



TrailSense: A Crowdsensing System for Detecting Risky Mountain Trail Segments with Walking Pattern Analysis

KEUNSEO KIM, KAIST
HENGAMEH ZABIHI, KAIST
HEEYOUNG KIM, KAIST
UICHIN LEE, KAIST

Trail surface information is critical in preventing from the mountain accidents such as falls and slips. In this paper, we propose a new mobile crowdsensing system that automatically infers whether trail segments are risky to climb by using sensor data collected from multiple hikers' smartphones. We extract cyclic gait-based features from walking motion data to train machine learning models, and multiple hikers' results are then aggregated for robust classification. We evaluate our system with two real-world datasets. First, we collected data from 14 climbers for a mountain trail which includes 13 risky segments. The average accuracy of individuals is approximately 80%, but after clustering the results, our system can accurately identify all the risky segments. We then collected an additional dataset from five climbers in two different mountain trails, which have 10 risky segments in total. Our results show that the model trained in one trail can be used to accurately identify all the risky segments in the other trail, which documents the generalizability of our system.

CCS Concepts: •Human-centered computing →Ubiquitous and mobile devices; Empirical studies in ubiquitous and mobile computing; *Empirical studies in HCI*;

General Terms: Crowdsensing system, risky mountain trail detection

Additional Key Words and Phrases: walking pattern analysis, one-class support vector machine, wavelet analysis

ACM Reference format:

Keunseo Kim, Hengameh Zabihi, Heeyoung Kim, and Uichin Lee. 2017. TrailSense: A Crowdsensing System for Detecting Risky Mountain Trail Segments with Walking Pattern Analysis. *PACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 65 (September 2017), 31 pages.

DOI: <http://doi.org/10.1145/3131893>

65

1 INTRODUCTION

Mountain climbing is a popular outdoor leisure activity. In the United States, for example, the number of mountain climbers has reached 4.5 million and has grown 16 percent over three years(2014–2016) [43]. However, climbing can sometimes lead to serious accidents. According to the American Alpine Club, there were 150 climbing accidents in the United States, excluding minor accidents, and the number of casualties was 120 in 2013. They found that the major causes of accidents were falling and slipping over rocks and ice. Similarly, according to the Korea Ministry of Public Safety and Security, the number of mountaineering accidents increased from 3,088 in 2010 to 7442 in 2014, an increase of 140% over five years [42]. It was also reported that the main cause of mountain accidents is a fall or loss of footing, which is very similar to the results of the American Alpine Club

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2474-9567/2017/9-ART65 \$15.00

DOI: <http://doi.org/10.1145/3131893>

Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Vol. 1, No. 3, Article 65. Publication date: September 2017.

survey. This clearly documents that the information about risky trail surfaces is critical to prevent major climbing accidents.

Thus far, manual inspection is commonly used to collect surface information. However, it is known that manual inspection is a very slow, costly process. For example, starting in 2007, Korea's National Forest Service has spent about 240 million USD over a 10-year period to collect mountain trail information and trail information services [12]. Furthermore, it is very difficult to thoroughly examine every possible route and to update new routes in a timely fashion—as time passes, existing trails disappear, and new trails emerge. For these reasons, only the surface condition information of highly popular trails is currently maintained; this means that we do not have any information about the most of the trails.

In this study, we proposed a mobile crowdsensing system that collects sensor data from the crowd and uses machine learning algorithms to automatically label whether trail segments are risky. This method can be easily embedded into a mountain trekking app that many hikers have already been using, and it requires minimal user involvements—simply turning-on the trekking app is sufficient. Our crowdsensing approach can reduce time and cost, increase trail coverage, and maintain up-to-date trail information.

Our work was inspired by recent urban road surface monitoring studies where motion sensors were used for detecting defects on pavement [11]. Given that most mountain trails are not accessible to vehicles, we consider humans as vehicles for trail surface monitoring. Our intuition is that climbers gait patterns would be significantly different when climbers pass through risky trail segments. Another closely related domain is the walkability studies of urban roads in which gait patterns are analyzed to infer typical urban surface conditions such as the presence of staircases and grass fields [60–62]. To the best of our knowledge, none of the prior studies examined mountain trails whose characteristics are very different from urban roads (e.g., high level of curviness and diverse types of obstacles). Our work is the first of its kind that analyzes walking patterns over mountain trails and proposes to infer risky trail segments by leveraging crowdsensing.

Our system analyzes gait patterns using smartphone sensor data. In particular, we analyze the cyclic nature of gait patterns to train machine learning models to infer whether a user passes through a risky trail segment. Wavelet-based features were extracted from the gait cycle. Specifically, we extracted the features of the gait from accelerometer signal data using a wavelet function called Daubechies 2. Next, the one-class support vector machine (SVM) was trained for detecting abnormal gait patterns. We addressed heterogeneity of walking behaviors by introducing an individualized learning model. We then used a window method to handle gait-level classification errors and a clustering method to aggregate the results from multiple users.

The main contributions of this work can be summarized as follows: (1) we tackled a novel problem by automatically inferring the surface conditions of mountain trails; (2) we proposed a mobile crowdsensing approach of building a crowdsourced safety map that can maintain up-to-date information at a much lower cost than existing methods; (3) we proposed a novel approach for monitoring trail surfaces by leveraging unique characteristics of walking behaviors (i.e., cyclic gait features for machine learning); and (4) we validated our approaches with two real-world datasets from different mountain trails ($n=14$, $n=5$) and released these datasets to the research community (<https://zenodo.org/record/826818#.WWd4aLQmiEE>).

2 RELATED WORK

We first provide an overview of mobile crowdsensing (MCS), a key concept of our mountain trail surface monitoring system. In addition, we surveyed the prior studies on road surface monitoring and walking pattern classification.

2.1 Mobile crowdsensing

Our work belongs to a body of recent research on mobile crowdsensing (MCS) [13], a distributed sensing paradigm where a large group of individuals collectively share data collected from their mobile devices in order to extract

information of common interest (e.g., identifying true colors of products with collective intelligence [24]). It refers to an extensive range of community sensing approaches, from participatory sensing [3] to opportunistic sensing [27]. Participatory sensing requires individuals to be involved in contributing sensor data (e.g., taking a photo/video), while opportunistic sensing requires minimal user involvement and collects sensor data autonomously (e.g., continuous sampling of user location data without any action by the user). In particular, MCS enables us to collect large-scale spatio-temporal data at low costs by leveraging existing infrastructure and previously deployed mobile devices in the fields [14].

A broad range of applications have adopted the MCS paradigm, from environmental and social applications [9, 38] to urban surveillance [30, 46] and infrastructure applications [11, 32].

In environment monitoring, Mun et al. [38] proposed an MCS called *PEIR*, the Personal Environmental Impact Report to collect the location data of individuals and to estimate carbon impact, sensitive site impact, smog exposure and fast food exposure. Eisenman et al. [9] proposed an MCS called *BikeNet* for cyclist experience mapping which uses sensors embedded on a cyclist's bicycle to collect sensor data about the cyclist's rides, including an embedded carbon dioxide (CO₂) meter to measure air pollution and shares air pollution measurements through a web-based portal. Citizen science work may require a group of people to visit the sites of interest and collaboratively collect images/videos with their mobile devices for environmental monitoring [23].

In urban surveillance and infrastructure applications, MCS can be applied for road safety monitoring, traffic forecasting, public transportation system design, and road condition monitoring as well. Citizen engagement with mobile sensing can significantly extend the scope of urban surveillance and infrastructure monitoring. For urban surveillance, MobEyes [29, 30] considers vehicles as mobile sensing platforms to collect various sensor data, including video data from the crowd (also known as vehicular sensor networks). Mobile Roadwatch [46] allows the citizen to report the captured videos of risky driving behaviors with mobile devices. For traffic prediction, Liu et al. [32] used a combination of the data from GPS embedded taxis and smart card records from buses. Similarly, the Real Time Rome project [4] collected data from buses, taxis and mobile phones to report real-time urban dynamics. For understanding the spatio-temporal dynamics of public transportation (e.g., examining the effects of weather on transit demand), Morency et al. [37] investigated bus boarding records in Canada. Zhou et al. [68] estimated the bus arrival time by collecting data about the bus passengers' surrounding environmental contexts. In addition, MCS has been adopted in road surface monitoring. Erikson et al. [11] collected the data from accelerometers attached to taxis to detect the location of potholes by clustering GPS data. Similarly, Xue et al. [64] collected accelerometer data via a driver's smartphone and proposed novel approaches to estimate the depth and length of potholes by aggregating individual reports.

2.2 Road surface monitoring

Public road condition monitoring is an important issue for municipalities. They try to monitor road conditions regularly to detect road surface defects, such as potholes, and repair them promptly. To achieve this objective, several automatic systems are proposed. Automatic systems can be categorized into three approaches: three dimensional (3D) reconstruction, vision analysis and vibration analysis.

Three-dimensional reconstruction approaches include 3D laser scanner methods [5, 31] and Kinect sensor methods [21, 35]. The 3D laser scanner can capture the 3D point cloud of objects. Chang et al. [5] used 3D laser scanners to detect distressed features of road pavement through a grid-based approach. However, this approach requires expensive equipment and 3D laser scanner and demands a high computational cost. Joubert et al. [21] used a low-cost Kinect sensor and a high-speed universal serial bus (USB) camera to detect the potholes. Although it is low cost compared to the 3D laser scanner, its accuracy is lower than that of laser scanners, and further research is needed to lower the error rates.

Vision analyses make use of two-dimensional (2D) images [25] and video data [18, 20]. Koch and Brilakis [25] collected the 2D image data using a robot-vehicle with a rear camera. Lokeshwor et al. [18] collected the video

data to overcome the limitation of the single frame in the 2D image method. In their method, a collected video was processed with a Critical Distress Detection, Measurement and Classification (CDDMC) algorithm which consists of five steps: image enhancement, image segmentation, visual properties extraction, detection and classification by decision logic, and quantification.

Yu and Yu [66] proposed a vibration approach for the preliminary evaluation of pavement conditions. The distress feature of pavement affects impacting forces on vehicles. In their approach, an accelerometer was used to measure the impacting forces and a PC oscilloscope was used to analyze the data. They showed the relationship between the standard deviation of the vertical acceleration of a vehicle and the pavement condition. Erikson et al. [11] proposed a mobile system called Pothole Patrol (P^2) which detected and reported the surface conditions on road. P^2 collected the accelerometer data, which were separated into different classes of defects by a simple rule-based classification approach. Defects were then reported to a central server for robust inference with location clustering of multiple reports.

In addition, several machine learning algorithms have been applied in vibration approaches. Sahin and Uzmay [65] applied a neural network to analyze the variation of vertical vibrations of vehicles. Tai et al. [56] evaluated machine learning methods with SVM for supervised learning and ranking systems for unsupervised learning for classification. Cong et al. [8] used wavelet decomposition and one-class SVM for pothole detection with vehicles.

Instead of designated accelerometer devices, there were several prior studies using smartphones to collect the data. For example, Mednis et al. [33] proposed a mobile sensing system for road surface detection using Android OS-based smartphones. They collected accelerometer data from smartphones in vehicles and applied z-thresh, z-diff, STDEV(Z), G-zero algorithms as in the prior work [11]. Xue et al. [64] proposed Perceiving Pothole Profiles (P^3) which is similar to P^2 , but it collects accelerometer data from the driver's own smartphone. P^3 can infer the depth and length of a pothole when the vehicle drives on the road. Furthermore, P^3 shows more accurate performance by aggregating individual results.

Our proposed system also uses an accelerometer for detection and aggregates multiple individual classification results for robust classification. Prior studies cannot be directly used because vehicles cannot enter the mountainous environments. This means that we cannot use any algorithms that basically detect anomalies of vibration signals generated from an accelerometer attached to the body of a vehicle. Instead, in our work, we collect motion sensor data from mountain climbers and analyze human walking data with cyclic gait patterns to infer whether a person walked over a risky trail segment.

2.3 Walking pattern analysis

Many studies analyzed walking patterns using motion sensors to enable various applications such as detecting the surface conditions that the user is walking on or the effects of inclination on walking patterns. Another application domain is healthcare where walking patterns are analyzed to detect unhealthy walking patterns or risky events such as falls.

Several studies addressed surface detection based on the changes in walking patterns, but these approaches only considered very simplistic urban roads. For example, Wang et al. [60–62] applied different feature extraction and machine learning methods to classify walking on different surfaces including flat, up-stairs and down-stairs based on accelerometer data. Wang et al. [63] also proposed a method to segment accelerometer data into strides to classify flat, stairs or sloped surface. Similarly, Lau et al. [28] used stride segmentation and a combination of accelerometer and gyroscope data for classification. Some of these studies also leveraged human gait phases in identifying the surface. Wang [59] classified different walking patterns using a decision tree based on different gait phase information like stance, push-off, swing, and heel-strike to classify walking on flat and stair surfaces.

There are also other studies that considered surface unevenness or a change in surface condition based on a combination of accelerometer and gyroscope data. Schuldhaus et al. [53] proposed several machine learning

techniques to classify surface smoothness (grass and paved) and inclination separately. Jain et al. [19] proposed a simple threshold-based method to identify walking on ramps and stepping off the curbs events for pedestrian safety with a shoe-mounted sensor. These events have very specific motion signatures which make their identification possible by simple thresholding methods. As shown in our results, acceleration patterns over mountain surfaces are very complex, and we cannot directly apply simple machine learning approaches to motion features.

There are many studies in the healthcare domain that aim to identify irregular events in walking patterns based on inertial sensors [10, 17]. Most of these studies identify a single event like a fall or turn which has an obvious pattern like body rotation or lying posture after a fall event. In mountain climbing several events and thus irregular patterns in signals can happen owing to surface condition. However, the problem is that these patterns are not uncommon or less predictable and are largely dependent on individual variation and surface unevenness. Other studies analyze irregular gait patterns caused by some diseases like Parkinson or disabilities after stroke or dropped foot [1, 34]. According to existing medical research, gait patterns remain constant. However, a change can gradually appear because of aging or the progress of a disease. Unlike these studies, in mountain climbing irregularities often happen in strides on risky trail segments and their motion signatures differ significantly. Overall, the context setting of these studies are different from ours in that we used mountain climbers as sensors for detecting mountain surface conditions.

Many studies have focused on classifying different human activities including walking, running, cycling, climbing stairs, etc. They usually used fixed body-worn sensors or smartphone placements but some of them also used unconstrained smartphones for data collection. Kwapisz et al. [26] compared different machine learning techniques to identify simple activities including walking, jogging, climbing stairs, sitting and standing from accelerometer data placed in a user's pocket. Kahn et al. [22] proposed a two-level classification method where in the first step they identified the dynamicity of the activity using neural networks and in the second step they classified the activity using autoregressive modeling. Similarly, He and Li [16] used a hierarchical classification method to identify multiple activities and postures using accelerometer data from smartphones that were placed in the chest pocket. Although our study can be considered as human activity recognition, but the nature of movements is very different. The daily activities that are referred to these studies have regular behaviors and repetitive patterns from which many features can be extracted for classification. However, mountain climbing over risky trail segments may not involve patterned movements. As shown earlier, people often showed heterogeneous walking behaviors over risky trail segments. Note that our findings about risky trail detection mechanisms can be applicable to other application scenarios such as indoor localization [39, 40] and risky driving behavior analysis [6, 46, 47].

3 TRAIL SURFACE CLASSIFICATION SYSTEMS: AN OVERVIEW

We describe the architecture of our system and briefly explain our proposed mountain trail surface classification algorithms. Our key idea is to collect accelerometer data from mountain climbers and use the dataset to collectively classify the condition of trail surfaces. Figure 1 shows the structure of the system. The system consists of a central server and client tracking apps that track climbers' motion data.

Users can begin data contribution by launching our client app when they start mountain climbing. The user's tracking app collects the raw data from the three-axis accelerometer and the GPS readings. Our algorithm exploits cyclic patterns of walking by first segmenting the motion data into strides, and each stride will be classified whether it belongs to the risky trail segments. Each stride is classified as a normal or abnormal (or risky) stride by using a machine learning method. We extract the features on time and frequency domains with wavelet coefficients. The algorithm 'learns' the normal stride patterns of the user. With this trained model, the algorithm classifies the stride data of the user. For robust classification, a window of strides is considered. That is, the window will be classified as risky with a simple voting mechanism; for example, the ratio is above a predefined threshold. Given that a user's stride length can be estimated as in prior studies [7], we can also infer the length

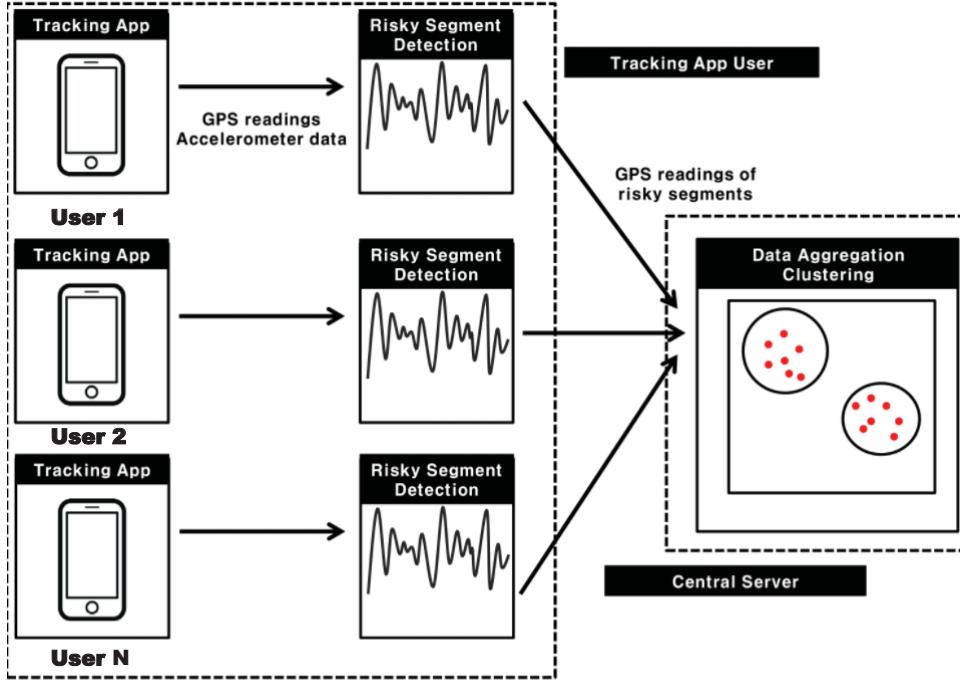


Fig. 1. Trail surface classification system overview

of the risky trail segments. The classification results will be then tagged with the GPS information, and will be reported to the server for aggregation. After collecting the results, the central server runs a spatial clustering algorithm to discover risky trail segments, and the discovered clusters are used to identify where the risky trail segments are located and what are their lengths are.

One of the major benefits of our approach is that the training efforts are not extensive. Unlike binary-class models, our work is based on single-class models such as single-class SVM. This means that in order for us to train the models we only need normal strides. Actually, it is much easier and safe to collect the training data of normal strides than that of abnormal strides over the risky segments. For example, users can easily find a 100 m of strides over the non-risky trails. Furthermore, according to existing literature, gait patterns remain constant and their change gradually appears such as due to aging [7]. Thus, users do not need to repeatedly collect the dataset to retrain the models.

4 DATA COLLECTION

For our experiments, we used the Nexus 5 smartphones with Android OS version 5.00 (lollipop) and collected the three-axis accelerometer data and GPS coordinates. We focus on accelerometers since they are the most widely available sensors on smartphones and consume the least amount of energy. To collect data, we developed a custom sensor-data logging program that uses the Android sensor API and recorded data with the sample rate set to FASTEST, which is approximately 50 Hz. We also recorded video of how our participants walk along the mountain trails for use in analyzing their walking behavior and labeling the mountain trails for classification.

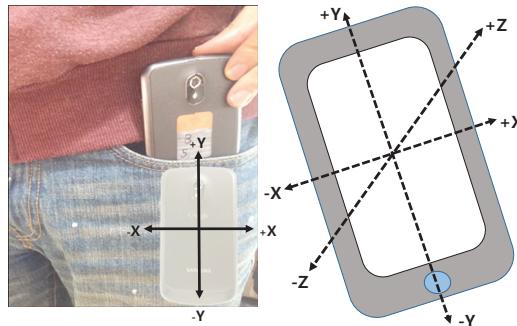


Fig. 2. Position of a smartphone, and its orientation (a smartphone's screen faced toward the body)

For ease of explanation, we will use the notation in Table 1 for three-axes of accelerometer raw data, and processed data throughout the paper.

Table 1. Notation

Notation	Explanation
A_X	Original X axis accelerometer signal
A_Y	Original Y axis accelerometer signal
A_Z	Original Z axis accelerometer signal
A'_Y	Low-pass filtered Y axis accelerometer signal
P_Y	Detected peaks by applying peak detection algorithm on A'_Y
S_X	Segmented X axis stride data by P_Y
S_Y	Segmented Y axis stride data by P_Y
S_Z	Segmented Z axis stride data by P_Y
\mathbf{X}	Features extracted by S_X, S_Y, S_Z
\mathbf{Y}	Classification result of stride by \mathbf{X}
\mathbf{Z}	Classification result of window by \mathbf{Y}

4.1 Position of smartphone

Motion sensor data vary according to the sensor position. In prior activity recognition studies, researchers considered various positions such as chest, waist, foot, and pants (pockets) [15, 26, 69]. In this study, we consider the front pockets on pants since they are natural places for holding smartphones during climbing. For ease of collection, all phone screens faced toward the body with the camera on top. In this orientation, X-axis means the left-right axis, Y-axis means the vertical axis and Z-axis means the inward-outward axis (see Figure 2).

4.2 Collected data description

We collected sensor data from 14 participants (7 males and 7 females) whose ages ranged from 22 to 32 years (Mean: 27.4, Std: 2.17). The experiment site was located in Gyeryongsan National Park, Daejeon, South Korea. We chose five different zones; each was chosen to reflect various mountain trails in real mountain climbing scenarios. A trail is a hiking path in the mountain that can be divided into shorter sub-trails which we refer to

as trail segments in this paper. We define that a trail segment consists of one or multiple strides. By stride we mean a single human gait cycle. A trail then may consist of multiple segments, each of which can be risky or not. Risky (R) and non-risky (NR) trail segments in each zone may alternate and every zone starts with a non-risky trail segment. Zone 1 consists of a single 145m non-risky trail segment. In Zone 2, there are nine trail segments with 22.1 (NR), 6.3 (R), 23.1 (NR), 9.0 (R), 7.5 (NR), 6.9 (R), 12.8 (NR), 3.4 (R) and 17.2 (NR) meters long. The total length is 102m. In Zone 3, there are 13 segments with 18.9 (NR), 5.5 (R), 4.5 (NR), 17.2 (R), 4.4 (NR), 12.7 (R), 3.8 (NR), 3.8 (R), 9.4 (NR), 14.3 (R), 4.8 (NR), 10.5 (R) and 2.2 (NR) meters long and the total length is 122.6m. The Zone 4 consists of three segments with 7.9 (NR), 8.4 (R) and 31.3 (NR) meters long and a total of 47.3m. The last zone includes five segments with 3.2 (NR), 8.2 (R), 7.6 (NR), 4.4 (R) and 1.1 (NR) meters long and the total length is 24.8m. We estimated the length by average number of strides from all participants in a segment multiplied by the average stride length, 1.4m. In the appendix, we summarize the length and the number of strides for all non-risky and risky trail segments in each zone in detail.

4.3 Definition of risky mountain trail surface

In our study, we define a trail segment as risky when a climber's normal walking patterns are disrupted owing to rough surface conditions. Mountain accidents are mainly caused by stony surface conditions, and we mainly considered stone height; if the height of the stone is above knee height, we consider a trail segment is risky. Here, knee height is defined as the distance from the sole of a foot to the most anterior surface of the femoral condyles of the thigh when the ankle and knee are bent to a 90 degree angle [57]. The average knee height as a percentage of body height is 30.5% [57]. According to the national statistics in Korea (2015), the average height of males is 173.5 cm and that of females is 160.4 cm [41]. Thus, the average knee height in Korea is approximately 50 cm. Another important factor is slope, because as the slope goes higher the risk potential increases. Given that slope can be easily calculated with GPS traces, we mainly focus on rocky mountain trails, which are particularly common in most mountains in Korea. We labeled the mountain trails in the collected data based on this definition. Two of the authors collaboratively labeled the data by manually reviewing the recorded video.

5 ANALYSIS OF WALKING PATTERNS OVER MOUNTAIN TRAILS

Human walking has a cyclic pattern consisting of different gait phases. During climbing mountains, these patterns can be affected owing to surface variation. Based on our investigation of sensor data and videos we found the three key characteristics of walking on mountain trails. The first characteristic is that there were individual variations in motion sensor data. Walking patterns vary among individuals because of their walking and climbing styles, and these variations are reflected in accelerometer data as well. The second characteristic is that risky and non-risky trail segments had significantly different acceleration patterns. In risky trail segments, walking patterns are disrupted owing to surface roughness. The third characteristic is that there were uphill and downhill variations. When comparing walking patterns in uphill and downhill, we found significant variations owing to human kinematics of walking. In the following, we systematically analyzed such characteristics and illustrated our system design decisions.

5.1 Human gait cycle

Walking is a cyclic activity consisting of several events, in that, each leg goes through the following gait phases during a cycle: stance, heel off, swing, and heel strike [45]. Given that the reference leg is on the left, then stance is the period when the left foot's sole is completely in contact with the ground. Heel off is the moment when the left foot toes are about to leave the ground. From the moment the left foot loses contact with the surface until its heel touches the ground is called the swing phase. During the swing phase of the left leg, the opposite leg is going through its heel strike, stance, and heel off phases. Finally the heel strike is the moment the left foot contacts the ground again. A gait cycle or a stride starts from one of these events until the next occurrence of the same event.

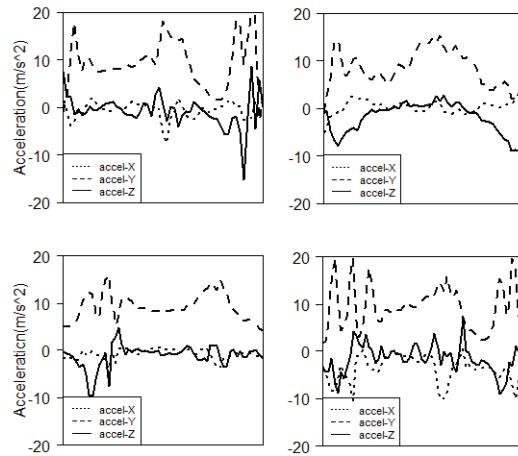


Fig. 3. Variations in single stride patterns among 4 different individuals on the flat surface.

We considered two consecutive heel strikes as the start and end of a stride because when the heel hits the ground, it exerts a transient intense force, which results in a peak in the acceleration. The impact of this spike is more obvious in the vertical direction. Based on our decision for phone placement, Y-axis acceleration is in a high alignment with vertical acceleration. In Section 6, we applied the peak detection algorithm to the Y-axis acceleration to separate the strides for the same reason.

5.2 Individual variations in walking

People have their own walking and climbing styles, which result in significant variations of acceleration signals. Figure 3 illustrates three accelerometer axes for a single stride pattern of four different participants for normal walking in a straight line on a flat surface. By flat surface, we mean a dirt trail segment in a mountain whose slope is close to flat. The results showed that each participant's acceleration data were considerably different, and it is difficult to choose a common stride pattern which matches all individuals' patterns.

Participants may climb the same trail segment differently, e.g., different postures or paths. If a segment is wide in width, a participant may take different paths which they find more suitable for them. In addition, they may use different postures which they find more comfortable, e.g., bending vs. sitting, to pass a risky segment.

In Figure 4, we plotted the mean values of acceleration data in each axis for all our participants. We observed that each box plot resides in a different range such that the median and inter-quartile range were different for every participant. This confirmed that there were significant individual variations. Individual variations make it difficult to train a generalized model for classification. In addition, normal walking patterns persistent in most trails, except risky trail segments, which will be shown later. This shows that it would be beneficial to consider a personalized one-class classification model.

5.3 Behavior analysis in risky and non-risky trail segments

Another characteristic is that there are signal variations owing to surface conditions (i.e., risky vs. non-risky trail segments). Figure 5 shows the difference between risky and non-risky surfaces in mountain trails. Body posture is less balanced in the risky surface (Figure 5 (a)) than that in the non-risky surface (Figure 5 (b)). In the risky trail segments, our participants tended to change their walking patterns in various ways to pass by a big rock or to

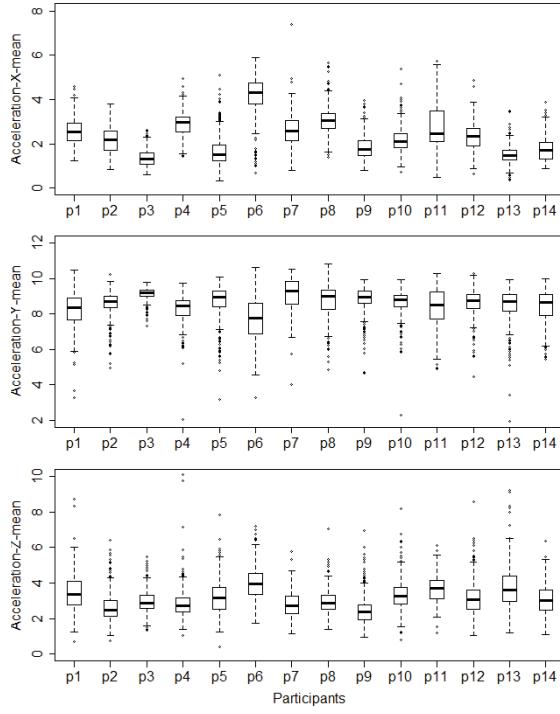


Fig. 4. Variations in individuals' accelerometer data based on stride features. For multiple strides of each participant's accelerometer mean distribution is plotted, separated by the different axis.

maintain their balance. When comparing walking patterns between non-risky and risky trail segments, we found significant differences in acceleration signals owing to posture changes in uphill (bending knees for climbing, pausing, and using hands for support which maintains the stability of the body) and downhill (using hand support, turning to the side and bending, pausing, sitting, jumping, and frequent short steps). These behaviors will induce significant variations in their acceleration patterns (see Figure 6). In contrast, in the non-risky trail segments, walking patterns are similar to those on flat surfaces as confirmed by our video analysis. The signal shape of a single stride is similar to that in the flat surface; and accelerometer values were approximately within the same range (see Figure 7). In the non-risky trail segments, we observed a few irregularities, but they were expected because along any non-risky segment, there were some uneven parts (e.g., small gravels, tree roots, branches). The key difference between flat and non-risky trail segments was that the speed was slower owing to slope.

Our results showed that the accelerometer signals are disrupted significantly in risky trail segments. The diversity and unpredictability of behaviors and consequently data patterns in risky trail segments, make it more challenging to model behaviors in risky trail segments using accelerometer data. In our system design, we considered both single- and binary-class classification models and examined whether single-class classification has some advantages against binary-class classification models.

5.4 Variations in uphill and downhill walking patterns

The last characteristic is that there is a significant difference between uphill and downhill walking based on our video analysis. We found that in downhill walking, participants tended to walk fast, and the patterns were less



(a)

(b)

Fig. 5. Risky trail segments (a) vs. non-risky segments (b). Body posture is less balanced on the risky segments.

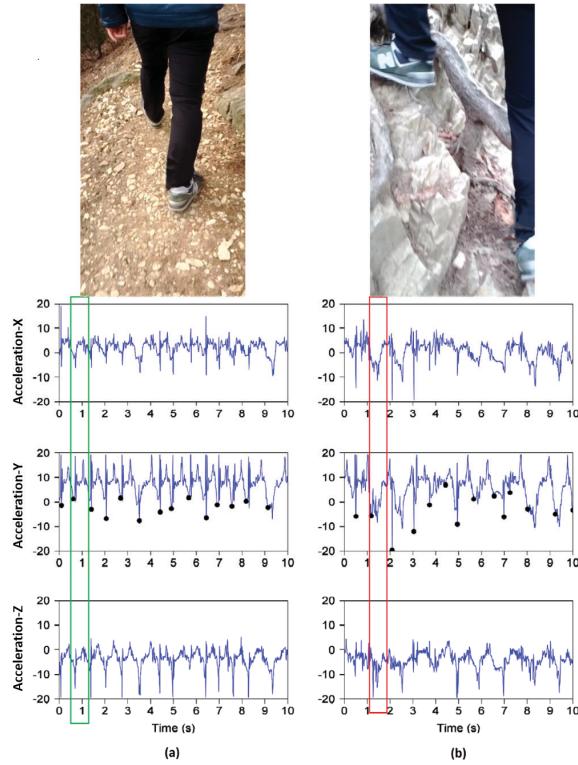


Fig. 6. Accelerometer visualization in non-risky segments (a) and risky segments (b). The moments of touching ground by the left foot in the photos are marked with green and red rectangles in the risky and non-risky segment plots respectively. Small black circles on the accelerometer Y-axis separate the strides.

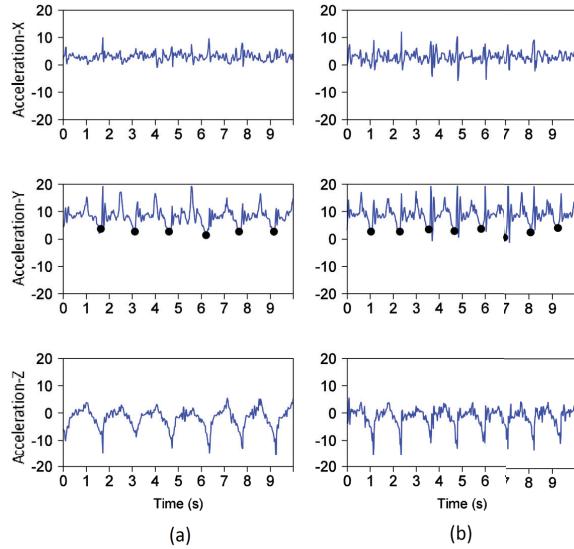


Fig. 7. Walking pattern on non-risky mountain segments (a). Walking pattern on flat surfaces (b) for the same person. Small black circles on the accelerometer Y-axis separate the strides.

stable than those in uphill walking. In contrast, in uphill climbing, people feel more stable and walk without haste. In Figure 8, we compare accelerometer patterns of a person in the uphill segment (a) and downhill segment (b) in the same non-risky segment for 10 seconds. Small black circles on the accelerometer Y-axis separate the strides. Generally speaking, the number of strides per second in downhill walking is higher when compared to uphill walking because the slope is negative, the stride length is shorter, and the speed is faster. The negative inclination in downhill also has other effects on walking pattern. We found that surface inclination caused participants to strike the ground more intensely to maintain balance and speed. Thus, even a normal stride pattern on non-risky trail segments in downhill walking differs from uphill normal stride patterns. Furthermore, even small obstacles such as tree roots and gravels in downhill walking can cause sudden slow-down or even pause. These behavioral characteristics in downhill walking cause a higher level of irregularities in acceleration patterns, although participants climb nonrisky trail segments.

By using only uphill data, we can avoid more sophisticated signal pre-processing and processing to remove the downhill effect. Uphill data will be sufficient if we can cover all the trails. Most mountain trails are bi-directional, which means that hikers go through the trails in both directions. In this case, we can easily collect uphill data of the entire trail. Therefore, with crowd-sensing the surface detection data, even if a person passes a trail for downhill only and his uphill data is not available, uphill data will be provided by other people climbing the same trail. This highlights that our method is able to cover all the trails when enough people climb them. Based on this observation, we chose to use only uphill data for model building and classification in our study.

6 MOUNTAIN SURFACE CLASSIFICATION

6.1 Data segmentation (Stride Segmentation)

Collected sensor data are segmented to strides for the purpose of modelling normal strides. Here, data segmentation means that collected sensor data are segmented to strides. There are many prior studies on the stride detection method using motion sensor data. In this study, we use the “peak detection method” to detect the

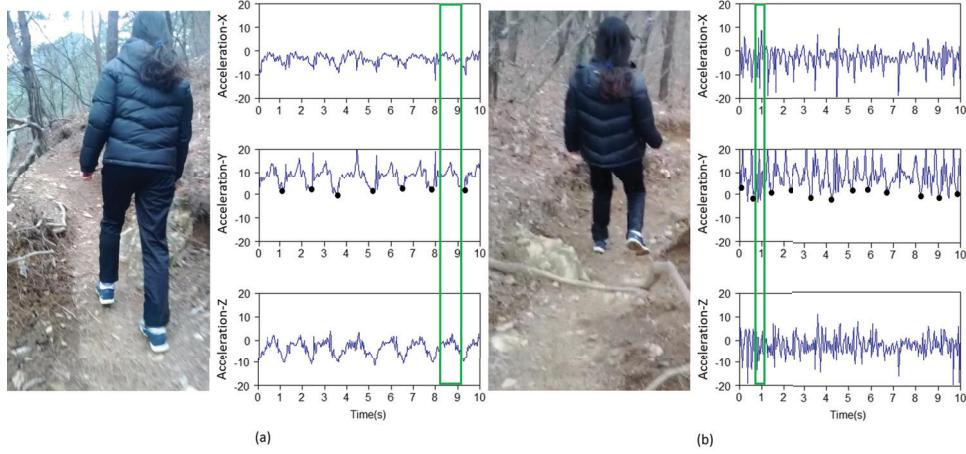


Fig. 8. Uphill (a) and downhill (b) variation in different accelerometer axes for the same person and the same non-risky trail segments. The moment of screenshots is marked on the plots. Small black circles on the accelerometer Y-axis separate the strides.

stride and segment the data. We use the peak detection method because it is the most robust approach when noise occurs on the signal data [2]. Prior research showed that the peak detection method demonstrated the lowest error rate when data were collected on an unconstrained smartphone [2]. Moreover, since a signal on a mountain trail has greater frequency of noise than on a flat road, the peak detection method is the correct choice for this experiment. In addition, the stride detection method needs to be applied properly in an abnormal stride data which has more noise. Thus, the peak detection method is the correct choice for this experiment. For peak detection, the acceleration data is segmented with a low pass filter as shown by Randell et al. [49], and we use the score function to detect peaks as in Palshikar et al. [44].

The key idea of the peak detection algorithm is to use the score function, which represents the extent to which one point differs from the other points. We denote T as a time series of N sensor readings and i th readings in T as a x_i . The window length parameter k is also needed to calculate the score function. The window length parameter k determines the frequency of the peaks to be detected. The score function is calculated by the mean of the largest difference value between each point and any other points in the left half window (i.e., $\max\{x_i - x_{i-1}, \dots, x_i - x_{i-k}\}$) and those in the right half window (i.e., $\max\{x_i - x_{i+1}, \dots, x_i - x_{i+k}\}$) as shown in Equation 1 below.

$$S(k, i, x_i, T) = \frac{\max\{x_i - x_{i-1}, \dots, x_i - x_{i-k}\} + \max\{x_i - x_{i+1}, \dots, x_i - x_{i+k}\}}{2} \quad (1)$$

Algorithm 1 is the pseudocode of our peak detection algorithm. After calculating the value of the score function for each point, the algorithm calculates the mean (m) and standard deviation (s) of the score function. The parameter h adjusts the threshold of the score function value that can be the candidate of peak. If the score function value of each data is larger than the threshold value ($m + h * s$), the corresponding data are stored as a candidate for the peak. Therefore, in order to have only one peak in one window, remove the point with the smaller value of the peak candidates in the same window .

Figure 9 shows an example of peak detection results. We labeled stride ground truth after reviewing video data (Green). As shown in this figure, our peak detection algorithm works well in regular stride patterns.

ALGORITHM 1: Peak detection algorithm

```

 $T = [x_1, x_2, \dots, x_N]$ 
k : window length, h : threshold parameter
Peaks = [ ], initialize peaks
score = [ ], initialize score function value
for  $i = 1$  to  $N$  do
     $\lfloor$  score[ $i$ ] =  $S(k, i, x_i, T)$ , score function value for  $i$ -th point
m = mean of score[ $i$ ], s = standard deviation of score[ $i$ ]
for  $i = 1$  to  $N$  do
     $\lfloor$  if score[ $i$ ] >  $m + h * s$  then
        Peaks = Peaks  $\cup$   $x_i$  ; Add  $x_i$  in Peaks if its score function value is above the threshold
for For all neighboring points in Peaks  $x_i, x_j$  do
     $\lfloor$  if  $|j - i| \leq k$  then
        Remove the smaller value of  $x_i, x_j$  in Peaks
return Peaks

```

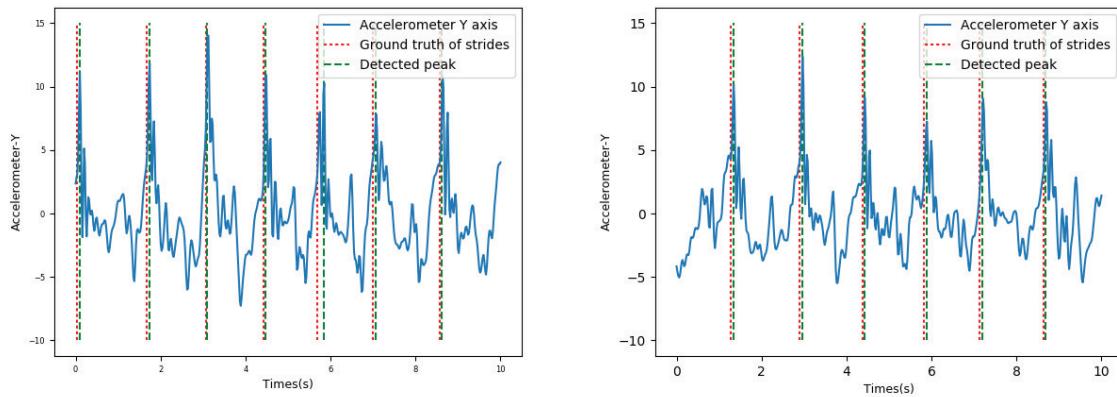


Fig. 9. Peak detection results

Therefore, we can segment the data into strides by using the peaks P_Y that were detected as shown in Figure 9. If we define m^{th} points of P_Y as y'_{i_m} in A_Y , data from i_m to $i_{(m+1)} - 1$ in A_Y are segmented as m^{th} stride data. Thus, m^{th} stride data of the accelerometer's X-axis S_{X_m} are $[x_{i_m}, x_{i_m+1}, \dots, x_{i_{(m+1)}-1}]$, Y axis S_{Y_m} are $[y_{i_m}, y_{i_m+1}, \dots, y_{i_{(m+1)}-1}]$ and Z-axis S_{Z_m} are $[z_{i_m}, z_{i_m+1}, \dots, z_{i_{(m+1)}-1}]$.

6.2 Feature extraction

The features were extracted to classify the normal stride pattern in the segmented data S_x, S_y , and S_z . The features were categorized into two groups: time domain features and time-frequency features. As time domain features, we had the absolute mean, standard deviation, and maximum values.

In the time-frequency domain, a wavelet function is used to extract the features. Wavelet analysis decomposes a continuous signal into orthogonal wavelets, and the signal can be represented with a weighted sum of wavelets—these weights are called wavelet coefficients. Wavelet analysis therefore can be applied to the non-stationary

time series data as well. Discrete wavelet decomposition is defined as follows.

$$x(n) = \sum_{j=1}^J \sum_{k \in \mathbb{Z}} d_j(k) \psi(n - 2^j k) + \sum_{k \in \mathbb{Z}} a_J(k) \phi(n - 2^j k) \quad (2)$$

In Equation (2), $j \in \mathbb{Z}$ represents the resolution, J is the depth of level, ψ is wavelet function, and ϕ is the scaling function [54]. In addition, wavelet coefficients $d_j(k)$ represent details of the original signal at different resolutions, and scale coefficients a_J are approximation of the original signal. Discrete wavelet analysis is applied on m^{th} stride data, $S_{X_m}, S_{Y_m}, S_{Z_m}$. The Daubechies 2 wavelet function was used and the data were decomposed at the depth of level 4 [54]. Thus, four wavelet coefficient sets at different level and scale coefficient sets are extracted from each of the three axes. A total of 15 coefficient sets therefore were extracted. Additionally, Preece [48] compares the performance of features from wavelet coefficient sets when accelerometer sensor data were used in activity recognition. The most superior features are the absolute mean and the standard deviation of each coefficients set. Therefore, we used the absolute mean and standard deviation of 15 coefficients sets, which results in a total of 30 features in the time-frequency domain. We notate X_j as a feature of the j^{th} stride, and this vector includes 9 features of time domain and 30 features of time-frequency domain.

6.3 Stride classification

We classify a stride as risky or nonrisky using a one-class model. We use one-class classification to classify whether a stride is normal or abnormal, because walking patterns of risky trail segments would be significantly different as stated in Section 5. Use of one-class classification is beneficial for bootstrapping—a new user does not have sensor data from risky trail segments initially, and collecting sufficient dataset for binary classification takes considerable time. In some cases, it may not be possible to collect risky segment data if a user wants to only visit non-risky trail segments. In our evaluation, we compare both one- and binary-class methods to check whether the one-class model works as well as the binary-class model.

To model the normal stride pattern using the extracted features, ground truth non-risky data were used, except for the test data. We use the one-class classification method by Schölkopf et al. [52]. A one-class SVM identifies the farthest hyperplane from the origin, which includes all normal data on one side. We can formulate a one-class SVM problem, where M represents the amount of training data used to model a normal stride pattern.

$$\max_{\mathbf{w}, \rho, \xi} \rho - \frac{1}{\nu M} \sum \xi_i \quad s.t. \quad (\mathbf{w} \cdot \Phi(X_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0 \quad (3)$$

Here, we used a kernel function Φ of a Gaussian kernel to model non-liner data. An algorithm trains the parameters \mathbf{w} and ρ by the given training data and Equation (3). After training, the algorithm classifies the j^{th} stride in test data as normal stride if the equation (4) is positive; otherwise, the results will be treated as negative (i.e., risky).

$$\begin{aligned} Y = f(X_j) &= 0 && \text{if } \operatorname{sgn}(\mathbf{w} \cdot \Phi(X_j) - \rho) \geq 0 \\ &= 1 && \text{if } \operatorname{sgn}(\mathbf{w} \cdot \Phi(X_j) - \rho) \leq 0 \end{aligned} \quad (4)$$

6.4 Windowing for robust classification

As shown earlier, we can find the risky mountain segments if a series of abnormal strides were observed. However, there could be many errors—a normal stride may be classified as an abnormal stride. Our intuition is that in the risky mountain segments, it is more likely that we will observe abnormal strides more frequently than normal strides. A simple approach is to use a window of strides, and check whether the relative ratio of abnormal strides is above a certain threshold. The term window means a part of stride data that consists of multiple strides

for robust classification. Let us say that there are J strides in a test mountain trail: i.e., $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_J]$ represents stride classification results and $\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_K]$ represents window data in length n and then $\mathbf{Z}_k = [\mathbf{Y}_{nk+1}, \mathbf{Y}_{nk+2}, \dots, \mathbf{Y}_{n(k+1)}]$. Thus, we simply classify the window as a risky trail segment in a given mountain trail if the ratio of the abnormal strides is above a predefined threshold. In this study, we empirically set the threshold to 0.5. Equation (5) summarizes our window-based classification method. It is important to note that there is a tradeoff between classification resolution and accuracy. The window size should be large enough for robustness. Nonetheless, the length of a detectable risky trail segment depends on the window size: if it is too large, only long risky trail segments can be detected. The tradeoffs will be analyzed in our evaluation section.

$$\begin{aligned} \mathbf{Z}_k &= 0 && \text{if } \sum_{m=nk+1}^{n*(k+1)} Y_m \leq n/2 \\ &= 1 && \text{if } \sum_{m=nk+1}^{n*(k+1)} Y_m \geq n/2 \end{aligned} \quad (5)$$

6.5 Aggregating results from the crowd

Each user's windowed data (along with GPS coordinates) are submitted to a central server. It is important to reliably aggregate the classification results from multiple users. There are two key challenges in this process. The first challenge is the fact that GPS data collected by a smartphone have a 10 to 15 meter margin of errors. Second, there are few false alarms in individual classification. Fortunately, even with errors, it is likely that detected trail segments reported by the crowd will be located nearby and make a cluster. Furthermore, owing to the low possibility of false alarms in non-risky trail segments, false alarms are scattered.

Thus, we use well known density-based spatial clustering called the density based spatial clustering of applications with noise (DBSCAN) for aggregation. This algorithm performs spatial density based clustering by tuning two parameters, i.e., the minimum number of data points in one cluster and the optimal epsilon (eps) parameter, which regulates the expanding of the cluster. By applying the DBSCAN algorithm on detected risky trail segments, we were able to make a cluster of true positive alarms about risky trail segments and eliminate the false positive alarms.

7 EVALUATION

In this section, we evaluate the peak detection algorithm which we used for data segmentation (or stride detection). Next, we evaluate our classification algorithms to validate whether risky trail segments can be estimated by aggregating the monitoring results of users. The dataset in Section 4 was used for analysis. Using the video data and motion sensor data, our results are compared with the manually tagged labels from the video data. Each stride in a video is labeled as risky if it is located in the risky trail segment based on our definition in Section 4, and a window of strides is labeled as risky if more than half of the strides in that window are labeled as risky.

Table 2. Identification of true positive, false positive, true negative, false negative events

		Classification result	
		Non-risky	Risky
Ground-truth label	Non-risky	true negative	false positive
	Risky	false negative	true positive

We used the leave-one-out method for cross-validation. To be precise, we used one window as the test data and the rest as the training data. Then the algorithm classified the test data whether it is risky or not. To validate

the classification performance, we define four mutually exclusive events, i.e., true positive, true negative, false positive and false negative in Table 2. In addition, we used the following performance measures, i.e., accuracy, precision, and recall as follows.

- Accuracy : The ratio of the number of correctly classified windows to the total number of windows

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FN+FP}$$

- Precision : The ratio of the number of true positives to the sum of true positives and false positives

$$\text{Precision} = \frac{TP}{TP+FP}$$

- Recall : The ratio of the number of true positives to the sum of true positives and false negatives

$$\text{Recall} = \frac{TP}{TP+FN}$$

7.1 Evaluation of peak detection algorithm

We evaluated the performance of the peak detection algorithm by comparing the number of detected peaks with the number of strides based on our ground truth labels. In detail, we calculated the error based on the surface conditions including flat, nonrisky and risky. Even the flat zone is considered as a nonrisky surface, we separated the results for the flat zone from the rest of the nonrisky trail segments to show their similarity. Thus, for each participant, we calculated three error rates for each surface condition. The equation below shows how we calculated the error rate for an individual participant p and for a specific surface conditions.

$$\text{Error rate}_{p,s} = \frac{\sum_{\text{segments}} |\text{Total } \# \text{ of ground truth strides} - \text{Total } \# \text{ of detected peaks}|}{\sum_{\text{segments}} \text{Total } \# \text{ of ground truth strides}}$$

$p \in \{\text{participants}\}, s \in \{\text{flat, non-risky, risky}\}$

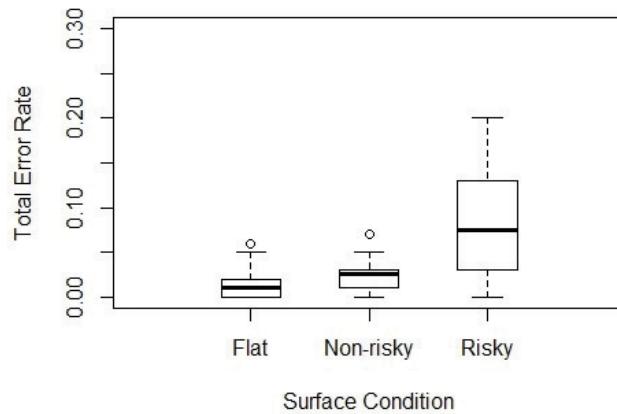


Fig. 10. Error of the peak detection algorithm on different surface conditions

Figure 10 summarizes total error of participants divided by the flat zone, non-risky trail segments, and risky trail segments. As we mentioned in Section 5.3, people's behavior in nonrisky trail segments is similar to that in

flat surface. Similarly, we expected that the peak detection algorithm would show comparable error rates for flat and non-risky trail segments. The average error rates for flat zone, non-risky trail segments, and risky trail segments are 1.6% ($sd = 1.8\%$), 2.4% ($sd = 1.9\%$), and 8.2% ($sd = 5.7\%$), respectively. As we expected, the algorithm performance for non-risky trail segments is slightly different from flat zone and the algorithm works close to perfect for these cases. High detection accuracy in non-risky trail segments is important in our approach, since our model works with classifying the normal strides. For a risky trail segment, the error rate is higher but it is still tolerable. Based on the final classification results, we can argue that our method works well despite of these errors.

Our manual examination of the detection results shows that errors are largely from the risky trail segments. In fact, when users pass through the risky trail segments, we found that our algorithm mostly overestimates the number of strides. We manually examined the recorded videos to understand why such results happen. We discovered that when users pass through the risky trail segments, they tended to make incomplete strides. Owing to the rocky surface, participants tried to search for a spot to place their foot to climb up, for example, by making a series of movements on one foot. In some cases, we noticed that their feet slipped along the rocky surface. Our ground truth labeling is conservative in that we consider a full gait cycle as a stride. For this reason, such incomplete strides were not counted as strides. However, we found that our peak detection algorithm was able to find most of these incomplete strides, which led to overestimation.

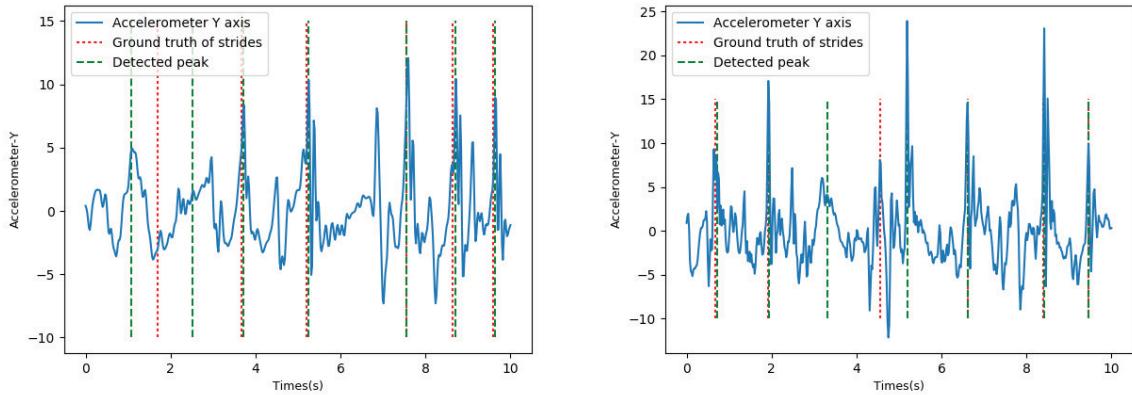


Fig. 11. Peak detection results over a risky trail segment

Figure 11 shows the peak detection results in a risky trail segment. While normal walking has cyclic patterns of acceleration signals, those in the risky trail segments had quite significant random noises. Despite this situation, our algorithm can fairly accurately detect the peaks that match with the ground truth. However, there were several false positives because our algorithm detected the peaks that were created owing to the participants' walking behaviors while they passed through the risky trail segments (e.g., making slipped steps and searching for stable next steps).

As shown in Figure 11, the peak detection algorithm tends to overestimate the strides in the risky trail segments. However, such errors do not significantly influence on the performance of our system. The key reason is that to identify risky trail segments, we trained our machine learning model using normal strides using one-class SVM. These kinds of errors are likely to be classified as abnormal strides, since their movement patterns are fairly different from those of normal ones.

7.2 Individual performance of the mountain trail surface classification

Figure 12 shows accelerometer data and classification results of one participant for each zone to explain how the surface classification algorithm works. The graph represents the Y-axis data from Zone 1 to Zone 5. One window has 10 strides. If the label of a particular window is risky, the signal value was plotted in red; otherwise, it was plotted in blue. In addition, the line below the graph is the classification result of the algorithm in a particular window. The line is colored in red if the algorithm classifies the window as risky; otherwise it is colored in blue.

In Figure 12, the short risky trail segments such as the beginning part of Zone 5 were classified as nonrisky. The reason of misclassification was that because we set the 10 strides data as one window, the algorithm could not perform well in the short risky trail segments which are shorter than 5 strides. These problems were owing to the smoothing effect of the windowing method. Thus, if we want to classify a short risky segment well, we should set a shorter window size. However, in the short window size, the overall performance of the algorithm could decrease. The detail trade-off of setting a window size will be explained in Section 7.4

In the accelerometer signal in Figure 12, there is a periodic stride pattern in the nonrisky labeled segments. In contrast, the signal changes have a non-periodic pattern in risky labeled segments. The algorithm classified these non-stationary patterns as risky. The main classification error is that the algorithm classifies some risky trail segments as nonrisky trail segments (i.e. false negative cases). If we examine the acceleration data of the false negatives, we find that the signals appear to be quite similar to those in nonrisky trail segments.

By checking video data to analyze the motion of participant, it is identified that participants make a detour to avoid the obstacles and keep their normal walking in corresponding segments. Skilled climbers showed this phenomenon frequently and change of those signals is relatively less sensitive to mountain surface. However, we aggregate monitoring results of many users to detect the risky trail segments. Thus we can detect the risky segment even though some skilled climbers cannot detect some risky trail segments. Furthermore, we can detect all risky trail segments by aggregating monitoring results. These results are explained in 7.5

7.3 Performance comparison of one-class SVM and binary-class SVM

In this section, we compare the classification performance between one-class SVM and binary-class SVM when a window size is set to 10 strides. We plot the results of multiple users using the box plot as shown in Figure 13. The algorithm using one-class classification shows 82% accuracy, 84% precision, and 62% recall; using binary class classification it shows 80% accuracy, 61% precision, and 69% recall.

This result shows that one-class classification is superior to binary class classification. It is interesting to note that mountain climbers have more chance to show abnormal walking in risky surface. However, this does not mean that all strides in risky surface are abnormal strides. It turns out that normal strides in risky surface disturb the machine learning algorithm to learn the normal walking if binary class classification is used. Thus, the performance of algorithm is inferior in the case of binary-class SVM.

7.4 Window length and algorithm performance

In this section, we explain the relationship between the window length and the algorithm performance. The shortest risky segment in the collected data described in Section 4 had an estimated average of 1.7 strides of 2.4 m length. In case of using the window method, if the window length is higher than twice the length of a risky segment, the algorithm classifies the window as nonrisky. Thus, with a longer window length, the possibility of missing risky trail segments increases. The left graph of Figure 14 shows the number of detectable risky trail segments if the algorithm performs perfectly. In a 5-stride window case, the algorithm can detect all the 13 risky trail segments, but in a 10-stride window case, there are only eight detectable risky trail segments. We observe that precision of the algorithm increases as the number of strides used in classification increases. This result can be seen in the right graph of Figure 14; the precision increases from 68% to 84% when the number of strides

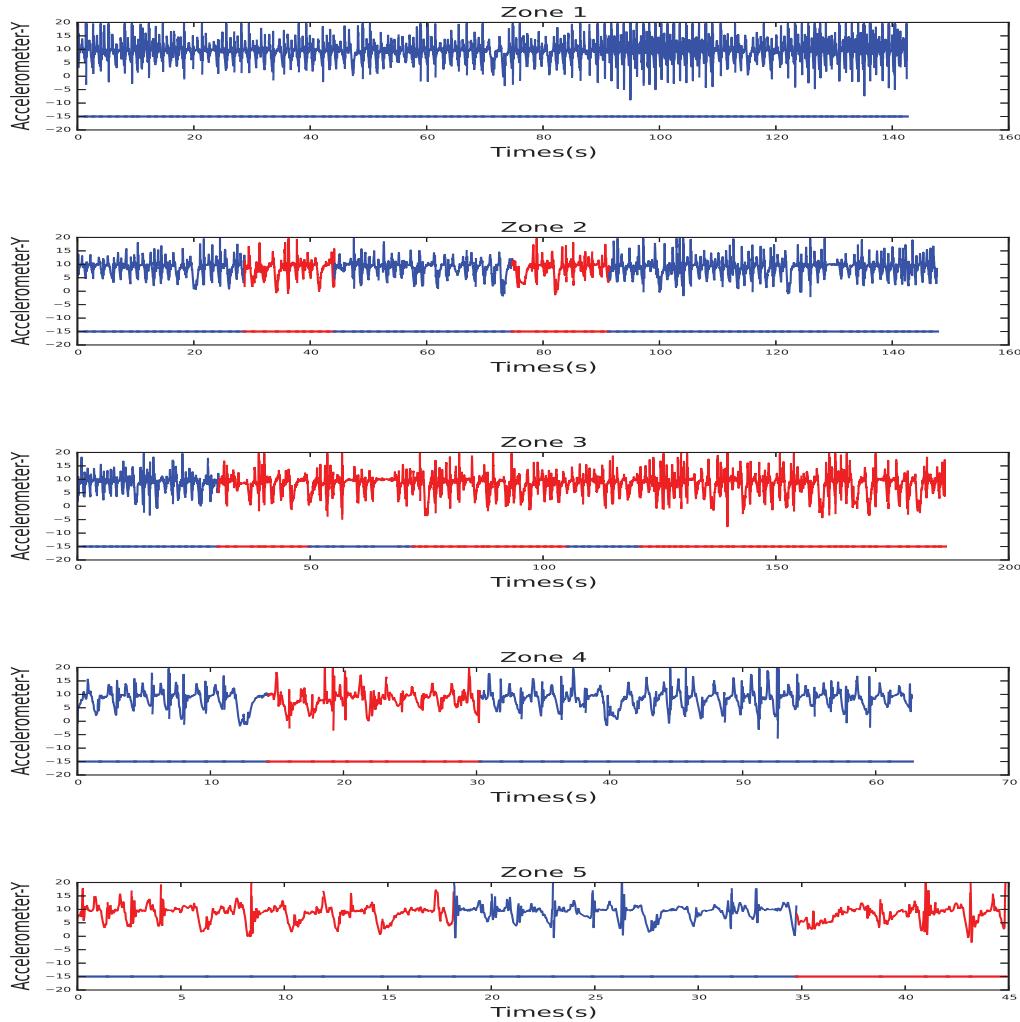


Fig. 12. Accelerometer Y-axis data and classification result from Zone 1 to the zone 5, respectively. A window size of 10 strides was used. If the label of a particular window is risky, the signal value has been plotted in red, and otherwise data was plotted in blue. Also, the line below the graph is the classification result of the algorithm in a particular window. The line colored red if the algorithm classifies the window as risky and otherwise colored blue.

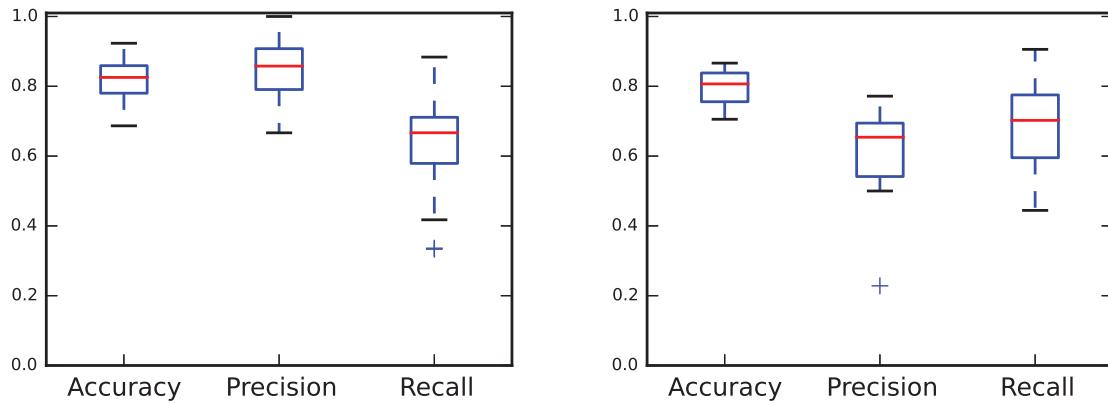


Fig. 13. The boxplot of performance of classification results with a window of 10 strides: one-class classification (left) and binary class classification (right)

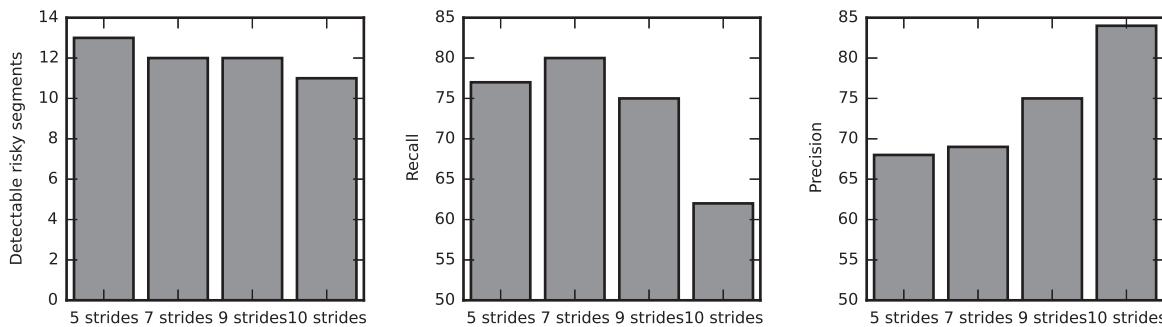


Fig. 14. The number of detectable risky trail segments (left), recall (middle), and precision (right) when the window length changes.

in one window increases from 5 to 10. Figure 15 shows the performance of multiple users when 5 stride data is partitioned as a single window using a boxplot. The one-class classification result shows an average of 78% accuracy, 66% precision, and recall of 76%. As mentioned above, this result shows lower precision and accuracy than those of 10-stride window. When using people as a sensor in contrast to vehicles, signals are not immediately affected by surface condition, but instead they change slowly with multiple strides. The longer window, with an adequate length for a person to react the surface, shows better performance in mountain surface classification.

7.5 Aggregation of individual monitoring results

Figure 16 shows the aggregation result of the DBSCAN algorithm for 10 stride window data of 14 participants in the Zone 1, Zone 2, Zone 3, Zone 4 and Zone 5 respectively. When applying the DBSCAN algorithm, the algorithm requires two parameters which are explained in Section 6. The minimum number of data points for clustering is set to 8 to create a cluster among 14 participants (this number is slightly greater than half of the

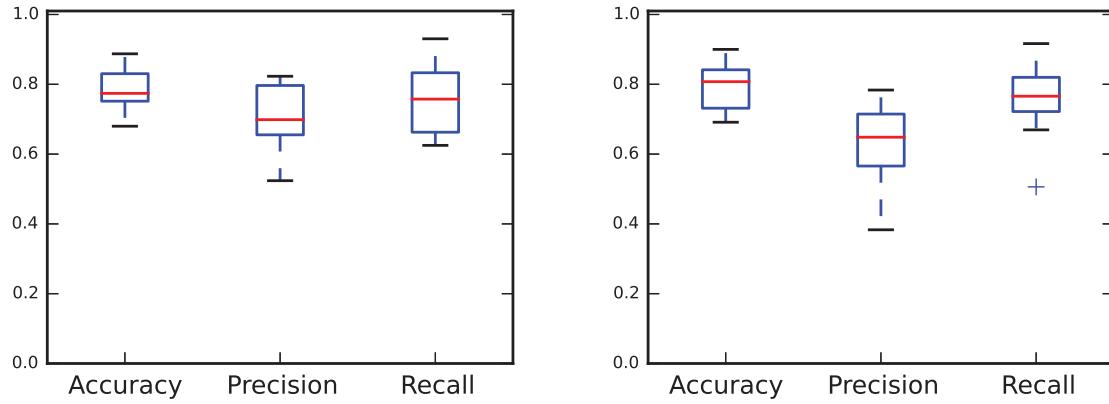


Fig. 15. The boxplot of performance of classification result using a window of 5 strides: one-class classification (left) and binary class classification (right)

participants. Additionally, the radius of the neighborhoods around a core data point (EPS parameter) is set to 10m, in order to consider the measurement error. In Figure 16, colored points represent the points which are included in some of the clusters, which means that the mountain surface classification algorithm is classified as risky. Also, the black points represent the points which are not included in any clusters, which means these points are considered to be false positives. Figure 16 shows that the number of clusters is smaller than the number of risky trail segments in the Zone 2 and Zone 3. In the Zone 2, four risky trail segments are merged into two clusters, and in the Zone 3, six risky trail segments are merged into three clusters. This is because risky trail segments are consecutively located such that short non-risky trail segments are located in between risky trail segments.

Table 3 shows a summary of the performance of the aggregation result. To identify whether the risky labeled data were included in the cluster, we count the monitoring result of each user, which is close to the GPS data of the risky labeled segments in the EPS parameter. As a result, we were able to accurately detect all risky segments using multi-user data by setting an appropriate EPS parameter. Owing to GPS measurement errors, the DBSCAN algorithm did not detect some risky trail segments when the EPS parameter setting was too short, such as 1m. However, the DBSCAN algorithm could detect all the risky trail segments with a sufficient EPS parameter, such as 10m, which could account for the GPS error. We were able to achieve 100% detection accuracy when we use the window size of 5 strides with an EPS of 10m.

Table 3. The number of detected risky trail segments

Window size\EPS	1m	5m	10m
5 Strides	10 (76%)	13 (100%)	13 (100%)
7 Strides	8 (61%)	11 (84%)	12 (92%)
9 Strides	7 (53%)	9 (69%)	12 (92%)
10 Strides	5 (38%)	8 (61%)	11 (84%)

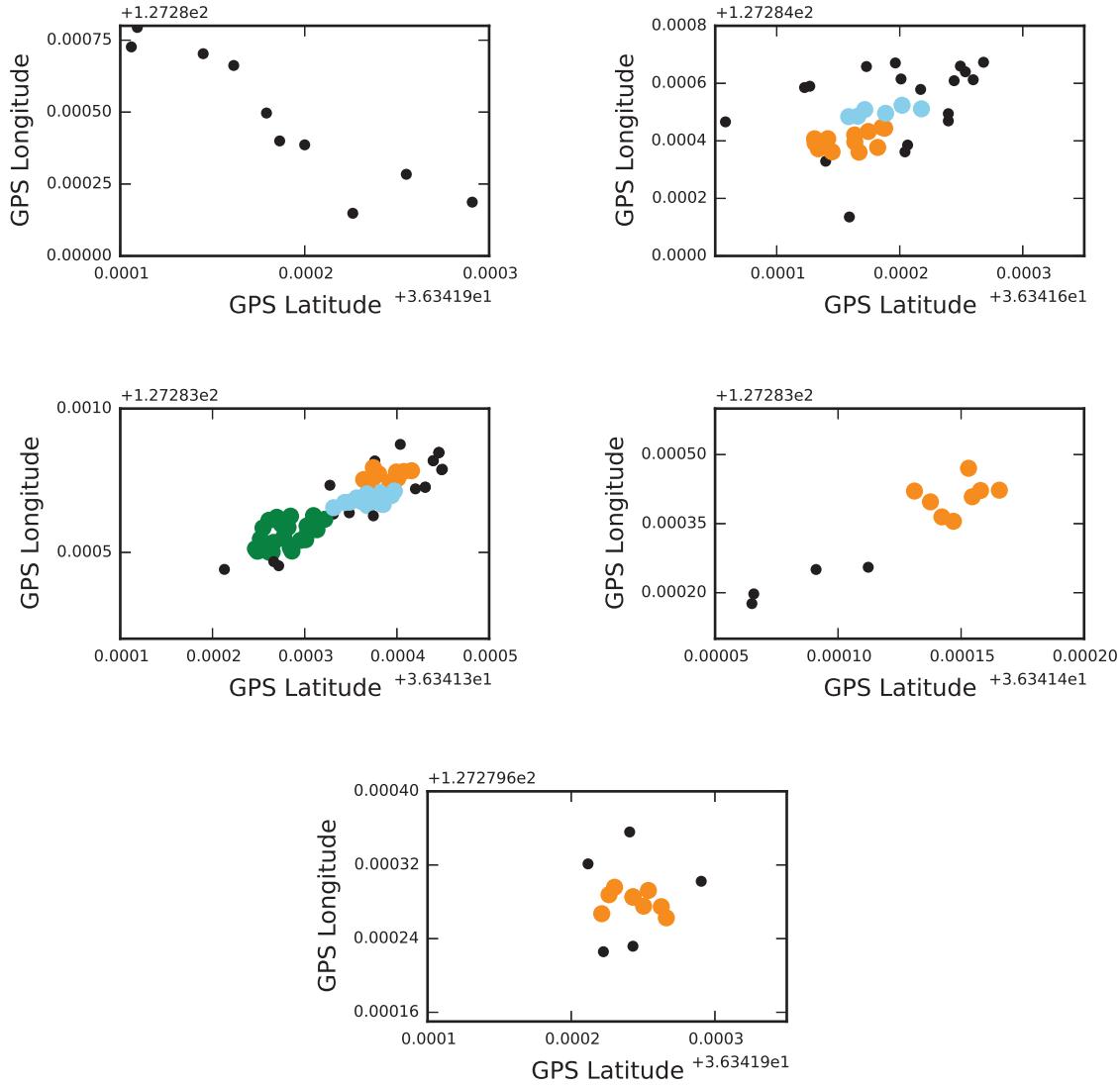


Fig. 16. DBSCAN algorithm results on the GPS readings of monitored risky window for individuals in Zone 1, Zone 2, Zone 3, Zone 4, and Zone 5. Black points represent the points which are not included in any clusters.

7.6 Further investigations with an additional dataset

We conducted an additional data collection to evaluate the algorithm when the algorithm trained by the data collected one trail applied to the other trails. In the previous experiment, we collected the data in one trail and separated the data into the five zones. We then used cross-validation to evaluate the performance. In reality, the model trained in one mountain trail should be also applicable to the other trails. Thus, we did an additional

experiment by collecting a new dataset from different mountain trails. We hired five participants (one female), and their mean age was 26.2 (SD=2.3). They trekked two different trails in the Gyeryongsan national park area, Daejeon, Korea, which were different from the location in the first dataset. The first trail was 1.5 km in length, containing nine risky trail segments, and the second trail was 0.9 Km in length with only one risky segment. In Table 4 risky trail segments are summarized with their length and average number of strides. In the second experiment we marked every risky trail segment by asking each participant to perform two jumps at the beginning and end of every risky and nonrisky segment. This act induces two consecutive sharp peaks in the acceleration. After the experiment, we visualize the acceleration data to detect these jumps and label the segments manually. Next, we used 0.9 Km of non-risky trail segments in one trail as training data when tested and nine, risky trail segments in the other trail to evaluate the algorithm when the algorithm trained by the data collected one trail applied to the other trails. In addiation, we used the leave-one-out cross validation method to test remaining data, nonrisky trail segments and 1 risky trail segments in that trail.

Table 4. Average number of strides (SC_{avg}) and length of risky (RS_n) and non-risky (NRS) segments in each trail (T_m).

	T1-RS1	T1-RS2	T1-RS3	T1-RS4	T1-RS5	T1-RS6	T1-RS7
Length (m)	20	29	18	25	15	14	10
SC_{avg}	29	35	22	35	22	18	19

	T1-RS8	T1-RS9	T2-RS10	T2-NRS
Length (m)	30	25	29	900
SC_{avg}	64	39	42	619

In order to examine the riskiness of each segment for each subject, we calculated the abnormal stride ratio for a window of 10-strides windows. The average ratio of all segments is summarized in Table 5. The minimum of 65% for risk ratio of segments shows that selected segments have fairly consistent level of riskiness. There were only three cases in which the risk ratio was less than 50%, but the average risk ratio for the related segment is still above 50%. The risk ratios for all the subjects in nonrisky trail segments were less than 50% as expected.

Table 5. Risk ratio of risky (RS_n) and non-risky (NRS) segments by each participant (P_m).

	RS1	RS2	RS3	RS4	RS5	RS6	RS7	RS8	RS9	RS10	AVG	NRS
P1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.75	0.98	0.25
P2	0.74	0.87	0.84	0.89	0.94	0.95	0.93	0.92	0.82	0.68	0.86	0.19
P3	0.65	0.62	0.56	0.47	1.00	0.88	0.57	0.54	0.54	0.84	0.72	0.17
P4	0.70	0.68	0.71	0.63	0.86	0.81	0.52	0.70	0.84	0.71	0.72	0.25
P5	0.40	0.57	0.57	0.67	0.66	0.55	0.52	0.91	0.83	0.39	0.61	0.21
AVG	0.70	0.75	0.74	0.73	0.89	0.84	0.71	0.81	0.65	0.67	0.78	0.21

We put aside all the risky windows along with 30 nonrisky windows for each subject which is proportion to the average number of risky windows for each subject in our test set and used the rest of the nonrisky windows as the training set. We summarized the classification results as true positive, true negative, false positive and false negative in Table 6 with respect to the individual participants. Our method achieved an average accuracy of 90% with average recall and precision of 81% and 98% respectively which shows its effectiveness of it in detecting risky trail segments. When compared with previous results, we posit that the improvement in classification

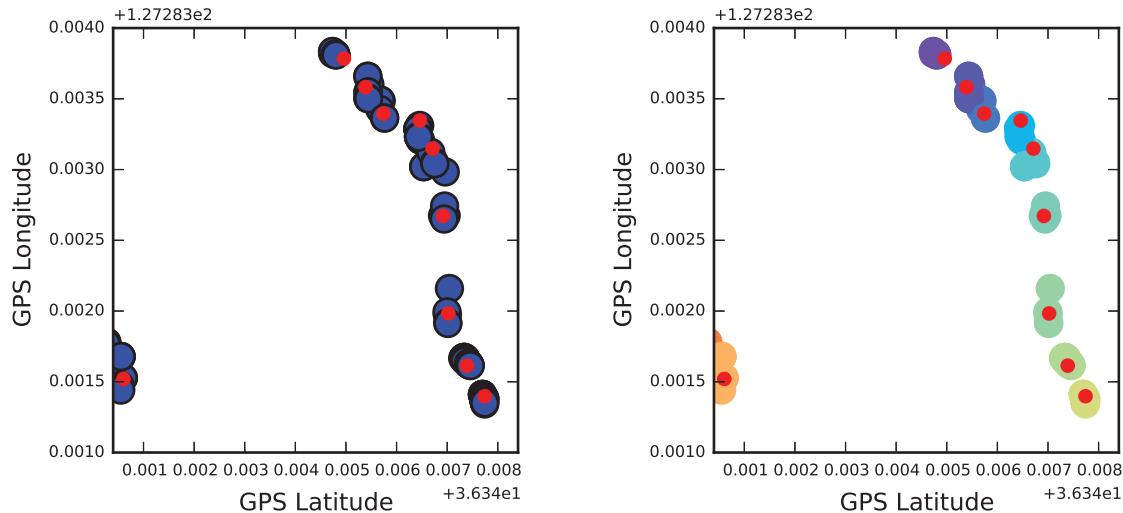


Fig. 17. Left : GPS readings of monitored risky trail segments for individuals. Right : DBSCAN algorithm results on the GPS readings of monitored risky trail segments for individuals. Red points represent the locations of risky trail segments based on the ground truth

performance is owing to the amount of non-risky segment data. In our previous dataset, we only collected non-risky segment data in approximately a 0.3 km length of mountain trails. To clarify the relationship between the performance of an algorithm and the amount of training data for one-class classification, we run the evaluation by varying the training dataset size (within an increment of one-third of the entire dataset). If two-thirds of training data is used, the mean accuracy, recall and precision of the algorithm declined to 91%, 86% and 94% respectively in individual data classification. When one-third of training data which is the same amount of training data in the first dataset is used, the mean accuracy, recall and precision of the algorithm declined even more, reaching 78%, 76%, 89% respectively. The results show that our method is able to achieve satisfactory results if there exists a sufficient training dataset. However, even with the lower-quality data of the first experiments, our method showed the 100% precision in detecting risky trail segments with the wisdom of the crowd after the aggregation step.

Table 6. Classification performance of each participant (P_m).

	TP	TN	FP	FN	Accuracy	Recall	Precision
P1	34	30	0	0	100%	100%	100%
P2	23	29	1	0	98%	100%	95%
P3	22	29	1	6	87%	78%	95%
P4	23	26	4	3	88%	88%	85%
P5	16	29	1	8	83%	66%	94%
AVG					91%	86%	94%

Figure 17(left) shows the GPS readings of a monitored risky segment for individuals and GPS readings of risky trail segments by the ground truth. Also, Figure 17(right) shows that GPS readings of monitored risky trail segments are clustered by the DBSCAN algorithm. As Figure 17 shows, 10 different colored clusters represent GPS readings of 10 risky trail segments. The difference between the center of a cluster and GPS reading of risky segment is the greatest in the purple cluster, but the difference is just 3.5m. Using this result, we can check that the all risky trail segments can be detected even with GPS error. To summarize, this additional experiment clearly showed that our algorithm is applicable over different mountain trails, and its performance has slightly increased owing to the size of the training dataset.

With the additional experiment, we can check that the algorithm can be applied in two different mountain trails and the performance of the algorithm is increased compared with the previous data set. This improvement comes from the amount of normal strides training data. We collected normal stride data in 0.9km length of nonrisky segment. Thus, in a real application, if the system can collect sufficient training data more than 0.9km length of a nonrisky segment, the algorithm can detect the risky trail segments accurately.

8 DISCUSSION

We discuss several approaches for dealing with phone placement/orientation issues, and then detail the limitations of our data collection and analysis methods.

8.1 Dealing with phone placement and orientation

We decided to use the fixed phone position because climbers are very likely to hold the phones in their pockets while climbing. The smartphone can be located on any part of the body but the front pocket is more common. Researchers examined various phone placements for activity recognition purposes and showed that the pocket has higher accuracy among other areas [50]. Moreover, the design of front pockets is usually such that the phone stands vertically in them. These findings support our decision to have our participants carry the phone in their pockets.

Although our work considered a single position, the phones can be freely movable in the pocket (small tilt angle and freedom to shake slightly was possible). We allowed the participants to place the phones naturally in their pocket. Despite of this small variance in phone placement, our method was still able to detect and classify gait cycles with acceptable accuracy; hence it is robust for these variations.

However, our model can be extended to consider different positions, ranging from different pockets or places with arbitrary orientation. Once the phone orientation is recognized, it can be changed to the target orientation decided in this paper. If the phone is fixed, but has a different orientation, the straightforward approach is to ask the user to reorient the phone manually. Another approach is to reorient it virtually using 3D transformations. If the phone is moving dynamically, dynamic reorientation or orientation-independent methods can be applied to address this issue. We discuss applicability of each of these approaches in the following.

Fixed placement and arbitrary orientation: As we mentioned above phones can have arbitrary placements. Even with the pocket placement, different orientations are still possible; a phone can be upward or downward, or it can be facing towards the body or upon the body. It is possible to apply an orientation-sensing step before data collection to alarm users that the position of a phone is not appropriate and requires correction. Orientation sensing also can run during the data collection to prevent the recording inaccurate data. Orientation correction can also be performed virtually using 3D transformation methods. For the transformation three rotation angles are required. The angles for 3D rotation can be calculated based on the amount of gravity acceleration projected onto other axes. Several studies have addressed the sensor orientation issue [36, 58]. For example, Tundo et al. [58] used a quaternion-based approach to apply the accelerometer transformation. Mohsen et al. [36] applied the 3D rotation matrices multiplications to reorient the acceleration. The rotation matrix can also be directly obtained from the Android API.

Dynamic placement and orientation: If the phone placement is dynamically changing, for example holding in hand or changing the placement multiple times, real-time reorientation will be challenging and may decrease the accuracy of transformed acceleration readings. Different methods were proposed to address dynamic reorientation [55, 67]. In our work, we did not consider dynamic placement, and further studies are needed to address this challenging issue.

Orientation-independent approach: Another approach is to use orientation-independent features such as acceleration magnitude instead of directly using 3D acceleration values [51]. For example, since the acceleration in the horizontal direction compared to gravity is small during walking, the magnitude of acceleration will mostly reflect the vertical acceleration. Using only acceleration magnitude will limit the features that can be extracted and result in information loss, although it has the advantage being applicable to both fixed and dynamic placements. The accuracy of our method adopting only orientation-independent features should be examined in the follow-up studies.

8.2 Limitations of collected data set and analysis

There are a few limitations for our data set and analysis. First, we only collected sensor data from a small number of young people from two sites. For generalization, further data collection is needed with a larger number of users in diverse age groups and at different sites. Our algorithm is based on walking patterns, which remain constant and their change gradually appears due to aging. Thus, we expect that our model can be generalized to other population and sites.

Second, the duration of data collection is limited since we collected the data from only part of a given trail. Data collection from a longer trail in a more naturalistic setting which allows us to explore several additional issues. For example, we can study how the length of a trail segment influences the performance. In addition, we can analyze how fatigue affects walking behaviors and their impacts on the performance.

Third, we excluded the downhill data. In Section 5.4 we thoroughly analyzed the variations in uphill and downhill walking patterns. We found that in downhill walking, participants tended to walk fast, and its patterns were less stable than those in uphill walking. In contrast, in uphill climbing, people feel more stable and walk without hastiness. A major limitation of excluding downhill data would be reduction of trail coverage, but this issue is not problematic. Most mountain trails are bi-directional, which means that hikers travel through the trails in both directions. In this case, we can easily collect uphill data of the entire trail. However, there could be mountain trails where hiking in one direction is mostly preferred by most hikers. In that case, it is difficult to collect the uphill dataset, thereby lowering the trail coverage. Despite this limitation, given that most trails are bidirectional, in practice trail coverage would not be a serious problem. In this paper, we did not further explore modeling downhill walking patterns, but it may be possible to leverage advance signal processing techniques, which we defer as part of future work.

9 CONCLUSION

We proposed a crowdsensing system for detecting risky mountain trail segments. By using sensor data collected from multiple hikers' smartphones, our system learned hikers' normal cyclic gait patterns and automatically inferred whether a user passes through a risky segment. Technically, wavelet analysis was used to extract the features of gait patterns, and one-class SVM was used to build a model for normal gait patterns and to identify abnormal gait patterns. Next, the system aggregated multiple hikers' results for robust classification using the DBSCAN algorithm. Our real-world dataset experiment showed that by walking pattern learning, the proposed system can accurately identify all the risky mountain trail segments after aggregating multiple reports. In addition, our additional experiment results showed that the model learned from one trail can be used to accurately identify risky trail segments in the other trail. There are several directions for future work. For generalizability, there should be further experimental evaluations by considering other user groups and possibly

by increasing the number of participants. To support practical deployment, we should carefully deal with the effect of orientation/placement variations as well as device heterogeneity. We can consider reputation models based on Bayesian frameworks to deal with incremental data collection from multiple climbers.

10 APPENDIX

These tables summarize the length and the number of strides for all non-risky and risky trail segments in each zone. For instance, in the tables, ‘RS1’ means the first risky segment, and ‘NRS1’ means the first non-risky segment in each zone. All segments are numbered sequentially.

Table 7. Average number of strides (SC_{avg}) and length of risky (RS_n) and non-risky (NRS_n) segments in Zone 1.

	Zone 1	NRS1	Total
Length (m)	149	149	
SC_{avg}	107	107	

Table 8. Average number of strides (SC_{avg}) and length of risky (RS_n) and non-risky (NRS_n) segments in Zone 2.

Zone 2	NRS1	RS1	NRS2	RS2	NRS3	RS3	NRS4	RS4	NRS5	Total
Length (m)	22.4	7.3	23.4	9.5	7.7	8.2	13.1	4.2	18.6	109.0
SC_{avg}	16	5.2	16.7	6.8	5.5	5.9	9.4	3.1	13.3	77.9

Table 9. Average number of strides (SC_{avg}) and length of risky (RS_n) and non-risky (NRS_n) segments in Zone 3.

Zone 3	NRS1	RS1	NRS2	RS2	NRS3	RS3	NRS4
Length (m)	19.7	5.74	4.3	18.62	5.7	12.1	4.0
SC_{avg}	14.1	4.1	3.1	13.3	4.1	8.7	2.9

Zone 3	RS4	NRS5	RS5	NRS6	RS6	NRS7	Total
Length (m)	4.0	9.5	23.2	5.0	10.8	2.4	125.4
SC_{avg}	2.9	6.8	16.6	3.6	7.7	1.7	89.6

Table 10. Average number of strides (SC_{avg}) and length of risky (RS_n) and non-risky (NRS_n) segments in Zone 4.

Zone 4	NRS1	RS1	NRS2	Total
Length (m)	7.8	8.4	34.6	47.3
SC_{avg}	5.6	6.0	24.7	36.3

ACKNOWLEDGMENTS

This work was partly supported by the Basic Science Research Programs through the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) (No. NRF-2015R1D1A1A01059497, NRF-2015R1C1A1A02037090), and KAIST HRHRP Project. The corresponding authors of this work are Uichin Lee (ucllee@kaist.ac.kr) and Heeyoung Kim (heeyoungkim@kaist.ac.kr).

Table 11. Average number of strides (SC_{avg}) and length of risky (RS_n) and non-risky (NRS_n) segments in Zone 5.

Zone 5	NRS1	RS1	NRS2	RS2	NRS3	Total
Length (m)	3.2	8.2	6.3	4.3	1.7	23.7
SC_{avg}	2.3	5.8	4.5	3.1	1.2	16.9

REFERENCES

- [1] Jens Barth, Jochen Klucken, Patrick Kugler, Thomas Kammerer, Ralph Steidl, Jürgen Winkler, Joachim Hornegger, and Björn Eskofier. 2011. Biometric and mobile gait analysis for early diagnosis and therapy monitoring in Parkinson's disease. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 868–871.
- [2] Agata Brajdic and Robert Harle. 2013. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 225–234.
- [3] Jeffrey A Burke, Deborah Estrin, Mark Hansen, Andrew Parker, Nithya Ramanathan, Sasank Reddy, and Mani B Srivastava. 2006. Participatory sensing. *Center for Embedded Network Sensing* (2006).
- [4] Francesco Calabrese and Carlo Ratti. 2006. Real time rome. *Networks and Communication studies* 20, 3-4 (2006), 247–258.
- [5] KT Chang, JR Chang, and JK Liu. 2005. Detection of pavement distresses using 3D laser scanning technology. In *Proceedings of the 2005 ASCE international conference on computing in civil engineering*, 12–15.
- [6] Hyojin Chin, Hengameh Zabihi, Sangkeun Park, Mun Yong Yi, and Uichin Lee. 2017. WatchOut: Facilitating Safe Driving Behaviors with Social Support. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. ACM, New York, NY, USA, 2459–2465. DOI : <http://dx.doi.org/10.1145/3027063.3053188>
- [7] Dae-Ki Cho, Min Mun, Uichin Lee, William J Kaiser, and Mario Gerla. 2010. Autogait: A mobile platform that accurately estimates the distance walked. In *Pervasive computing and communications (PerCom), 2010 IEEE international conference on*. IEEE, 116–124.
- [8] Fengyu Cong, Hannu Hautakangas, Jukka Nieminen, Oleksiy Mazhelis, Mikko Perttunen, Jukka Riekki, and Tapani Ristaniemi. 2013. Applying wavelet packet decomposition and one-class support vector machine on vehicle acceleration traces for road anomaly detection. In *International Symposium on Neural Networks*. Springer, 291–299.
- [9] Shane B Eisenman, Emiliano Miluzzo, Nicholas D Lane, Ronald A Peterson, Gahng-Seop Ahn, and Andrew T Campbell. 2009. BikeNet: A mobile sensing system for cyclist experience mapping. *ACM Transactions on Sensor Networks (TOSN)* 6, 1 (2009), 6.
- [10] Mahmoud El-Gohary, Sean Pearson, James McNames, Martina Mancini, Fay Horak, Sabato Mellone, and Lorenzo Chiari. 2013. Continuous monitoring of turning in patients with movement disability. *Sensors* 14, 1 (2013), 356–369.
- [11] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. 2008. The pothole patrol: using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*. ACM, 29–39.
- [12] Korea forest service. 2007. Basic plan for climbing support. (2007).
- [13] Raghu K Ganti, Fan Ye, and Hui Lei. 2011. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine* 49, 11 (2011), 32–39.
- [14] Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y Yen, Runhe Huang, and Xingshe Zhou. 2015. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM Computing Surveys (CSUR)* 48, 1 (2015), 7.
- [15] Yuya Hanai, Jun Nishimura, and Tadahiro Kuroda. 2009. Haar-like filtering for human activity recognition using 3D accelerometer. In *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th*. IEEE, 675–678.
- [16] Yi He and Ye Li. 2013. Physical activity recognition utilizing the built-in kinematic sensors of a smartphone. *International Journal of Distributed Sensor Networks* 2013 (2013).
- [17] Yi He, Ye Li, and Shu-Di Bao. 2012. Fall Detection by built-in tri-accelerometer of smartphone. In *Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics*. IEEE, 184–187.
- [18] Lokeshwor Huidrom, Lalit Kumar Das, and SK Sud. 2013. Method for automated assessment of potholes, cracks and patches from road surface video clips. *Procedia-Social and Behavioral Sciences* 104 (2013), 312–321.
- [19] Shubham Jain, Carlo Borgiattino, Yanzhi Ren, Marco Gruteser, Yingying Chen, and Carla Fabiana Chiasserini. 2015. Lookup: Enabling pedestrian safety services via shoe sensing. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 257–271.
- [20] GM Jog, C Koch, M Golparvar-Fard, and I Brilakis. 2012. Pothole properties measurement through visual 2D recognition and 3D reconstruction. In *Proceedings of the ASCE International Conference on Computing in Civil Engineering*. 553–560.
- [21] Deon Joubert, Ayanda Tyatyantsi, Jeffry Mphahlehle, and Vivian Manchidi. 2011. Pothole tagging system. (2011).
- [22] Adil Mehmood Khan, Young-Koo Lee, Sungyoung Y Lee, and Tae-Seong Kim. 2010. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE transactions on information technology in biomedicine* 14,

- 5 (2010), 1166–1172.
- [23] Auk Kim, Sungjoon Kang, and Uichin Lee. 2017. LetsPic: Supporting In-situ Collaborative Photography over a Large Physical Space. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4561–4573. DOI: <http://dx.doi.org/10.1145/3025453.3025693>
 - [24] Jaejeung Kim, Sergey Leksikov, Punyotai Thamjamrassri, Uichin Lee, and Hyeon-Jeong Suk. 2015. CrowdColor: Crowdsourcing Color Perceptions Using Mobile Devices. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 478–483. DOI: <http://dx.doi.org/10.1145/2785830.2785887>
 - [25] Christian Koch and Ioannis Brilakis. 2011. Pothole detection in asphalt pavement images. *Advanced Engineering Informatics* 25, 3 (2011), 507–515.
 - [26] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. 2011. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter* 12, 2 (2011), 74–82.
 - [27] Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. 2010. A survey of mobile phone sensing. *IEEE Communications magazine* 48, 9 (2010), 140–150.
 - [28] Hong-Yin Lau, Kai-Yu Tong, and Hailong Zhu. 2008. Support vector machine for classification of walking conditions using miniature kinematic sensors. *Medical & biological engineering & computing* 46, 6 (2008), 563–573.
 - [29] Uichin Lee and Mario Gerla. 2010. A survey of urban vehicular sensing platforms. *Computer Networks* 54, 4 (2010), 527 – 544. DOI: <http://dx.doi.org/10.1016/j.comnet.2009.07.011> Advances in Wireless and Mobile Networks.
 - [30] Uichin Lee, Eugenio Magistretti, Biao Zhou, Mario Gerla, Paolo Bellavista, and Antonio Corradi. 2006. MobEyes: Smart mobs for urban monitoring with a vehicular sensor network. *IEEE Wireless Communications* 13, 5 (2006), 52–57.
 - [31] Qingguang Li, Ming Yao, Xun Yao, and Bugao Xu. 2009. A real-time 3D scanning system for pavement distortion inspection. *Measurement Science and Technology* 21, 1 (2009), 015702.
 - [32] Liang Liu, Assaf Biderman, and Carlo Ratti. 2009. Urban mobility landscape: Real time monitoring of urban mobility patterns. In *Proceedings of the 11th International Conference on Computers in Urban Planning and Urban Management*. Citeseer, 1–16.
 - [33] Artis Mednīs, Girts Strazdīns, Reinholds Zviedris, Georgijs Kanonirs, and Leo Selavo. 2011. Real time pothole detection using android smartphones with accelerometers. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*. IEEE, 1–6.
 - [34] Chihiro Mizuike, Shohei Ohgi, and Satoru Morita. 2009. Analysis of stroke patient walking dynamics using a tri-axial accelerometer. *Gait & posture* 30, 1 (2009), 60–64.
 - [35] I Moazzam, K Kamal, S Mathavan, S Usman, and M Rahman. 2013. Metrology and visualization of potholes using the microsoft kinect sensor. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 1284–1291.
 - [36] Nesma Mohssen, Rana Momtaz, Heba Aly, and Moustafa Youssef. 2014. It's the human that matters: accurate user orientation estimation for mobile computing applications. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 70–79.
 - [37] Catherine Morency, Martin Trepanier, and Bruno Agard. 2007. Measuring transit use variability with smart-card data. *Transport Policy* 14, 3 (2007), 193–203.
 - [38] Min Mun, Sasank Reddy, Katie Shilton, Nathan Yau, Jeff Burke, Deborah Estrin, Mark Hansen, Eric Howard, Ruth West, and Péter Boda. 2009. PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 55–68.
 - [39] Youngtae Noh, Hirozumi Yamaguchi, and Uichin Lee. 2017. Infrastructure-Free Collaborative Indoor Positioning Scheme for Time-Critical Team Operations. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* PP, 99 (2017), 1–15. DOI: <http://dx.doi.org/10.1109/TSMC.2016.2615652>
 - [40] Youngtae Noh, Hirozumi Yamaguchi, Uichin Lee, Prerna Vij, Joshua Joy, and Mario Gerla. 2013. CLIPS: Infrastructure-free collaborative indoor positioning scheme for time-critical team operations. In *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 172–178. DOI: <http://dx.doi.org/10.1109/PerCom.2013.6526729>
 - [41] Ministry of education Korea. 2015. Educational Statistics Annual 2015.
 - [42] Ministry of public safety and MPSS security. 2016. Climbing accident increased by 140% for 5 years. Press Release. (April 2016).
 - [43] Outdoorfoundation. 2015. Outdoor recreation participation topline report 2015. (2015).
 - [44] G Palshikar and others. 2009. Simple algorithms for peak detection in time-series. In *Proc. 1st Int. Conf. Advanced Data Analysis, Business Analytics and Intelligence*.
 - [45] Ion PI Pappas, Milos R Popovic, Thierry Keller, Volker Dietz, and Manfred Morari. 2001. A reliable gait phase detection system. *IEEE Transactions on neural systems and rehabilitation engineering* 9, 2 (2001), 113–125.
 - [46] Sangkeun Park, Emilia-Stefania Ilincai, Jeungmin Oh, Sujin Kwon, Rabeb Mizouni, and Uichin Lee. 2017. Facilitating Pervasive Community Policing on the Road with Mobile Roadwatch. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3538–3550. DOI: <http://dx.doi.org/10.1145/3025453.3025867>

- [47] Sangkeun Park, Joohyun Kim, Rabeb Mizouni, and Uichin Lee. 2016. Motives and Concerns of Dashcam Video Sharing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4758–4769. DOI:<http://dx.doi.org/10.1145/2858036.2858581>
- [48] Stephen J Preece, John Yannis Goulermas, Laurence PJ Kenney, and David Howard. 2009. A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *Biomedical Engineering, IEEE Transactions on* 56, 3 (2009), 871–879.
- [49] Cliff Randell, Chris Djallis, and Henk Muller. 2003. Personal Position Measurement Using Dead Reckoning. In *Proceedings of the 7th IEEE International Symposium on Wearable Computers (ISWC '03)*. IEEE Computer Society, Washington, DC, USA, 166–. <http://dl.acm.org/citation.cfm?id=946249.946879>
- [50] Sasank Reddy, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. 2008. Determining transportation mode on mobile phones. In *Wearable computers, 2008. ISWC 2008. 12th IEEE International symposium on*. IEEE, 25–28.
- [51] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. 2010. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)* 6, 2 (2010), 13.
- [52] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural computation* 13, 7 (2001), 1443–1471.
- [53] Dominik Schuldhaus, Patrick Kugler, Ulf Jensen, Bjoern Eskofier, Heiko Schlarb, and Magnus Leible. 2012. Classification of surfaces and inclinations during outdoor running using shoe-mounted inertial sensors. In *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2258–2261.
- [54] Masaki Sekine, Toshiyo Tamura, Toshiro Fujimoto, and Yasuhiro Fukui. 2000. Classification of walking pattern using acceleration waveform in elderly people. In *Engineering in Medicine and Biology Society, 2000. Proceedings of the 22nd Annual International Conference of the IEEE*, Vol. 2. IEEE, 1356–1359.
- [55] Dianxi Shi, Ran Wang, Yuan Wu, Xiaoyun Mo, and Jing Wei. 2017. A novel orientation-and location-independent activity recognition method. *Personal and Ubiquitous Computing* (2017), 1–15.
- [56] Yu-chin Tai, Cheng-wei Chan, and Jane Yung-jen Hsu. 2010. Automatic road anomaly detection using smart mobile device. In *conference on technologies and applications of artificial intelligence, Hsinchu, Taiwan*. Citeseer.
- [57] Andrew J Teichtahl, Anita E Wluka, Boyd J Strauss, Yuanyuan Wang, Patricia Berry, Miranda Davies-Tuck, and Flavia M Cicuttini. 2012. The associations between body and knee height measurements and knee joint structure in an asymptomatic cohort. *BMC musculoskeletal disorders* 13, 1 (2012), 1.
- [58] Marco D Tundo, Edward Lemaire, and Natalie Baddour. 2013. Correcting Smartphone orientation for accelerometer-based analysis. In *Medical Measurements and Applications Proceedings (MeMeA), 2013 IEEE International Symposium on*. IEEE, 58–62.
- [59] Jeen-Shing Wang, Che-Wei Lin, Ya-Ting C Yang, and Yu-Jen Ho. 2012. Walking pattern classification and walking distance estimation algorithms using gait phase information. *IEEE Transactions on Biomedical Engineering* 59, 10 (2012), 2884–2892.
- [60] Ning Wang, Eliathamby Ambikairajah, Branko G Celler, and Nigel H Lovell. 2008. Accelerometry based classification of gait patterns using empirical mode decomposition. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 617–620.
- [61] Ning Wang, Eliathamby Ambikairajah, Nigel H Lovell, and Branko G Celler. 2007. Accelerometry based classification of walking patterns using time-frequency analysis. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 4899–4902.
- [62] Ning Wang, Eliathamby Ambikairajah, Stephen J Redmond, Branko G Celler, and Nigel H Lovell. 2009. Classification of walking patterns on inclined surfaces from accelerometry data. In *2009 16th International Conference on Digital Signal Processing*. IEEE, 1–4.
- [63] Ning Wang, Stephen J Redmond, Eliathamby Ambikairajah, Branko G Celler, and Nigel H Lovell. 2010. Can triaxial accelerometry accurately recognize inclined walking terrains? *IEEE Transactions on Biomedical Engineering* 57, 10 (2010), 2506–2516.
- [64] Guangtao Xue, Hongzi Zhu, Zhenxian Hu, Wen Zhuo, Chao Yang, Yanmin Zhu, Jiadi Yu, and Yuan Luo. 2016. Pothole in the Dark: Perceiving Pothole Profiles with Participatory Urban Vehicles. *IEEE Transactions on Mobile Computing* (2016).
- [65] Sahin Yildirim and Ifjibrahim Uzmay. 2003. Neural network applications to vehicle's vibration analysis. *Mechanism and Machine Theory* 38, 1 (2003), 27–41.
- [66] Bill X. Yu and Xianbao Yu. 2006. Vibration-based system for pavement condition evaluation. In *Applications of Advanced Technology in Transportation. The Ninth International Conference*.
- [67] Yu Zhong and Yunbin Deng. 2014. Sensor orientation invariant mobile gait biometrics. In *Biometrics (IJCB), 2014 IEEE International Joint Conference on*. IEEE, 1–8.
- [68] Pengfei Zhou, Yuanqing Zheng, and Mo Li. 2012. How long to wait?: predicting bus arrival time with mobile phone based participatory sensing. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 379–392.
- [69] Chun Zhu and Weihua Sheng. 2009. Human daily activity recognition in robot-assisted living using multi-sensor fusion. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2154–2159.

Received February 2017; revised May 2017; accepted July 2017.