UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

Department of Computer Science

Computer Science Honours Final Paper 2016

Title:	MOVER Runner: Using Widely Available Smartphone Accelerometers to Detect Runner Falls
Author:	Gavin Wiener
Project abbreviation:	MOVER
Supervisor:	Tommie Meyer and Francois Mouton

Category	**Min**	**Max**	**Chosen**
Requirement Analysis and Design	0	20	
Theoretical Analysis	0	25	
Experiment Design and Execution	0	20	
System Development and Implementation	0	15	
Results, Findings and Conclusion	10	20	
Aim Formulation and Background Work	10	15	
Quality of Paper Writing and Presentation	10		
Quality of Deliverables	10		
Overall General Project Evaluation	0	10	
Total marks			

MOVER Runner: Using Widely Available Smartphone Accelerometers to Detect Runner Falls

Write-Up

Gavin Wiener
University of Cape Town
wnrgav001@myuct.ac.za

ABSTRACT

Each year the stockpile of old and new Android smartphone devices increases [21]. In addition, the functionality and hardware capabilities increase and many older devices, fully and partially functional, are replaced. The underutilized resources can be and have been re-purposed. South Africa has a rich running, outdoor activities culture, and a strong Android mobile-device presence [3]. Outdoor activities have risks though; participants are often isolated, terrain can be uneven and dangerous, and there are incidents of crime [23, 17, 22, 8]. Re-purposed smart devices has successfully detected motor-vehicle accidents [2, 16], geriatric falls [24], devices' accelerometers have sufficient sensitivity to identify potholes when a road's quality deteriorates [9, 7] and reckless driving, [4]. Identifying falls among runners shared approaches with these previous works; magnitudes of the accelerometer's value vectors were recorded at increasing running speeds while the runner carried an Android device in an arm-band, pocket, or backpack, and along various routes. In addition; magnitudes were also calculated and recorded while simulating falls a runner may experience. A static magnitude threshold of approximately 15 detected nine out of 10 falls on a final obstacle course but false-positives were numerous. Reducing false-positives may require reanalysis of fall magnitudes, analyzing proceeding and preceding magnitude values would provide context whether a relative magnitude is abnormal, or a combination of both. Further; the field includes potential for various machine-learning or learning-algorithms due to the physical differences between runners and preferences of running style and equipment.

Keywords

Android; collision-detection; running, falling, accidents

1. INTRODUCTION

The total number of smartphones (specifically entry-level or budget) increases every year. Smartphone technology is becoming more powerful and the costs are decreasing [6]. Increased turnover is further promoted by many cellular contracts offering a new smartphone every few years and a cultural demand for the most recent mobile devices. Decreasing costs allow entry-level devices to have Global Positioning System ('GPS') and accelerometer capabilities, capabilities which may only have been available to high-end smartphones until recently. Further; mobile penetration is increasing (especially in developing countries)[3] which extends these capabilities to members of countries which may lack technical infrastructure.

An increasing total number of smartphones results in underutilized computational power of millions of smartphones, and potential pollution. Many smartphones may be thrown away if they can no longer be used as intended (the speaker phone or touch screen is not fully responsive) or are replaced and despite recycling this can create electronic waste, batteries which contain Lithium and various metals (Cobalt, Nickel, Silver) can leak [15]. These devices can be (and have been) effectively re-purposed; functions including in-car parking meters.

These concerns are not Android specific but the Android platform is often chosen because the Android operating system ('OS') is well-maintained and open-source which allows modifying the OS to run on less powerful devices or for a specific purpose. Android devices have also gained wide use in developing countries and worldwide [21, 19]. The sensors available vary between devices but sensors like accelerometers and GPS have become standard since Android 2.3 (Gingerbread). Gingerbread was released in 2010 and ss of September 5 2016, there are only 0.1% Android devices running an OS older than Gingerbread [13]. Therefore there is a large pool of potentially underutilized Android smartphones which can provide rich sensor functionality.

The Council for Scientific and Industrial Research ('CSIR') has developed the Mobile On-Board Vehicle Event Recorder ('MOVER') project which aims to detect motor-vehicle collisions and high-impact runner events which may cause injury. In addition; there is a specific focus on re-purposed Android devices as they align with the economic environment of South Africa, it is a developing country and Android devices are some of the cheapest smartphones. Re-purposed devices have found use in multiple fields; motor-car accident collision, human-machine safety in manufacturing, and road-condition analysis [9, 7, 2, 4, 16, 5, 20]. South Africa has a large community and culture of running and interaction with nature; these interactions include competitive and recreational road running, trail-runs, and hiking. These activities include an aspect of isolation, which many enjoy, but

DOI: 10.1145/1235

there are also risks; there have been instances of runners or hikers falling and injuring themselves, getting lost, and being physically assaulted when the locations are located near city centres which have areas of high crime and poverty rates [23, 17, 22, 8]. It is often recommended a hiker or runner keeps a mobile device with them (and is often a requirement for trial run events) but they may not have the opportunity to contact emergency services as they may not be conscious or have access to their mobile device if they have been injured or attacked.

Could past approaches of collision or fall detection using Android smart-phones be extended to runners?

Accelerometer values must be sampled from use-cases which a smartphone would experience during regular usage in the context of a runner. During a trial-run or road run, a phone is likely to be in three different locations; the runner's arm-band, the runner's front or back pocket in their pocket, or in a backpack which is often a requirement during a trial-run by the event organisers.

Section 2 will discuss the reason for choosing the Android-based mobile-devices, the accelerometer functionality available to these devices, and previous work which re-purposed the devices. *Section 3* will discuss the chosen method for establishing a fall detection threshold and considerations and adjustments which had to be made to capture and log the accelerometer values and various events while performing trials.*Section 4* discusses the different routes and environments under which accelerometer values were captured with the goal of satisfying as many of the experiences a mobile-device carried by a runner would undergo. Finally; *Section 5* will discuss and display notable results from the various trails, highlight the analysis process which determined a fall-detection threshold, and any complications which were experienced.

2. BACKGROUND

2.1 Android Operating System and Mobile Devices

Android is an open-source operating system ('OS') developed by Google implementing and Linux-based kernel. Unlike closed OS's implemented by some alternate mobile-device developers and producers, Android promotes improvement and re-use, some Android-based devices allow custom firmware by default without any further technical involvement from the end-user or developer [12]. Therefore, entry-level smartphones significantly lean towards Android because performance and features can be catered and stripped to match the hardware capabilities of these devices. Android based devices are ideal for re-purposing.

2.2 (Android) Accelerometer

An accelerometer is a component which measures forces applied on the device across three axes of movement; x-axis, y-axis, and z-axis. The units are metres per second squared (m/s^2). Accelerometers used in smartphones introduce an interesting use-case as the device is mobile along all three axes i.e. the device can be moved in a three-dimensional space and rotated. Given this; the three axes are not constant with the ground (and thus gravity) as the three axes are constant with the sides and face of the mobile-device [11].

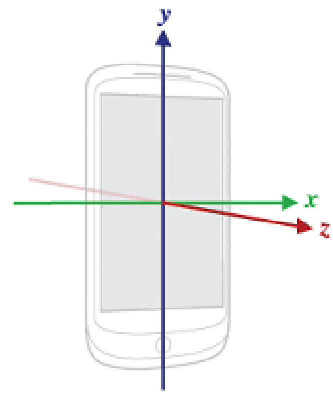

Figure 1: A diagram indicating the direction of movement and association with the three axes of a mobile-device

Given the definition of the diagram; the z-axis will always been associated with the face (screen) of the mobile device, the y-axis with the sides of device associated with its width etc.

The values returned by the accelerometers include the effects of gravity and are a vector i.e. a stationary device will reflect an approximate accelerometer value of 9.82 m/s^2 along one of the three axes associated with gravity.

2.3 Previous Work

Re-purposing Android devices for accident collision has leaned towards motor-vehicles, with success, and the accelerometer was the one sensor consistently utilized between works [26, 2, 27, 16]. Few works have focused on runners but success of works focusing on pothole detection, reckless driving, and road quality identification (which don't include as large accelerometer forces as accidents) indicates that a runner's fall or collision is likely to be within the capabilities of accelerometers.

Several works have attempted to implement other sensor types like microphone and light in addition to an accelerometer but the improvements in accident collision were minimal [26]. Detection of accidents were mostly achieved by either matching accelerometer templates or thresholds.

2.3.1 Reckless Driving Detection

Drunk driving detection was achieved with pattern matching. Researchers established a template for various actions which indicate reckless driving. If recorded data matched multiple established templates, the chances that an occupant is drunk or driving recklessly would increase. The cues included; weaving, drifting, swerving across lanes, accelerating or decelerating suddenly, braking erratically and stopping inappropriately, slow response to signals, no headlights and wide turns. Swerving and drifting affects lateral acceleration, abnormal side to side movements (drifting across lanes). Spastic acceleration affects longitudinal acceleration (front-to-back). Instead of relying on mounting the device, calibrating to the orientation was implemented, this makes more sense considering a drunk driver or reckless driver is unlikely to go through the effort of mounting a phone.

2.3.2 Car Crash Detection on Smartphones

[16] considered the use of their application as a produc-

tion application, if a suspected accident occurred, the location of the incident was recorded. A g-force threshold using accelerometer values was established whereby it could stated, any values greater than the threshold were not possible under regular driving conditions. 'Regular' conditions included hard-braking and rapid acceleration, and dropping the smartphone during usage. The smartphone was not mounted, and thus forces exerted on the mobile by intended usage were established; carry the mobile in a pocket, walking around, walking up stairs. This threshold was approximately $3g$'s. Dropping the mobile has a distinctive pattern. Testing the algorithm with real motor collisions is not feasible; the paper acquired test data by the US American governmental database which provides free velocity and accelerometer data from motor-vehicle test crashes [16, 1].

2.3.3 *Pothole Detection and Road Quality Approximation*

Pothole detection and road quality estimation would be impossible without the accelerometer (except in one case). The paper records accelerometer values along all three-axes when a driver encounters an incident of deterioration in road quality [9] i.e. sudden drops into a pothole or ditch [20], wobbling to the left and right on uneven roads. Isolating pothole detection and recall to only the local device was not feasible as a driver is more likely to avoid the same pothole along a route often traveled - there is little benefit indicating a driver has experienced a pothole when it was first-hand experienced by the same driver. Therefore; crowd-sourcing was also a focus for the paper. Pothole data is uploaded from each device running the application to a central server, and other drivers running the application benefit from others' data [7]. Crowd-sourcing benefits the driver, but papers stated the opportunity for government programmes or departments which focus on road maintenance to use recorded data and confirmed data by crowd-sourcing to preempt serious road quality deterioration. Further; if a driver is notified that a road has poor quality, the driver is likely to avoid the road and thus further prevent additional deterioration.

2.3.4 *ICEDot Crash Sensor*

ICEDot is a retail product which is marketed towards cyclists. It is a bluetooth device which attaches to the helmet of a rider and if critical forces are detected (as quoted on the website) the mobile-device will send an emergency response. The ICEdot includes the same sensor types as most smartphones; the common triple-axis accelerometer (a x, y, and z axis as discussed) which costs approximately 1.00 USD, and a triple axis gyroscope which is not often included with entry-level smart-phones because it is a more expensive component than an accelerometer but is still often included in the higher prices brackets, gyroscopes are required for most Virtual Reality and Augmented Reality mobile applications [14].

3. ESTABLISHING THRESHOLDS

There are two approaches from previous works which could determine a potential runner collision; either the forces of a collision upon a mobile-device are established and if those forces are met, an collision has occurred. Alternatively the forces upon a mobile-device during expected usage are established and if forces which are outside of the expected range occur, a collision has likely occurred - establishing a threshold. 'Expected usage' is a runner storing the mobile-device in either their pocket, arm-band or backpack, and using the device during various running speeds (some more vigorous than others), on various path types, bypassing obstacles.

The chosen approach must minimise false-negatives and false-positives. Context; false-positives occur when it is determined an accident has occurred, but there has not be an accident. False-negatives occur when no accident has been determined, but there has been an accident. False-negatives would have the greatest consequences as a runner may genuinely be injured and potentially unconscious but emergency services are not notified. False-positives have a larger financial consequences as rescue services may be dispatched unnecessarily.

Accelerometers alone were used as the sensor of choice because previous works have had successes with accelerometer usage alone. Though gyroscopes and linear acceleration sensors would be useful (some of which are virtual and not hardware based), the sensors are not available to entry-level smart-devices, even if the device has been released recently as the sensors are more expensive than accelerometers.

The three-axis accelerometer system described in 1 is problematic because evaluating values across individual axes cannot be easily isolated, and a mobile-device in expected use will seldom have forces invoked solely on one axis. Therefore; the magnitude of the vector opposed to analysing forces upon individual axes is calculated. In addition; a low-pass filter was implemented to remove the effects of gravity (as recommended and outlined by Android in their developer's sensor documentation). A low-pass filter and a magnitude value negates having to establish the orientation of the mobile-device for each accelerometer vector, and establishing an axis-dependent collision algorithm. In addition; if gravity is not filtered it will exaggerate magnitude values; this won't affect collision detection between it is relative but gravity filtering (or non-filtering) must be consistently applied.

$$\sqrt{(x)^2 + (y)^2 + (z)^2}$$

The aim of establishing an accident threshold requires the data being acquired in an environment emulating use-cases of the mobile-device would undergo when in use by a runner as close as possible. This requires considering the physical environment and physical events a runner may experience, as running and trail-running is a physical sport, but is unfortunately not constant as running can be performed almost anywhere. The three environments considered which should cover most variations are roads (which are designed to be as flat and constant as possible), hills, and uneven trail-running like environments which potentially include climbing or jumps (an obstacle-course of sorts).

Extending this; it is necessary to consider how a runner would realistically keep a mobile-device with them during the duration of a run. Mentioned earlier, these locations are likely to be in a runner's arm-strap, a runners pocket, or a runners backpack. These are also listed in order of how much movement the mobile phone may experience. Though a mobile-device stored in an arm-strap will move through the greatest range-of-motion (ROM), the device is likely to be the most constrained by its container.

3.1 Logging

Log files are created for each trial and magnitude values are logged with a timestamp. Further; a stopwatch was used for each trial to determine the speed of movement.

3.1.1 *Logging Interval*

Magnitudes were initially logged every five seconds but this interval was too long for sprints, there were too few data points as a sprint likely lasts less than a minute. Logging was decreased to one second; this interval was sufficient for most trials but was too large collision events as climbing, jumping, falls etc. last less than a few seconds (climbing over a structure is a longer event but dropping from the structure to the ground or another structure is the former) - the changes in sensor readings during impact periods could not be captured with the original interval. Logging interval was finally decreased to 1/8 of a second.

4. TESTING ENVIRONMENTS

For each environment, consistency is key; a single route was chosen which best emulates the average instance of that environment for a runner, the same distance was traveled for each trial, the same direction, and preferably maintain average speed per trial with minimal deviation. The environments are described below, and any routes indicated on a Google Map has personal identifiable information removed.

4.1 Road

Figure 2: Google Maps screenshot indicating the route and distance of the path approximately followed each road trial

Figure 2 represents the most basic environment a runner would encounter; a flat public road in a neighbourhood. The route is approximately 450 metres (m) and includes running onto and off pavements (road safety).

Figure 3 indicates the route during a sprint, this route is a subset of the road route as a runner will often intermittently sprint shorter distances during their route. Sprints were approximately 62m long (three lamp-posts).

Figure 3: Google Maps screenshot indicating subset of the road route which was sprinted

Figure 4: Google Maps screenshot indicating the route and distance of the path approximately followed each hill trial)

4.1.1 *Hills*

Figure 4 has a constant gradient of 6.9% (a change in elevation of 6.779m) [18] and is approximately 98m long. Moving up the hill and moving down the hill were treated as separate trials.

4.2 Simulated Falls and Obstacles

As discussed in previous works, falls (or car crashes) cannot actually be performed because there is risk for injury and mobile-device could potentially be damaged, thus simulating the collisions as close as possible to the real event without creating risk is necessary [2]. Falls were simulated by falling onto a mattress which was 32 centimetres (cm) thick. There is variation between mattresses; some are softer than others, the springs are old and worn and do not resist force as expected etc. The mattress will act a lot like a car's airbag, it decreases the deceleration rate [10]. The definition of force is;

force = mass x acceleration m/s^2 [25]

If acceleration (negative acceleration) is decreased, force is decreased. Therefore, the more cushioning the mattress provides, the smaller the forces will be exerted, and falling to a hard floor which creates sudden negative acceleration will result larger forces. The various obstacles a trial-runner may encounter during an event or trial was considered; the runner may encounter rocks or walls of some sort which will require climbing over, the runner may be required to jump with most jumps being a small hop over an object in the route but also some larger knee-height objects (or a jump requiring a knee-height jump to satisfy the distance), and falls. A structure was chosen which may emulate a runner climbing over a rock shelf, wall, or boulder.

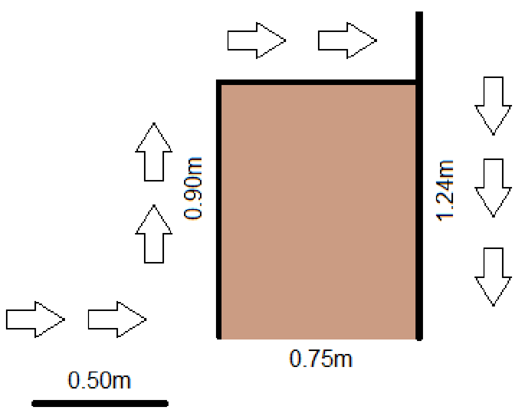

Figure 5: The climbing structure. The direction of movement is indicated by arrows.)

5. RESULTS AND DISCUSSION

Average magnitude readings are established for regular usage; running, walking or sprinting (run-events) with the mobile-device contained in either an arm-band, pocket, or backpack. The upper-bound of deviation should be noted because these magnitudes are most likely to generate false-negatives if there is overlap between simulated falls (fall-events). Following; simulated falls for each expected use-case must be compared i.e. a simulated fall while running with the average runner magnitude values, a simulated fall while sprinting with the average sprinting magnitude values etc. If the fall magnitude values are greater than the average magnitude values during expected usage, a 'greater-than threshold' accident logic can be implemented but this does not negate false-positives; flagging the event as a potential accident and considering the context will be required if this situation occurs. If simulated fall magnitude values are not greater than the average magnitude values, or close, the probability of false-negatives will increase as the simulated fall magnitudes approach the expected magnitude values (and vice-versa). If the latter, a threshold accident logic will not suffice in the context of runner accident collision and either there will need to be a greater focus on context or the algorithm for runner collisions will need to be reconsidered. It is important to note that runners unlike motor-vehicle or bicycle collisions do not implemented suspension and breaking systems which may reduce the variation of magnitude values.

There is likely variation between the various ways to carry the mobile-device. This is expected but it may create interaction between average magnitude values and magnitudes of simulated falls for other run-events which will increase the chance of false-positives. False-negatives are least likely because magnitude values for fall-events would need to less than average values of expected run-type magnitude values. The preceding sentences support the use of context; there may be an average change between magnitude values preceding shared between the run-events or the smallest change is less than the other more disperse and vigorous run-events.

The average magnitude values paired with the standard deviation for these use-cases are the most important as setting a threshold based on these values will satisfy the majority of usage. The average of max magnitude values cannot be ignored though because these spikes may trigger false-positives. Interestingly; greater forces are effected on the mobile-device when the user carries the phone in their pocket but only when the user is walking. This may be that a mobile-device in the user's pocket is experiencing a greater range of motion and the fall of their feet on the path during usage is experienced by the device. Following, the upper body (and a backpack by extension) does not have as wide range of motion and remains in a rather constant position during a walking pace versus running or sprinting. A mobile-device may be more loosely packed in user's pocket where there may even be a factor of swinging involved in their pocket determined by the looseness, fit, and design of their clothing. This looseness may have also effect running and sprinting because an arm-band which is too large may allow shifting of the mobile-device and a backpack is more likely to experience bounce and range of motion under quicker paces.

Consistently the variance increases as the running speed increases; keeping a device in a pocket has highest average magnitude and variance. This trial variation has a average magnitude and standard deviation of 8.251 and 4.331 respectively, the upper magnitude bound thus being 12.582. Almost every other trial variation including one standard deviation to the upper bound does not exceed magnitude of 10. Following the previous definitions of two different approaches of determining an accident, a threshold of approximately 13 could be assigned but referencing the columns of *Max. Magn.* and *Avg. Max. Magn.* in Table 2 and Table 1 there are values greater than that range during usage which may trigger false-positives.

Running on a hill with the device in a pocket has an up and down pattern, this may match with a runner's stride on the given leg which the pocket is located. Running on a hill with pocket which constrains and minimizes the shifting of a mobile device increases the average magnitude values and variance when the run-type changes, the increases due to the speed are more easily isolated from the shifting in the pocket.

Variation Type	Trial Type	Avg. Magn.	Std Dev.	Max. Magn.	Avg. Max Magn.	Avg. km/h
Arm	Walk	2.406	1.053	6.031	5.175	5.69
Arm	Run	7.959	3.559	14.368	13.702	11.31
Arm	Sprint	6.383	4.327	16.385	13.287	18.03
Pocket	Walk	4.686	2.594	14.202	13.248	5.93
Pocket	Run	8.251	4.331	19.800	18.555	10.88
Pocket	Sprint	5.813	5.210	25.686	19.220	18.6
Bag	Run	6.257	3.237	16.943	15.027	10.43
Bag	Sprint	4.299	4.054	20.684	17.142	17.54

Table 1: A table presenting the various magnitude averages and maximum magnitudes experienced during the road running trials

Variation	Trial Type	Up/Down	Avg. Magn.	Std Dev.	Max. Magn.	Avg. Max Magn.	Avg. km/h
Arm	Walk	Up	2.518	1.058	4.486	4.745	5.81
Arm	Walk	Down	2.30	1.487	5.760	6.433	6.03
Arm	Run	Up	6.693	3.326	12.605	13.194	10.98
Arm	Run	Down	6.495	3.456	12.996	18.509	11.34
Pocket	Walk	Up	4.922	2.833	17.830	14.014	6.46
Pocket	Walk	Down	4.899	3.147	17.287	15.803	6.71
Pocket	Run	Up	6.616	4.281	20.011	18.809	11.78
Pocket	Run	Down	6.708	5.059	19.574	20.507	12.20
Bag	Walk	Up	2.920	1.552	6.887	8.804	6.36
Bag	Walk	Down	2.842	2.258	9.392	10.782	6.46
Bag	Run	Up	4.987	3.511	12.992	14.934	10.78
Bag	Run	Down	6.471	3.398	13.355	14.668	10.66

Table 2: A table presenting the various magnitude averages and maximum magnitudes experienced during the hill running trials

Interestingly; pockets had a higher average magnitude value and variance than a backpack, if the mobile is constrained in the backpack and does not have a lot of room to shift, the device is moving through area and does not experience the forces upon the legs during a run.

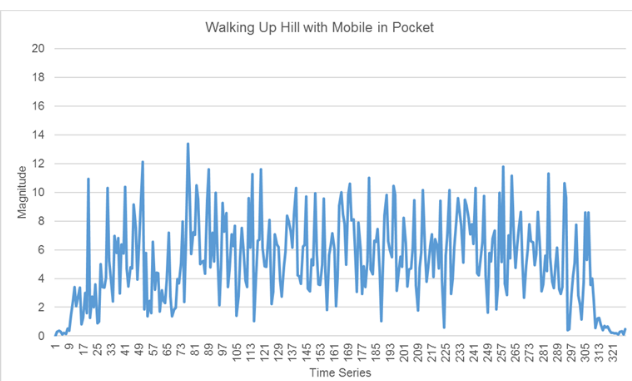

Figure 6: Spikes in magnitude values during an uphill run with mobile-device in user's pocket

Referencing the plots of Figure 7 and Figure 6; even with the walking-pace, there are rhythmic spikes in the magnitude readings, this is likely to be the landing of the user's foot which becomes more pronounced in 7 where the user would be landing with more force each step when they run compared with walking up a hill. The relationship between the magnitude maximums, speed-type, and carry-type become problematic because a static threshold likely cannot

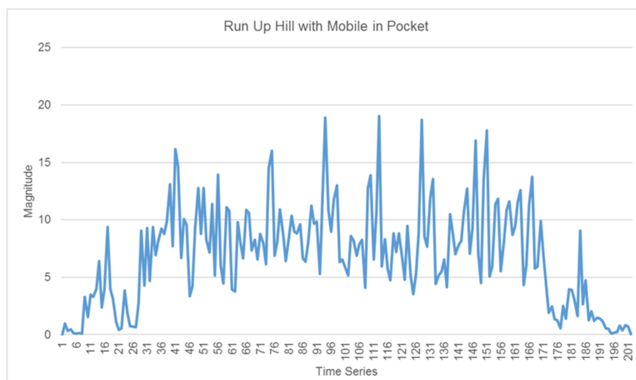

Figure 7: Spikes in magnitude values during an uphill run with mobile-device in user's pocket

be set without incurring a high rate of false-positives.

5.1 Fall/Climbing Events

Events which occur outside of regular usage, but with enough frequency to be considered are the previously mentioned fall or climbing events. During a trial of running with the mobile-device in a non-locking pocket, the mobile-device fell out of the pocket. In addition; arm-bands can fail or slip.

The following table indicates the magnitudes values of various potential false-positive events. Note; not all fields are applicable for each event e.g. mobile-device can only be dropped one way and the events are too quick for a time series. In addition; some data has been excluded because more 'intense' variations of the trial were included and the data does not contribute to the discussion.

Potential false-positive events on average sit within the same ranges of regular-usage and thus are not likely to false-positives. This can be further supported by the trial in which the mobile-device fell out of a pocket during a run.

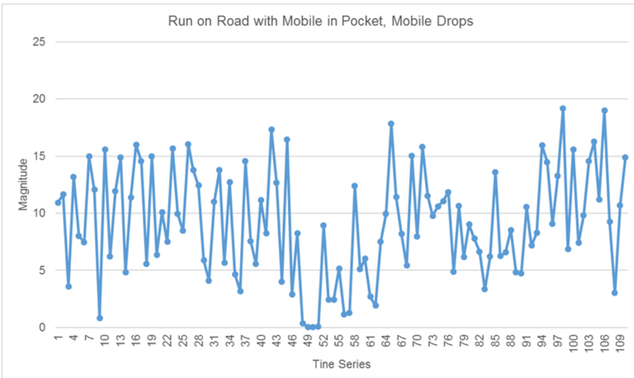

Figure 8: Plot indicating a mobile-device falling out of the user's pocket while running

In the trial indicated by Figure 8 there is a high variance because the mobile-device was loosely constrained by the user's pocket, and as result the mobile-device fell midday of the route. The plot does not indicate this event clearly despite the brief moment of rest while the device was recovered and thus the mobile-device would not determine the drop event as significant among the trial.

Simulated falls are the last required trials. These are events in which a runner has fallen and may injured; these events were simulated by falling onto mattress. If the magnitude values of simulated falls are greater the average and max magnitudes of values during expected usage, a static threshold can be set and and tested on the final route. As the run-types and contain-types for these events are the same as trial run events, the average maximum thresholds will be focused and compared with the respective running trial.

The average maximum magnitudes of fall events, which would be simply a moment in time and thus we can look specifically at the magnitude spikes, are greater than the average magnitude in addition to one standard deviation to the upper-bound of regular usage. As mentioned; there *are* spikes in magnitude during regular usage which may be as a result of; error, variation of clothing, variation in speed, misstep etc - there is a lot more opportunity for inconsistency and variation in a running context compared with motor-vehicle whereby the device may be mounted to the dashboard.

Nevertheless; excluding walking with a mobile-device in an arm-band, the lowest average max magnitude in Table 4 is that of walking with a mobile-device in the user's pocket (the interesting case of walking with mobile-device in pocket as mentioned earlier). This trial variation would be more so indicative of user hiking or a post-run rest and thus the pace is also lower and least liking to injure themselves *if* a fall occurs at this pace. At a slower pace the user have greater opportunity to prevent the biggest impact of the fall with their hands and a slower pace will result in a small force during a fall.

The maximum Avg. Max. Magn. in Table 4 is 12.062, and is greater than the Avg. Magn. presented in 2 and thus a threshold of between 13 and 15 could be set with a likely chance of near-to-zero false-negatives and minimal false-positives.

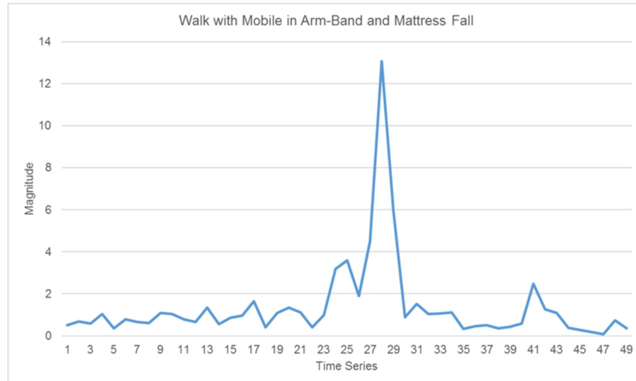

Figure 9: Walking towards and falling onto a mattress, mobile-device strapped to arm

The first figure represents the most ideal form whereby the device is firmly constrained with minimal movement during the user's actions; the sudden changes are most obvious. The proceeding graphs indicate trials whereby the runner was either moving with a greater pace (sprinting) or the mobile-device is likely to be not as constrained.

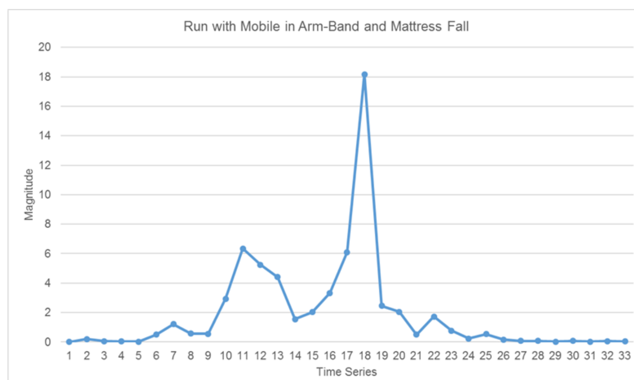

Figure 10: Running and falling onto a mattress using a mobile-phone arm-band

Increasing the speed of the device user increases the average accelerometer magnitude registered (and forced) involved with the user falling to the mattress. The two preceding graphs are the most clear and clean value plots unless a greater smooth filter is used on the series of values. The proceeding graphs represent the increasing looseness of the mobile-device in the runner's bag, pocket and how that along with increasing speed a sprint or even at the same speed of a run will increase the range and deviation of values from each other.

5.2 Testing

A final obstacle course incorporating the discussed events which may generate false-positives, various run-speeds and one fall was designed. The route consists;

Event Type	Contain Type	Avg. Magn.	Std Dev.	Max. Magn.	Avg. Max Magn.
Phone drop				12.900	8.792
Structure climb & jump	Arm	1.344	1.711	12.482	8.944
Structure climb & jump	Pocket	2.340	2.873	24.211	12.504
Structure climb & jump	Bag	2.215	2.437	15.503	12.802
Obstacle jump, 42cm x 42cm	Arm	1.923	2.749	14.760	11.580
Obstacle jump, 42cm x 42cm	Pocket	3.016	3.235	14.344	12.701
Obstacle jump, 42cm x 42cm	Bag	2.231	2.436	18.059	15.121

Table 3: A table presenting the various magnitude averages and maximum magnitudes experienced during events which are not falls, but may trigger a collision event.

Contain Type	Trial Type	Max. Magn.	Avg. Max Magn.
Arm	Walk	13.090	9.464
Arm	Run	20.998	15.040
Arm	Sprint	19.251	15.654
Pocket	Walk	17.297	12.062
Pocket	Run	17.426	15.444
Pocket	Sprint	21.158	17.348
Bag	Walk	19.167	12.292
Bag	Run	18.430	14.739
Bag	Sprint	21.385	17.115

Table 4: A table presenting the various magnitude averages and maximum magnitudes experienced during simulated falls

1. 50m run from the start on a flat road
2. Another 10m before the user jumps over an obstacle and continues running
3. User sprints approximately 30m with a 1.5m uphill
4. Runs approximately 10m on uneven Earthy ground
5. User climbs over the structure in Figure 5
6. User runs approximately another 3m until falling to a mattress

This course will be run while having the accident threshold set to the value of 15, as established in the previous section.

The previously mentioned variance and inconsistency between contain-types reappeared in the final obstacle course. A threshold of 15 was able to detect the eventual fall to a mattress in all trial runs except one trial where the mobile-device was contained in an arm-band. False-negatives have the greatest real repercussions because the user may be injured. In addition; multiple false-positives were detected during each trial with bag contain-type haven't the least. This may be due to that a correct fitting backpack with tight straps (including a strap across the chest which is common among numbers) will essentially have the same forces exerted on it as a runner's upper-body would, they would effectively be moving as much, and at more vigorous speed-types the mobile-device will have the shortest ROM (if the device is tightly packed within the backpack with minimal shifting or movement within the backpack).

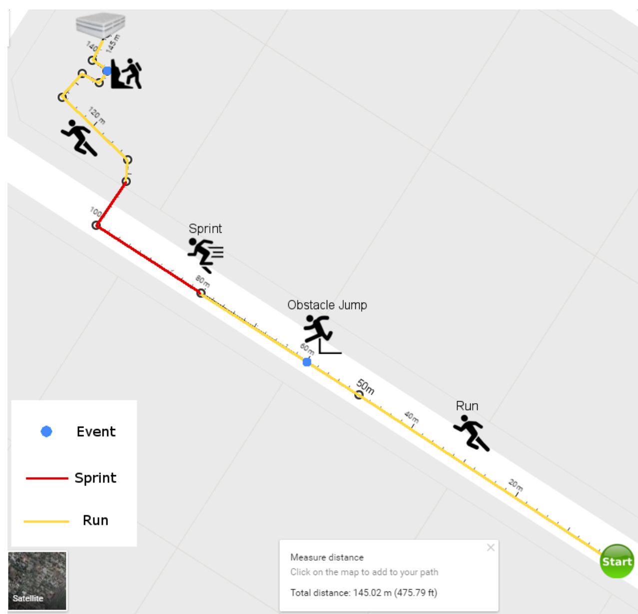

Figure 11: A depiction of the final obstacle course

6. CONCLUSIONS AND FUTURE WORK

MOVER Runner aimed to add to the two fields of repurposed smart-phones, specifically Android, and collision detection. Previous works have had success with using a static threshold, using accelerometer values, and an accident is likely to have occurred once the threshold is breached. Establishing a potential threshold required recording accelerometer values while performing runs carrying the smartphone the various ways a runner would (in an arm-band, pocket, or backpack) and at various speeds (walking, running, and sprinting). Having established averages and upper-bounds, events which may trigger a suspected accident but are not had to be addressed. Potential false-positives included runners performing various jumps, climbing over struc-

Contain Type	Detected Falls	Avg. False-Positives
Arm	4/5	7
Pocket	5/5	15.8
Bag	5/5	2

Table 5: A table presenting the results of the final obstacle-course

tures, and dropping a smart-phone. Variation was present; variables like the freedom of movement of the device within its container, running style, and the movement of the device container e.g. a non-tightly secured backpack to the runner's body, introduced value spikes and inconsistencies. Finally; the average accelerometer values of emulated falls, at various running speeds, were recorded. Comparing the two groups of data indicated the average values of emulated falls at a pace greater than walking was greater than the upper-bounds of values recorded from expected usage. Though there were spikes in the recorded values of expected usage which were greater than the average max values of emulated falls, the spikes are inconsistent and difficult to compensate or negate.

A magnitude threshold of 15 was able to detect of a total of 14/15 fall events, 93.3%, but the instances of false-positives were high, specifically when the smart-phone was carried in a pocket. This was discussed earlier in that carrying the mobile-device in a pocket will result in more potential for variation as pace increases or runner's stride varies.

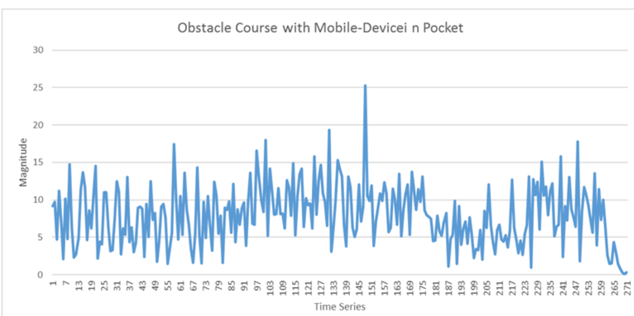

Figure 12: A final run on the obstacle course

Running lacks the consistency between 'models' (runners) and the dampening of suspension systems. Detecting runner falls is possible but the final obstacle course data supports the notion that running contains a lot more variation and inconsistency than a motor-vehicle would provide and establishing a threshold which will apply to all runners, dress styles, and physical running operations is difficult. The threshold value should be reevaluated and potentially adjusted upwards as a fall to a less cushioned mattress (or the ground) would be expected to generate impacts forces upon the runner and thus the mobile-device. This would be increase the fall magnitudes while the previously established averages and magnitudes would be unchanged (excluding their natural variances which are experienced due the wide variation among runners). The number of false-positives would decrease without increasing false-negatives (requiring testing for confirmation) and without having to implement additional algorithmic considerations. Additional algorithm features would be a focus on context of a large increase in magnitude i.e. this is would assist identifying falls in the trials like walking which has a lower level and lots of interaction with values generated by quicker paces as there are greater forces generated. Specialization of contain-type could allow establishing more detailed graph forms and plot values, this would require the user choosing the contain-type of their mobile-device before starting the application. This could allow estimating the speed-type of the runner by the variance of magnitude values (as the variances and expected ranges could be established for each contain-type and the variations of run, walk, and sprint). Extending the previous idea, implementing a learning algorithm whereby the an individual user's running style is profiled and applied to their user account's magnitude values during usage to reduce the potential false-positives simply due to a runner being heavier, having a longer stride, or a lumbering and heavy-footed running style.

7. REFERENCES

[1] N. H. T. S. Administration. Vehicle database query results - latest 10 test. http://www-nrd.nhtsa.dot.gov/database/VSR/veh/LatestTestInfo.aspx, 2016. Accessed: 2016-11-06.

[2] F. Aloul, I. Zualkernan, R. Abu-Salma, H. Al-Ali, and M. Al-Merri. ibump: Smartphone application to detect car accidents. In *Industrial Automation, Information and Communications Technology (IAICT), 2014 International Conference on*, pages 52–56, 2014. ID: 1.

[3] S. Asongu. The impact of mobile phone penetration on african inequality. *International Journal of Social Economics*, 42(8):706–716, 2015. http://dx.doi.org/10.1108/IJSE-11-2012-0228.

[4] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan. Mobile phone based drunk driving detection. In *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*, pages 1–8, 2010. ID: 1.

[5] V. Douangphachanh and H. Oneyama. Formulation of a simple model to estimate road surface roughness condition from android smartphone sensors. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6, 2014. ID: 1.

[6] X. Ducrohet. Android 2.3 platform and updated sdk tools. http://android-developers.blogspot.co.za/2010/12/android-23-platform-and-updated-sdk.html, December 2010. Accessed: 2016-10-02.

[7] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. 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*, MobiSys '08, pages 29–39, New York, NY, USA, 2008. ACM.

[8] J. Etheridge. Four to appear on charges linked to fatal Tokai Forest attack. http://www.news24.com/SouthAfrica/News/

four-to-appear-on-charges-linked-to-fatal-tokai-forest-attack-20160331, March 2016. Accessed: 2016-10-02.

[9] A. Ghose, P. Biswas, C. Bhaumik, M. Sharma, A. Pal, and A. Jha. Road condition monitoring and alert application: Using in-vehicle smartphone as internet-connected sensor. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 489–491, 2012. ID: 1.

[10] J. P. Huffman. The physics of: Airbags. *Car and Driver. http://www. caranddriver. com/features/the-physics-of-airbags-feature*, 2011.

[11] G. Inc. Sensors overview. https://developer.android.com/guide/topics/sensors/sensors_overview.html. Accessed: 2016-10-02.

[12] G. Inc. Welcome to the Android Open Source Project! https://source.android.com/. Accessed: 2016-10-02.

[13] G. Inc. Dashboards. https://developer.android.com/about/dashboards/index.html, September 2016. Accessed: 2016-10-02.

[14] A. Industries. ICEdot Crash Sensor Teardown. https://www.youtube.com/watch?v=PjzeCkkP5w0, September 2013. Accessed: 2016-10-02.

[15] D. H. P. Kang, M. Chen, and O. A. Ogunseitan. Potential environmental and human health impacts of rechargeable lithium batteries in electronic waste. *Environmental Science & Technology*, 47(10):5495–5503, 2013. PMID: 23638841; http://dx.doi.org/10.1021/es400614y.

[16] J. Lahn, H. Peter, and P. Braun. Car crash detection on smartphones. In *Proceedings of the 2Nd International Workshop on Sensor-based Activity Recognition and Interaction*, WOAR '15, pages 12:1–12:4, New York, NY, USA, 2015. ACM.

[17] K. Legg and N. Mposo. UCT student dies in fall on mountain. http://www.bloodhoundssc.com/forces, August 2013. Accessed: 2016-10-02.

[18] D. Logic. Google Maps Find Altitude. https://www.daftlogic.com/sandbox-google-maps-find-altitude.htm, October 2016. Accessed: 2016-10-02.

[19] MyBroadband. Android crushing its competitors in South Africa's mobile race. http://mybroadband.co.za/news/smartphones/127040-android-crushing-its-competitors-in-south-africas-mobile-race.html, March 2015. Accessed: 2016-10-02.

[20] S. Raj, A. Jain, and R. Misra. Smartphone sensing for large data set collection of potholes. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, pages 517–518, New York, NY, USA, 2013. ACM.

[21] I. I. Research. Smartphone OS Market Share, 2016 Q2. http://www.idc.com/prodserv/smartphone-os-market-share.jsp, 2016.

[22] I.-M. L. Roux. CT woman raped, held captive for over 24 hours in Newlands Forest. http://ewn.co.za/2016/03/10/CT-woman-28-raped-in-Newlands-Forest, 2016 2016. Accessed: 2016-10-02.

[23] K. Singh. Runner rescued from KZN's kloof gorge after falling down cliff. http://www.news24.com/SouthAfrica/News/runner-rescued-from-kzns-kloof-gorge-after-falling-down-cliff-20160614, June2016 2016. Accessed: 2016-10-02.

[24] F. Sposaro and G. Tyson. ifall: An android application for fall monitoring and response. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6119–6122, 2009. ID: 1.

[25] B. UK. Forces and motion. http://www.bbc.co.uk/schools/gcsebitesize/science/add_gateway_pre_2011/forces/motionrev2.shtml. Accessed: 2016-11-06.

[26] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt. WreckWatch: Automatic Traffic Accident Detection and Notification with Smartphones. *Mobile Networks and Applications*, 16(3):285–303, 2011.

[27] J. Zaldivar, C. T. Calafate, J. C. Cano, and P. Manzoni. Providing accident detection in vehicular networks through obd-ii devices and android-based smartphones. In *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pages 813–819, 2011. ID: 1.

8. APPENDIX

All data recorded during trails can be viewed and downloaded from Google Drive:
https://goo.gl/idq9zG