Sacred Heart University

From the SelectedWorks of Tolga Kaya

2015

From Pit to Long Lie: A Falldetection Algorithm for Smart Phones

J.A. Pulcifer-Stump Steven Kettler J.L. Koch Patrick Seeling Dr. Tolga Kaya, *Sacred Heart University*



Available at: https://works.bepress.com/tolga-kaya/12/

FROM PIT TO LONG LIE: A FALL-DETECTION ALGORITHM FOR SMARTPHONES

Jeffry A. Pulcifer-Stump, Central Michigan University; Patrick Seeling, Central Michigan University; Janet L. Koch, Garden City High School; Steven Kettler, Alma High School; Tolga Kaya, Central Michigan University

Abstract

The elderly population of the world continues to grow and, therefore, creates a need for heightened safety measures. Smartphones have been broadly adopted throughout age categories and mobile applications are extremely useful during all types of daily activities; popular applications today not only target recreational uses, but also health tracking and awareness. As men and women over the age of 65 continue to lead active lives, a smartphone application that can detect falling incidents would be very useful. In this research project, a fall-detection application was developed that utilized the acceleration sensor embedded in most mobile devices. The application continuously monitored a person's movement and checked multiple threshold points for a fall impact. Upon impact, phone settings were utilized to communicate with a contact for immediate assistance. The algorithm was successfully implemented, based on the Android platform and its accuracy was tested with eight subjects, who performed 135 fall experiments overall. The measured sensitivity was calculated to be 92% and the measured specificity was 100%.

Introduction and Related Works

The life expectancy of the elderly population, specifically regarding those over the age of 65, continues to increase, thanks to better healthcare technology and available treatment options [1]. The median age increased significantly from 29.5 to 37.2 in the time period from 1960 to 2010, resulting in a 13% increase of the overall elderly population in the U.S. [2]. Therefore, different challenges for the elderly population were introduced to health professionals; these are commonly related to fatigue and incidents in addition to particular illnesses [1], [3], [4]. Among these identified incidents, falls represent one of the most frequent cases that challenges the elderly population. Although it is difficult to define a specific fall event and collect the fall-related reports, various studies have shown that almost 30% of elderly persons fall at least once a year [1], [3], a number that corresponds to more than 10 million elderly in the U.S., who would experience a fall incident within a year [2]. The consequences from falling can be fatal in extreme cases or can lead to serious health problems [1], [3]. Thus, several studies have focused on the causes, consequences, and prevention methods of falls in the elderly population [5-7].

On the other hand, fall detection methods have been explored increasingly during the last two decades. Although several techniques are commonly used to detect a fall incident, studies are often classified into two main groups: accelerometer/gyroscope-based and video-based [8-11]. While accelerometer-based sensors typically have to be carried by the subject on certain locations of the body, such as one's waist, knees, or head, video-based systems suffer from complicated installation procedures and privacy issues that stem from monitoring [8-12]. In this paper, the authors focus on the application of accelerometer-based sensors, which rely on fluctuations in tilt and motion to detect a fall incident.

The majority of accelerometer-based fall detectors rely on 2D or 3D accelerometers and/or gyroscopes. In most cases, the location of the sensing device determines the accuracy of the system. Typically, multiple sensors in different regions of the body were employed in past studies and threshold-based fall detection algorithms were utilized [11], [13-15]. However, these prior works relied on the sensor to be fixed on the waist, which limited the practical use of the system. Commercial sensor boards were also implemented as wearable fall and posture detectors [16], [17]. Although all these systems reached high sensitivity and specificity levels, they either utilized more than one worn sensor or a limited, fixed location of the sensor itself.

Healthcare companies have also been working on efficient and affordable solutions that can help with a fall incident. Most of the products are simply push-button-activated emergency call systems in which users carry a wirelessenabled button, usually worn as a wristband, pendant, or belt clip, requiring button activation when the fall occurs. This signal simply activates a phone call to the healthcare unit and a 2-way telephone communication is established [18-20]. Sensitive floor mats have also been used to track a person and determine if he/she reached the bedroom door while traveling from the bed [21]. Some of the most common systems currently on the market are not phone-based applications and, hence, require a separate sensing system and charge a monthly fee. For example, The Brickhouse Alert Fall Monitoring System relies on a CST (Custom Sensors and Technology) sensing system with 24-hr service. Similarly, Link to Life Fall Detector, CST-L TL, and Prime Medical Alert Amber Select use the same technology [22]. [23]. Each relies on a tilt sensor to detect a fall and has a device attached to the person, most commonly a pendantstyle device. One of the most complete devices is Halo Monitoring, which uses an accelerometer sensor similar to this smartphone application (App). It is worn on either the chest or on a belt clip and tracks vital signs, sleeping patterns, and activity levels. Caregivers are notified by user preference but the systems require a landline or computer for tracking.

While there are successful prototypes and products available today, almost all of the systems require significant attention by the user and feature rigorous installation procedures. An affordable and easy-to-use point-of-care solution for fall detection and notification has attracted significant interest. With recent advances in smartphone technology, researchers have been taking advantage of using embedded sensors that are readily available in the phone and have been developing algorithms for fall detection [24-27]. However, most of the algorithms currently require that the phone be kept in a certain orientation. Smartphone Apps such as T3Lab Fall Detection and Fade are free and utilize the accelerometer sensor built into the phone. The T3Lab App, which continuously runs in the background of the smartphone, allows an alarm to sound when a fall is detected. However, the App is very sensitive and can easily be false-triggered. The Fade App requires the user to turn it on and off and has a recovery feature that allows for movement beyond a potential fall impact.

With an aging population that features increasing numbers of adults over the age of 65 that remain independently living in their homes, the need for reliable fall detection is apparent. The convenience of a fall-detection App downloaded to one's smartphone creates a sense of security, as daily activities are carried out without the need for any extra equipment. In this paper, the authors propose an Androidbased smartphone as a platform for fall detection. The proposed algorithm is based on multiple thresholds including free fall, impact, orientation cross, and a long lie. The algorithm is able to successfully detect an occurring fall, regardless of the device's orientation and placement. The proposed approach enables reliable and accurate fall incident monitoring through the utilization of smartphone applications by the elderly, which allows them to live independently in their own homes. Furthermore, the authors focus on the materials and methods of the fall-detection algorithm and test protocol.

Fall Detection Algorithm

The proposed algorithm was developed and tested using an HTC 4G LTE smartphone, which utilizes the Android 4.1.1 ("Jelly Bean") operating system. The phone itself represents the current mid-level range of smartphones available to users, with an approximate price of \$200 with carrier subventions. The program's offline data storage system allows gathering of data in real time during program execution, with data processing available at a later time. A more in-depth description of the general Android application environment, which also features a simple interface suitable for a target population group and safeguards for real-life scenarios (e.g., setting a personal calling number instead of 911) can be found in the study by Oner et al. [27].

Acceleration data were collected using the device's embedded accelerometer sensor in the fastest sampling mode, which was 50 samples per second (i.e., one sample every 20 ms). The overall geometric average of the acceleration points, A, in each direction was calculated using Equation (1):

$$A = \sqrt{A_x^2 + A_y^2 + A_z^2}$$
(1)

where,

 A_x = acceleration in the -x direction in m/s² A_y = acceleration in the -y direction in m/s² A_z = acceleration in the -z direction in m/s²

An average was calculated for every 25 data points (approximately 500 ms). As a new data point was gathered, the oldest data point was discarded and a new average was calculated, implemented as a moving average. For example, at a point in time, where the data(i) was recorded, averaging for data(i-24) to data(i) was performed. The subsequent average calculation starting with the new data point i+1 would, hence, consist of the window of values from data(i-23) to data(i+1).

Increasing the window size would further smooth the data, but alter the information needed in order to detect the fall with the proposed algorithm. In addition to the sliding window that was used by the fall detection algorithm, a second long-term average was created to increase the algorithm's accuracy. This average value of the overall data was generated and reset every 30 seconds (approximately every 1500 samples).

The fall detection algorithm itself was primarily based on three different thresholds that were evaluated successively in order to identify different fall characteristics, as illustrated in Figure 1. These thresholds were based on the magnitude of changes found in the data over a fixed time period, in combination with phone orientation changes. As an additional precaution, the algorithm also evaluated the impacts that would indicate a fall (i.e., hitting a surface trailed by a period of lying). It should be noted that several of the threshold values that were part of the algorithm were experimentally derived.



Figure 1. Proposed Algorithm

The initially filtered data were first evaluated against the long-term average data for a continuous decline, which was termed as the "Pit." The Pit was set when the initially filtered data were 3% below the moving average data. The pit threshold was reached if the real-time data went below 75% of the pit's value, which was the first indication of the fall. While checking for this threshold, an assessment of whether the overall average had increased over the overall average at the time the pit was set within one second from the time that the pit threshold was triggered. The algorithm checked for the impact that constituted a typical fall, once both previous triggers had been set. As soon as the person hit the floor surface, acceleration reached a maximum value, due to the impact. If this impact's reported short-term moving average data were larger than 1.925 times the overall average and within 1.2 seconds after the pit threshold was triggered, the algorithm would trigger the Peak detection switch (see Figure 2). These values were experimentally derived and found to be accurate, as described in the results section.

Upon the trigger of the pit being set, a statistical analysis was performed. Once the data were below 75% of the overall average, the first point, P1, was stored. Then, the next point, P2, was stored when the data exceeded 75% of the data. Once both points were set and the data were greater than the overall average, the standard deviation and the percentage of data between the two stored points was calculated. The standard deviation was calculated using Equation (2):

$$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2}$$
(2)

where,

N = sample size

- x_i = the *i*th data value from the combined acceleration value
- x = the mean of the data gathered between the pit set and pit release points



Figure 2. A Representative Fall Event

The percent of data to each point is calculated as the statistical data if it reaches a critical value of 0.12, which triggers the statistical switch. If the algorithm detects an impact, where the statistical threshold is exceeded, a fall detection signal is generated. Another way of ensuring that the fall is detected is to evaluate the orientation of the phone or the sharpness of the free-fall signal. If a person falls, the phone would most likely be pointing in the z direction. The orientation of the phone would be checked to determine if the acceleration in the z direction is close to 1 g, which would trigger the orientation switch. On the other hand, the derivative of the signal is calculated after the pit is set. If the derivative of the acceleration in the z- direction, $\partial A_z/\partial t$, is greater than 11 times the amount of the overall average, the rate switch is triggered.

If an impact after falling is identified or the data are determined to be close to the threshold, the algorithm evaluates whether a long-lie case is detectable. The data are considered close to the upper threshold if it rises above 1.75 times the overall average. In case of a longer time period after a fall, where a person lies, the acceleration magnitude becomes relatively stagnant (within 10% of a deviation from the overall average) for at least four seconds from the time of when the data breach the impact threshold or are close to that threshold (though other time periods could be facilitated in a straightforward manner). During the four seconds of evaluating the data, if the amount of data outside the 10% deviation is less than 20% then the long-lie trigger is activated. If the algorithm detects a time period of stagnant data, indicating a long lie, the orientation, and the rate of change correlates to a fall incident then a fall detection signal is generated (see Figure 2).

An idealistic representation of the typical data obtained and a representative sample of real-world data are illustrated in Figure 2. This test was performed, where the subject threw himself on a low bed, while the phone was in his pants' pocket. The overall average and the moving average data were plotted. Corresponding threshold points were marked as the pit threshold, peak (impact), and long lie. The subset shows the axes crossing. A recovery algorithm was also implemented. Once the fall was detected, and if the original data stayed within 15% of the moving average, the subject was considered to be lying on the floor. However, if more than 35% of the data points for a certain time interval (the default was set to five seconds) were above or below the 15% range then the recovery switch was turned on. Depending on user preference, a fall with a recovery could be reported immediately or saved for future diagnosis.

Figure 3 shows a snapshot of the Android App that was used for the tests. Two sensors were implemented in the algorithm: a fall sensor and a pedometer. The pedometer was used to test the functionality of the phone's embedded accelerometer [27]. Each threshold was prompted at the App screen to monitor the program's progress. When a fall was detected, the App gave a pop-up message that a fall had been detected. The App would be simplified significantly for the end-user.



Figure 3. A Snapshot of the Application at the Time of the Detection of a Fall Event

Human Subject Tests

It was a big challenge to evaluate a fall detector in realworld settings. Since it is ethically wrong to perform tests on elderly subjects, most studies focused on younger adults for experimental trials. Furthermore, there are no standard test scenarios that would be reasonable approximations for real-life simulations. Thankfully, some research groups have focused on developing test standards and evaluation metrics taking these considerations into account. Noury et al. [12], [28] developed a good set of scenarios that would generate both positive and negative fall events.

Simulated real falls were also studied to evaluate different algorithms in terms of the sensitivity and specificity settings [29], [30]. Using these definitions, the specificity (ability to detect only a fall) and the sensitivity (ability to detect a fall) of the fall detection device were evaluated. In evaluating falls, there could be four different decisions that could occur [12], [28]:

- 1. True positive (*TP*): A fall event occurs and the device detects it.
- 2. False positive (*FP*): No fall occurs but the device gives a fall detection signal.
- 3. True negative (*TN*): No fall occurs and the device does not give a fall detection signal.
- 4. False Negative (*FN*): A fall event occurs but the device does not detect it.

Sensitivity and specificity can then be defined as TN/(TN+FP) and TP/(TP+FN), respectively [12], [28]. Test scenarios adapted from the work by Noury et al. [12], [28] were used to assess the proposed fall detection system's accuracy. The following scenarios were expected to trigger a fall event:

- 1. Backward fall
 - a) ending sitting
 - b) ending lying
 - c) ending in a lateral position
- 2. Forward fall
 - a) on the knees
 - b) with forward arm protection
 - c) ending lying flat
 - d) with rotation, ending in the lateral right position
 - e) with rotation, ending in the lateral left position
- 3. Lateral fall
 - a) ending lying flat to the right
 - b) ending lying flat to the left
- 4. Real-world backward fall
 - a) assisted falling without letting the patient know the exact timing

The following scenarios were not expected to trigger a fall event:

- 1. Syncope
 - a) a vertical slipping against a wall finishing in a sitting position
- 2. Neutral
 - a) sit down on a chair then stand
 - b) lie down on the bed then rise up
 - c) walk a few meters
 - d) bend down, catch something on the floor then rise up
 - e) cough or sneeze

Real-world backward fall tests were performed using the method implemented by Klenk et al. [30]. In this test, subjects were held inclined backwards about 30 to 40 degrees and allowed to fall onto a mattress on the floor with the instruction of "try not to fall." Subjects did not know when the fall would be initiated. This way, real-world backward fall situations were simulated and the results were used to optimize the fall detection algorithm. Subjects were chosen from healthy adults (a health survey was conducted) ranging from 20 to 40 years old. Consent forms were given to the subjects and experiments were initiated after they signed. The University's Institutional Review Board (IRB) approved the experimental procedures. Each fall scenario was first demonstrated and then the subjects were asked to perform those scenarios by putting the phone into their front pants' pocket. It should be noted that placement of the phone in different locations could yield different results. Real-world backward fall tests were only performed with the subjects that felt comfortable executing the scenario. Each subject was asked to perform the 11 positive and six negative fall executions described above. Data from each fall were recorded at the event and subsequently analyzed.

Eight subjects (six male and two female) evaluated the fall scenarios. The average age of subjects was 26.9 years with a standard deviation of 6.7. The average body weight of the subjects was 162.3 lb. with a standard deviation of 30.5. A total of 87 fall and 48 no-fall events were recorded. A set of fall event examples are given in Figure 4. More than 90% of falls were detected with the trigger of upper and lower thresholds and the statistical switches. Each data collection session started with a few seconds of preparation (clicking the button to start recording, putting the phone into the pocket, and waiting for a few seconds) and was followed by the execution of the event. Exemplary data captured from backward, forward, and lateral fall events are shown in Figures 4(a), 4(b), and 4(c), respectively. Fluctuations after the fall were due to the bouncing from the bed mattress that was positioned on a carpeted floor in order to limit the impact of the falls, as required by the experimental protocol that was approved by the IRB. These fluctuations were more prominent for forward and lateral falls. The steady portion of the data right after the fall was the long lie, where the subjects lie on the mattress for a few seconds. Figure 4 (d) shows the data for bending, where a decline in the moving average occurs that triggers the pit threshold. However, the impact threshold was not exceeded. Similarly, sitting down did not trigger the impact threshold, even though the pit threshold was set; see Figure 4(e). The walking event had the periodical dips and hills that could be associated with pit threshold and impact switched; see Figure 4(f). However, since the impact was not high enough or the moving average did not go below a certain threshold, fall events were not recorded.



Figure 4. A Representative Fall Event

A summary of the success rate for the tests is given in Table 1. The overall sensitivity was calculated as 92% and the specificity was 100%. Sensitivity is defined as the ability to detect a fall, whereas specificity is defined as only detecting a fall [12], [28]. Therefore, specificity can only be calculated for no-fall events. On the other hand, sensitivity was calculated for actual fall events. Table 1 shows "N/D" for undefined cases.

Table 1.	Success	Rates
----------	---------	-------

Tests	Success rate		
	Sensitivity	Specificity	
Backwards	94%	N/D	
Forward	90%	N/D	
Lateral	88%	N/D	
Bending	N/D	100%	
Walking	N/D	100%	
Sitting	N/D	100%	
Lying	N/D	100%	

While being able to avoid false positives would be the result of using a smartphone during activities of a person's daily life, some falls were not detected by the algorithm. Overall, a contributing factor was the mattress that was required by the IRB to limit the impact of the experimental falls for the participants. While softening the fall for the volunteers, it also limited the maximum values that were obtained in the data gathering process from the accelerometer. Secondly, the volunteers themselves likely altered their "natural" fall behavior through event anticipation and appropriate precautious activity as well as for conducting multiple experiments in a single setting. For the lateral fall scenarios, for example the lowest detection rate, the volunteers likely cushioned their fall before turning to their sides, which could have had a significant impact on the sensor readings obtained. It is highly anticipated that employing the algorithm in future studies with more volunteers will significantly increase the already high success rate.

Conclusions and Future Works

In this study, a new fall detection algorithm was proposed that was implemented on an Android-based smartphone. Through various experiments, the algorithm was found to result in a sensitivity of 92% and a specificity of 100%. The algorithm was, in turn, very successful at not detecting false falls that could be derived from daily activities (such as sitting, walking, and bending). On the other hand, a high rate of success was achieved in the detection of real falls, while undetected falls were only the result of experimental limitations. As smart sensors and phone-enabled technologies are on the rise, the authors believe that this proposed work is timely and relevant [31-33]. The authors also believe that the algorithm presented will contribute to the use of smartphones as medical monitoring devices, specifically fall events. Current commercial devices are either very expensive or require a response from the patient in order to invoke the alarm system. Although a simple button can serve this purpose, in the case of a user of these systems losing consciousness or not being willing to seek help, the system will not be functional. Therefore, having a smartphone in their pockets can continuously monitor their movements and the response method can easily be adjusted on the phone. Response methods considered were to call 911 automatically, send a text, email, or initiate a phone call to either a local health unit or a relative.

Due to ethical and practical restrictions, the experiment was limited to a younger population than ultimately targeted. The distribution of the developed algorithm in the form of an Android App will be part of future works. The distributed App will afford crowd-source human subject trials without restrictions and provide further data for refinements of the algorithm. Other future works are directed to the evaluation of different phone positions and daily life scenarios, which could trigger a detected fall.

References

- [1] Tideiksaar, R. (1997). Falling in Old Age. Springer.
- [2] United States Census Bureau, 2010 Census Briefs (2011). Retrieved September 11, 2014, from http:// www.census.gov/prod/cen2010/briefs/c2010br-03.pdf
- [3] Wild, D., Nayak, U. S., & Isaacs, B. (1981). How Dangerous Are Falls in Old People at Home? *British Medical Journal (Clinical research ed.)*, 282(6260), 266.
- [4] Tinetti, M. E., Speechley, M., & Ginter, S. F. (1988). Risk Factors for Falls among Elderly Persons Living in the Community. *New England Journal of Medicine*, 319(26), 1701-1707.
- [5] Kiel, D. P. (1991). Falls. *Rhode Island Medical Jour*nal, 74(2), 75-79.
- [6] Salvà, A., Bolíbar, I., Pera, G., & Arias, C. (2004). Incidence and Consequences of Falls among Elderly People Living in the Community. *Medicina Clinica*, *122*(5), 172-176.
- [7] Ryynänen, O. P., Kivelä, S. L., Honkanen, R., & Laippala, P. (1991). Falls and Lying Helpless in the Elderly. *Zeitschrift fur Gerontologie*, 25(4), 278-282.
- [8] Luštrek, M., & Kaluža, B. (2009). Fall Detection and Activity Recognition with Machine Learning. *Informatica*, 33(2), 197-204.
- [9] Auvinet, E., Multon, F., Saint-Arnaud, A., Rousseau, J., & Meunier, J. (2011). Fall Detection with Multiple Cameras: An Occlusion-resistant Method Based on 3 -d Silhouette Vertical Distribution. *IEEE Transactions on Information Technology in Biomedicine*, 15 (2), 290-300.
- [10] Sixsmith, A., & Johnson, N. (2004). A Smart Sensor

to Detect the Falls of the Elderly. *IEEE Pervasive Computing*, *3*(2), 42-47.

- [11] Bourke, A. K., O'Brien, J. V., & Lyons, G. M. (2007). Evaluation of a Threshold-based Tri-axial Accelerometer Fall Detection Algorithm. *Gait & Posture*, 26(2), 194-199.
- [12] Noury, N., Fleury, A., Rumeau, P., Bourke, A. K., Laighin, G. O., Rialle, V., et al. (2007). Fall Detection-Principles and Methods. 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, (pp. 1663-1666).
- [13] Bourke, A. K., & Lyons, G. M. (2008). A Thresholdbased Fall-detection Algorithm using a Bi-axial Gyroscope Sensor. *Medical Engineering & Physics*, 30 (1), 84-90.
- [14] Kangas, M., Vikman, I., Wiklander, J., Lindgren, P., Nyberg, L., & Jämsä, T. (2009). Sensitivity and Specificity of Fall Detection in People Aged 40 Years and Over. *Gait & Posture*, 29(4), 571-574.
- [15] Kangas, M., Konttila, A., Lindgren, P., Winblad, I., & Jämsä, T. (2008). Comparison of Low-complexity Fall Detection Algorithms for Body Attached Accelerometers. *Gait & Posture*, 28(2), 285-291.
- [16] Chen, J., Kwong, K., Chang, D., Luk, J., & Bajcsy, R. (2006). Wearable Sensors for Reliable Fall Detection. 27th Annual International Conference of the Engineering in Medicine and Biology Society, (pp. 3551-3554).
- [17] Li, Q., Stankovic, J. A., Hanson, M. A., Barth, A. T., Lach, J., & Zhou, G. (2009). Accurate, Fast Fall Detection using Gyroscopes and Accelerometer-derived Posture Information. *Sixth International Workshop* on Wearable and Implantable Body Sensor Networks, (pp. 138-143).
- [18] Brickhouse Alert Home Systems. (n.d.). Fall Monitoring and Panic Button System. Retrieved September 11, 2014, from http://www.brickhousealert.com/ personal-emergency-medical-alarm.html
- [19] Alert 1 Medical Alert Systems (n.d.). Retrieved September 11, 2014, from http://www.alert-1.com/ system/medical-alert-system/504
- [20] Vivago. (n.d.). Alarm Buttons. Retrieved September 11, 2014, from http://www.vivago.com/pro/products/
- [21] Betterbuysrus. (n.d.). Cordless Fall Guard Monitoring Systems. Retrieved September 11, 2014, from http://www.betterbuys-r-us.com/Fall-Guard-Monitor-Cordless-Ststem.htm
- [22] Pioneer Emergency Medical Alarms. (n.d.). Auto Fall Guard. Retrieved September 11, 2014, from http:// www.pioneeremergency.com/product/auto-fallguard/
- [23] Tunstall. (n.d.). Fall Detectors. Retrieved September 11, 2014, from http://www.tunstall.com/solutions/fall

-detectors

- [24] Dai, J., Bai, X., Yang, Z., Shen, Z., & Xuan, D. (2010). Mobile Phone-based Pervasive Fall Detection. *Personal and Ubiquitous Computing*, 14(7), 633 -643.
- [25] Sposaro, F., & Tyson, G. (2009). iFall: An Android Application for Fall Monitoring and Response. *Annu*al International Conference of the IEEE Engineering in Medicine and Biology Society, (pp. 6119-6122).
- [26] Zhang, T., Wang, J., Liu, P., & Hou, J. (2006). Fall Detection by Embedding an Accelerometer in Cellphone and Using KFD Algorithm. *International Journal of Computer Science and Network Security*, 6(10), 277-284.
- [27] Oner, M., Pulcifer-Stump, J. A., Seeling, P., & Kaya, T. (2012). Towards the Run and Walk Activity Classification through Step Detection-An Android Application. 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, (pp. 1980-1983).
- [28] Noury, N., Rumeau, P., Bourke, A. K., O'Laighin, G., & Lundy, J. E. (2008). A Proposal for the Classification and Evaluation of Fall Detectors. *IRBM*, 29 (6), 340-349.
- [29] Bagalà, F., Klenk, J., Chiari, L., Cappello, A., & Becker, C. (2012). Accelerometer-based Algorithms on Real Falls Database. *Gait & Posture*, 35, S28.
- [30] Klenk, J., Becker, C., Lieken, F., Nicolai, S., Maetzler, W., Alt, W., et al. (2011). Comparison of Acceleration Signals of Simulated and Real-world Backward Falls. *Medical Engineering & Physics*, 33(3), 368-373.
- [31] White, T. S., Petsch, K., & Kaya, T. (2015). Evaluation of the Silicon-on-Glass Microfabrication Process for MEMS Accelerometers. *International Journal of Modern Engineering*, 15(1), 36-45.
- [32] Winter, C., Soylemez, N., Trivedi, J., Pickens, M., Craig, C., & Vaidyanathan, V. (2006). Design of a Sensor Based Smart Sprinkler System. *International Journal of Modern Engineering*, 7(1).
- [33] Lee, Y., Zhu, X., Pandya, A., & Hsu, S. (2010). Interactive Data Visualization and Analysis for Mobile-Phone Performance Evaluation. *International Journal of Modern Engineering*, 11(1), 82-94.

Biographies

JEFFRY A. PULCIFER-STUMP is a recent graduate from the Computer Science Undergraduate program of Central Michigan University. Mr. Pulcifer-Stump may be reached at <u>pulcilja@cmich.edu</u>

PATRICK SEELING is currently an Assistant Professor

FROM PIT TO LONG LIE: A FALL-DETECTION ALGORITHM FOR SMARTPHONES

in the Department of Computer Science at Central Michigan University. He received his Dipl.-Ing. Degree in Industrial Engineering and Management from the Technical University of Berlin (Berlin, Germany) in 2002 and his Ph.D. in Electrical Engineering from Arizona State University in 2005. He was a Faculty Research Associate and Associated Faculty member with the Department of Electrical Engineering at Arizona State University from 2005 to 2007. From 2008 to 2011, he was an Assistant Professor in the Department of Computing and New Media Technologies at the University of Wisconsin-Stevens Point. He currently leads the Distributed Internetworked Systems and Content (DISC) lab at Central Michigan University, reachable at DISClab.org. His research interests are in networking (with a focus on multimedia and energy optimizations), distributed and mobile systems, assistive technologies, and computer -mediated education, which are typically embodied in smart -device implementations. Dr. Seeling may be reached at seeli1p@cmich.edu

JANET L. KOCH is currently a high school science teacher at Garden City High School. She has been involved in the National Science Foundation granted Research Experiences for Teachers program at Central Michigan University during the summers of 2012 and 2013. Ms. Koch served as the president of the CMU chapter of the National Science Teachers Association for several years. Ms. Koch may be reached at <u>koch1j@cmich.edu</u>

STEVEN KETTLER is currently a high school science and mathematics teacher at Alma High School. He has been involved in the National Science Foundation granted Research Experiences for Teachers program at Central Michigan University during the summers of 2012 and 2013. Mr. Kettler has been actively involved in Science Olympiads and robotics projects as a mentor. Mr. Kettler may be reached at <u>skettler@almaschools.net</u>

TOLGA KAYA is currently an Assistant Professor at the School of Engineering and Technology of Central Michigan University (CMU). Prior to joining CMU, Dr. Kaya worked as a post-doctorate associate at Yale University's Department of Electrical Engineering. He received his Ph.D. degree in Electronics Engineering in 2007 from Istanbul Technical University, where he also worked as a graduate and teaching assistant for eight years. Dr. Kaya also worked as a technology consultant at Brigthwell Corporation in 2006 and a senior circuit designer at Microelectronics Circuit Design in Turkey from 1999 to 2006. His research interests include the design, fabrication, and testing of micro-scale sensors and microfluidic-based cell manipulation. Dr. Kaya is also exploring efficient ways of incorporating education technology and active learning strategies in his classrooms. Dr. Kaya may be reached at kaya2t@cmich.edu