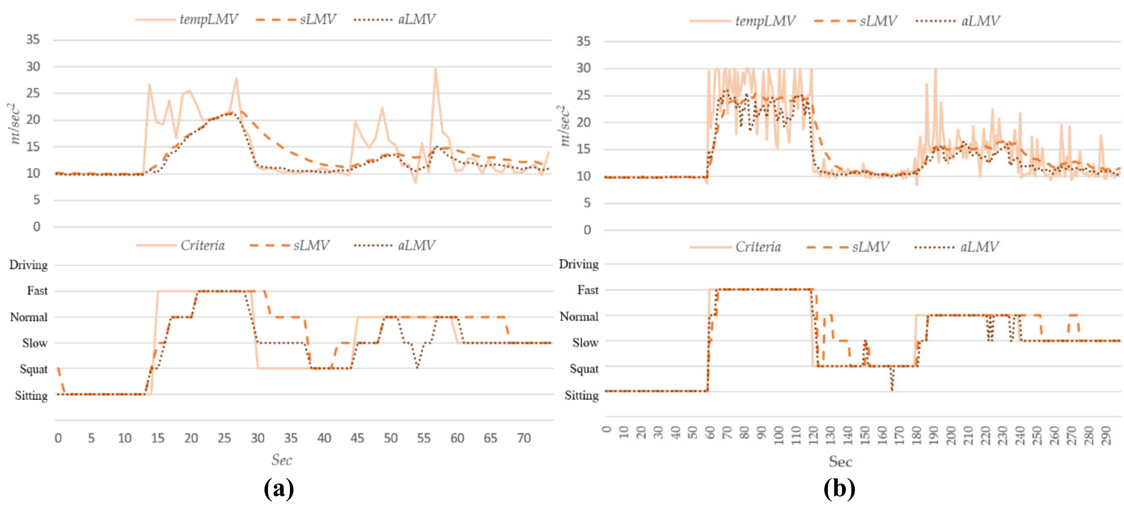


Figure 12. Recognition of the mix sequence: **(a)** Holding time = 15 sec; **(b)** Holding time = 60 sec.

As described earlier, we defined three types of consecutive motion sequences and measured motion transition data when a holding time was 15, 30 and 60 seconds. Each experiment was performed 30 times. Table 3 summarizes the averaging recognition rate of the consecutive motion based on the proposed method.

In case of the up sequence, the static weighting method has better results than the adaptive weighting method at all holding times. However, in case of the down sequence and the mix sequence, the adaptive weighting method has better results. This is because the adaptive weighting method quickly adjusts the weight when a radically low LMV is measured. As a result, the performance of the adaptive weighting method shows a good result because it's likely to appear a combination of different motion scales such as the mix sequence case in a daily life circumstance.

Table 3. Recognition results of the composite motion

Holding time	15 Sec			30 Sec			60 Sec		
Sequence type	Down	Up	Mix	Down	Up	Mix	Down	Up	Mix
Static	62%	84.6%	60%	61.1%	84.6%	70%	85.6%	93.8%	85.5%
Adaptive	84%	76%	64%	78.2%	77.6%	78.8%	89.1%	83.3%	92%
difference	+22%	-8.6%	+4%	+17.1%	-7%	+8.8%	+3.5%	-10.5%	+6.5%

5. Conclusions

To recognize the user's consecutive motion and vulnerable condition in real time while eliminating too much computational overhead, in this paper, we firstly collected human movement data to recognize five targeting single motions through the Android smartphones. To remove affecting the sensor attached position from aggregated motion data, the AVM based LMV is utilized. Furthermore, to recognize a consecutive motion in real time, we have proposed the adaptive weighting method for LMV adjustment. It changes the weight of the newly collected LMV to adjust the LMV weight sensitivity when the incoming data changed dramatically. For real-time motion classification as well as the previous state information, we have also proposed the state transition model which defines a target state and its transition conditions.

To evaluate the proposed mechanism, we defined three types of consecutive motion sequences and measured motion transition data when a holding time was 15, 30 and 60 seconds. In case of the up sequence, the static weighting method has better results than the adaptive weighting method for every holding time. However, in case of the down sequence and the mix sequence, the adaptive weighting method had better results. There was 3.5% enhancement for the down sequence and 6.5% enhancement for the mix sequence when holding time was 60 sec. This is because the adaptive weighting method quickly adjusts the weight when a radically low LMV is measured. As a result, we have successfully demonstrated that the proposed method can recognize a consecutive motion in real-time with a single smartphone and recognize a vulnerable condition based on motion and situation recognition results.

As the future work, we will perform a long-term experiment in daily life without defining a target motion to evaluate the proposed mechanism. Also, to improve recognition ratio, we will study a real-time motion transition extraction method based on ML.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

[1] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-Based Activity Recognition," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 790-808, Nov. 2012, doi: http://dx.doi.org/10.1109/TSMCC.2012.2198883.

[2] S. W. Lee and K. Mase, "Activity and location recognition using wearable sensors," IEEE Pervasive Computing, vol. 1, no. 3, pp. 24-32, Dec. 2002, doi: http://dx.doi.org/10.1109/MPRV.2002.1037719.

[3] Y. Kawahara, H. Kurasawa, and H. Morikawa, "Recognizing user context using mobile handsets with acceleration sensors," in *Proc. of the 2007 IEEE Int. Conf. on Portable Information Devices*, pp. 1-5, May 25-29, 2007, doi: http://dx.doi.org/10.1109/PORTABLE.2007.12.

[4] L. Bedogni, M. Di Felice, and L. Bononi, “By train or by car? Detecting the user's motion type through smartphone sensors data,” in *Proc. of the 2012 IFIP Wireless Days*, pp. 1-6, Nov. 21-23, 2012, doi: https://dx.doi.org/10.1109/WD.2012.6402818.

[5] M. Mobark, S. Chuprat, and T. Mantoro, “Improving the accuracy of complex activities recognition using accelerometer-embedded mobile phone classifiers,” in *Proc. of the 2017 2nd Int. Conf. on Informatics and Computing*, pp. 1-5, Nov. 1-3, 2017, doi: https://dx.doi.org/10.1109/IAC.2017.8280606.

[6] NXP Semiconductors, “NXP® Sensor Fusion,” [Online] Available: https://www.nxp.com/design /sensor-developer-resources/nxp-sensorfusion:XTRSICSNSTLBOXX

[7] F. De Backere, F. Ongenae, F. Van Den Abeele, J. Nelis, P. Bonte, E. Clement, M. Philpott, J. Hoebeke, S. Verstichel, A. Ackaert, and F. De Turck, “Towards a social and context-aware multi-sensor fall detection and risk assessment platform,” Computers in biology and medicine, vol. 64, pp. 307-320, Sep. 2015, doi: https://dx.doi.org/10.1016/j.compbiomed.2014.12.002.

[8] K. Chaccour, R. Darazi, A. H. El Hassani, and E. Andrès, “From fall detection to fall prevention: A generic classification of fall-related systems,” IEEE Sensors Journal, vol. 17, no. 3, pp. 812-822, Nov. 2016, doi: https://dx.doi.org/10.1109/JSEN.2016.2628099.

[9] J. S. Kim, Y. Cho, and T. H. Lim, "Prediction of the Location of the Glottis in Laryngeal Images by Using a Novel Deep-Learning Algorithm," IEEE Access, vol. 7, pp. 79545-79554, Jun. 2019, doi: https://dx.doi.org/10.1109/ACCESS.2019.2923002.

[10] University of Waikato, “Weka 3: Machine Learning Software in Java,” [Online] Available: https://www.cs.waikato.ac.nz/ml/weka/

[11] W. Wu, S. Dasgupta, E. E. Ramirez, C. Peterson, and G. J. Norman, “Classification accuracies of physical activities using smartphone motion sensors,” Journal of Medical Internet Research, vol. 14, no. 5, p. e130, Sep. 2012, doi: https://dx.doi.org/10.2196/jmir.2208.

[12] X. Pang and B. Liang, “Motion recognition based on Kinect for human-computer intelligent interaction,” Journal of Physics: Conference Series, vol. 1187, no. 3, pp. 1-7, 2019, doi: https://doi.org/10.1088/1742-6596/1187/3/032028.

[13] M. P. Kumar, M. Sudhakaran, and R. Seyezhai, “Kinect Sensor Based Human Fall Detection System Using Skeleton Detection Algorithm,” in *Proc. of the Int. Conf. on Engineering Innovations and Solutions (ICEIS-2016)*, pp. 15-18, Apr. 2016.

[14] A. K. Bourke and G. M. Lyons, “A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor,” Medical engineering & physics, vol. 30, no. 1, pp. 84-90, Jan. 2008, doi: http://dx.doi.org/10.1016/j.medengphy.2006.12.001.

[15] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, “Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture Information,” in *Proc. of the 2009 Sixth Int. Workshop on Wearable and Implantable Body Sensor Networks, IEEE Computer Society*, vol. 9, pp. 138-143, Jun. 3-5, 2009, doi: http://dx.doi.org/10.1109/BSN.2009.46.

[16] C. Xu, J. He, X. Zhang, H. Cai, S. Duan, P. H. Tseng, and C. Li, “Recurrent transformation of prior knowledge based model for human motion recognition,” Computational intelligence and neuroscience, vol. 2018, pp. 1-12, Jan. 2018, doi: http://dx.doi.org/10.1155/2018/4160652.

[17] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” ACM SigKDD Explorations Newsletter, vol. 12, no. 2, pp. 74-82, Dec. 2011, doi: http://dx.doi.org/10.1145/1964897.1964918.