34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Towards the Run and Walk Activity Classification through Step Detection – An Android Application

Melis Oner, Jeffry A. Pulcifer-Stump, Patrick Seeling, *Senior Member, IEEE*, and Tolga Kaya, *Member, IEEE*

Abstract— **Falling is one of the most common accidents with potentially irreversible consequences, especially considering special groups, such as the elderly or disabled. One approach to solve this issue would be an early detection of the falling event. Towards reaching the goal of early fall detection, we have worked on distinguishing and monitoring some basic human activities such as walking and running. Since we plan to implement the system mostly for seniors and the disabled, simplicity of the usage becomes very important.**

We have successfully implemented an algorithm that would not require the acceleration sensor to be fixed in a specific position (the smart phone itself in our application), whereas most of the previous research dictates the sensor to be fixed in a certain direction. This algorithm reviews data from the accelerometer to determine if a user has taken a step or not and keeps track of the total amount of steps. After testing, the algorithm was more accurate than a commercial pedometer in terms of comparing outputs to the actual number of steps taken by the user.

Index Terms— **Activity classification, Android, Fall Detection, Mobile Applications.**

I. INTRODUCTION

The world population is experiencing an enormous growth and studies predict that the elderly population will make up the majority of that population in the near future [1]. As the health insurance costs for the elderly are significantly higher than for younger generations, the increase of the number of senior members of societies worldwide will have a negative impact on any nation's budget, unless preventive healthcare systems are implemented [2]. Even though it might not be possible to avoid some health problems due to aging, it is quite possible to reduce the implications of predictable accidents, such as falling. Continuous monitoring of senior citizens in their daily activities to prevent or alleviate the impact of these types of events would only be feasible with personal medical or home aides – an approach that is clearly not practical. In turn, home accidents, in general, are one of the most seen accidents, which primarily include falls [3].

Therefore, significant recent efforts were focused on detecting a fall event or to classify a subject's activities over time. Most commonly, several 3-axis accelerometers are used in different locations of the subject such as ear, chest, arm, wrist, waist, knee, and ankle [4]. While gyroscopes can also be used to detect fall events [5], accelerometers and gyroscopes can be combined to gather more accurate data [6, 7].

It is obvious that the detection accuracy would significantly be enhanced by using multiple physically widespread sensors; however, it is impractical and uncomfortable for subjects to wear many sensors on specific locations.

We propose to use a mobile smart phone to detect a fall event regardless of the phone's position or orientation. Smart phones include various sensors such as gyroscopes, accelerometers, proximity sensors, see, e.g., [8-10], and have become affordable and ubiquitous. User interfaces that are incorporating assistive technologies make these devices usable for all population groups. In turn, using mobile applications for patient monitoring can alleviate some of the cost-related issues and improve both a patient and a caretaker's quality of life.

In this paper, we present our initial work on a pedometer mobile application component that is coupled with an e-mail component that can be used to notify medical assistants or family members. The developed algorithm counts the steps and identifies the type of activity with a higher accuracy than many of the applications that are commercially available.

The remainder of this paper is structured as follows. In Section II, we describe the developed pedometer algorithm and present performance results in Section III. We conclude and describe future works in Section IV.

M. Oner is with the School of Engineering and Technology, Central Michigan University, Mount Pleasant, MI 48859, USA (e-mail: oner1m@cmich.edu).

J. A. Pulcifer-Stump is with the Department of Computer Science, Central Michigan University, Mount Pleasant, MI 48859, USA (e-mail: pulci1ja@cmich.edu).

P. Seeling is with the Department of Computer Science, Central Michigan University, Mount Pleasant, MI 48859, USA (e-mail: pseeling@ieee.org).

T. Kaya (corresponding author) is with the School of Engineering and Technology, Central Michigan University, Mount Pleasant, MI 48859, USA (e-mail: kaya2t@cmich.edu, phone: 989-774-1494).

II. ALGORITHM AND CODE

The main idea for the step detection in this work relies on the detecting peaks in the data produced by the accelerometer sensor. These points represent the highest acceleration of data along the axes. In this study, walking is assumed to be a cyclical pattern according to Kokshenev's research about human walking dynamics [11]. With the common properties of a cyclical pattern, an algorithm can be formed to analyze the data more accurately. Therefore, with data representing a cyclical pattern, steps can be detected by finding the peak points within a period.

The algorithm implemented within the application gathers data from the accelerometer sensor that is built within smart phones. Every piece of data is analyzed to ensure a thorough analysis. The algorithm first establishes a control point equal to the value of the first piece of data. This control point is used as a variable to compare against to help determine whether a step has occurred. After the control point is set, a comparison is made between the first piece of data and the next piece. If the two pieces of data are equal to each other, the next piece of data is then compared. This process continues until the next value of the current piece of data does not equal the previous piece. Once the current value does not equal the previous, a comparison between the two values is made to determine whether the value resembles a peak. If the current value is greater than the previous then the control point is set to the current value. Then the algorithm continues through the data. If it is less, then the current value is compared against the control point to decide whether the value is a peak or not. If the value is less than the control point then continue through the data, otherwise compare the value against the range to decide if the value represents misleading data such as quick repetitious movements. The range is calculated by gathering the average of all the data thus far, and multiplying that by a correction factor, 1.15 in this case. (We note that this represents widening the margin for captured accelerometer activities by 15 percent.) If the value is greater than the range then it is considered a step and continues through the data. Otherwise the value is counted as a misleading step. Once at the end of the data, the total amount of steps is calculated by subtracting the total number of peaks representing steps minus the number of peaks representing misleading steps. The pseudo code for the algorithm is provided below.

```
SET n to first element          //index
SET controlPoint to the value of the first element
SET peakCounter to 0            //total steps
SET underAvgCounter to 0  //steps that fall within the range

WHILE not at the end of the Data {
   //IF current value is not equal to the next value THEN
   if(data[n] != data[n+1]){

      //IF current value appears to be a peak THEN
      if(data[n]>data[n+1]){

         //IF current value is not a peak THEN
         if(data[n]<control point){
            n++;}
         else{
            //IF current value is within the range THEN
            if (data[n]<avg*factor){
               underAvgCounter++;}

            peakCounter++;
            n++;}
         }
      //ELSE set new value of the control point
      else{
         controlPoint = data[n+1];
         n++;
      }
   }
   //ELSE continue through the data
   else{
      n++;
   }
}
   //Calculate steps taken
   peakCounter-=underAvgCounter;
```

III. TESTS

We conducted various tests to analyze the output of the application. These tests were conducted with the application installed onto a Samsung Admire smart phone which runs an Android operating system version 2.3. The installation process was performed through Eclipse, a software development program. It is a stand-alone application which allows for the activity to be started and stopped according to the user's desire. For comparison purposes, the tests were also conducted with a commercial pedometer built by Sportline, Incorporated. In order to help keep a steady pace when performing the tests, the subject walked to the pace of a metronome.

The first series of tests were designed to show graphical representations of a walking, running, and falling events. The procedure of these tests required the subject to walk twenty steps at sixty beats per minute (60 bpm), then run twenty steps at one hundred and four beats per minute (104 bpm), and finally fall on the ground. Fig. 1 shows the collected data from the smart phone's accelerometer sensor. It is clear that the energy of the walk is much less than the run, which could ultimately be used to distinguish a walk event from a run event. On the other hand, even though the peak for the fall event is not very high, the duration of the event is significantly longer (on the order of a few seconds). This unique characteristic of the fall event would be used to detect a fall event.

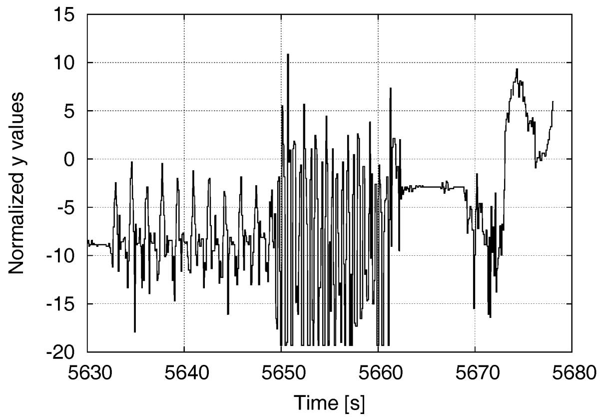

Figure 1. The data collected from the 3-axis accelerometer sensor events of a smart phone. Events are walking for 20 steps at 60 bpm, running for 20 steps at 104 bpm, and falling at the end of the run.

In order to characterize the accuracy of the algorithm using the smart phone as a platform, several test setups were designed to compare the smart phone application (which we refer to as "Application" scenario in the following) and commercial pedometer (which we refer to as "Commercial" scenario in the following) outputs against the actual number of steps taken under a different conditions.

In the first test, the subject was required to wear clothing that restricted the movement of the phone while it is inside the subject's pocket. The first sets of tests followed a procedure in which the subject was to walk a specified number of steps at a pace of one hundred and twenty beats per minute (120 bpm). Tests were repeated 5 times. Each test was performed by using the smart phone and the commercial pedometer at the same time. Tests were repeated for different step numbers from 20 steps to 300 steps. The resulting data is illustrated in Fig. 2. The solid line represents the ideal line, while markers with connected lines are commercial and application data. It is clear that application data agree very well with the commercial pedometer data as well as the theoretical line.

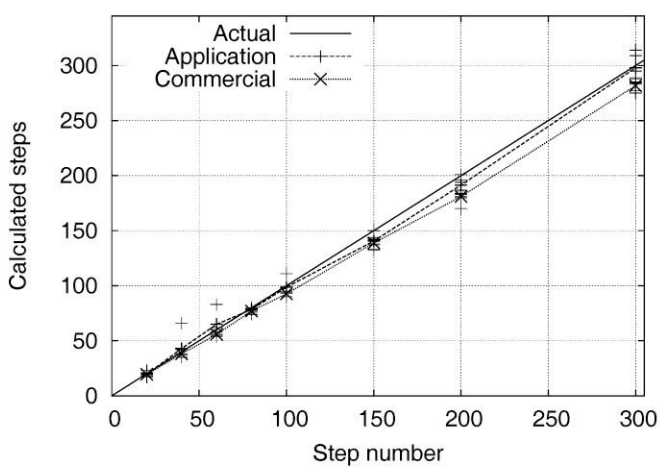

Figure 2. Reported steps from the developed smart phone application and a commercial pedometer. The trials are based on walking a specified number of steps with a fixed pace of 120 bpm with sensor units fixed in the pockets of the subject's pants.

In the second test, a specific number of steps were chosen (100 steps in this particular test) and steps were counted at different paces from 60 bpm to 130 bpm. Similarly to the first test, a set of 5 trials were performed with the sensor devices being fixed to the subject.

Fig. 3 illustrates the number of steps reported by the mobile application and the commercial pedometer. The results depict that the reported steps by both devices are impacted by the pace of the subject. We initially note a significant deviation from the actual number of steps for low paces (bpm). This is possibly due to the fact that slow pace movement introduces more noise to the system. With an increase in the pace, both application and commercial pedometer readings are approaching the actual number of steps taken. Overall, we note that the results become more accurate (within a 10% error margin) after 80 bpm.

Although the test results are very promising, it is important to note that Fig. 2 and Fig. 3 were derived from data obtained by fixing the sensor devices, so the noise effects would be minimized. However, in a real life scenario, specifically the case of elder usage, it is crucial that the sensor could still generate a consistent data even if the device is not positioned correctly. Therefore, another test setup was designed where the devices were placed in a loose pocket of the subject. To fulfill this condition, the subject was required to wear clothing that did not restrict the movement of the phone while it is inside the subject's pocket, i.e., in the side pockets of cargo pants.

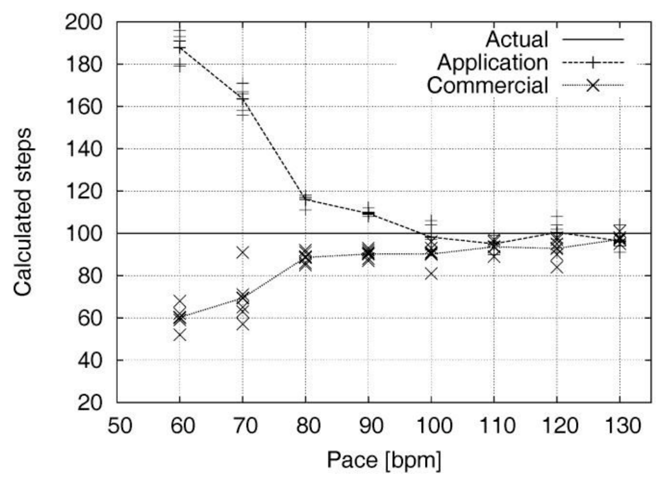

Figure 3. Steps are calculated at different paces from 60 bpm to 130 bpm. A total of 100 steps were recorded for each trial within this condition. The sensor units were fixed in the pockets of the subject's pants.

We illustrate the reported step counts from both pedometers using a fixed pace of 120 steps per minute (similar to the results presented in Fig.2), but with loose sensor placement, in Fig. 4. We initially observe that the smart phone application results in consistent step calculations, which furthermore are close to the number of actual steps. We additionally observe that the commercial pedometer's reported step count deviates significantly from the actual number of steps on the majority of trials conducted.

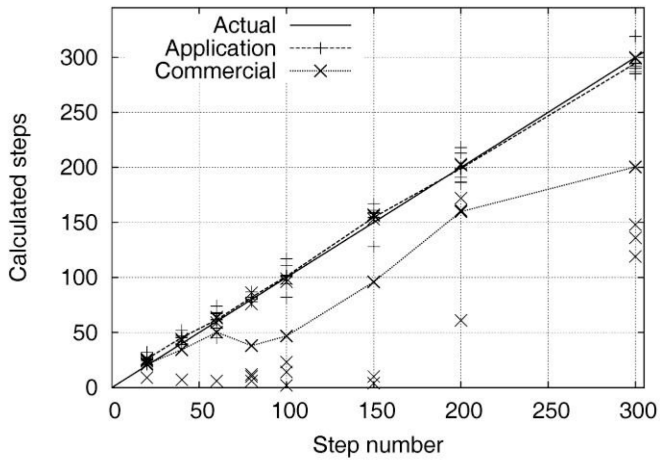

Figure 4. Reported steps from the developed smart phone application and a commercial pedometer. The trials are based on walking a specified number of steps with a fixed pace of 120 bpm with sensor units loose in the side pockets of the subject's cargo pants.

Comparing the results presented for the impact of the number of steps at a steady pace in Fig. 2 and Fig. 4, it becomes apparent that the commercial pedometer's position (fixed vs. loose) in the pocket plays an important role on the performance, whereas the proposed algorithm works consistently regardless of the device's orientation.

We illustrate the reported step counts from both pedometers using a fixed number of 100 steps at different paces from 60 bpm to 130 bpm in Fig. 5. We initially observe that for low paces, both pedometers' reported number of steps deviate significantly from the actual step count. With an increase in pace, however, we observe an increase in accuracy for the developed application, whereas the commercial pedometer continuously provides inconsistent results.

Comparing the results presented for the impact of the pace variation with a constant number of steps in Fig. 3 and Fig. 5, we initially note that the accuracy of the reported steps correlates with the pace for both developed application and commercial pedometer, whereby lower paces typically result

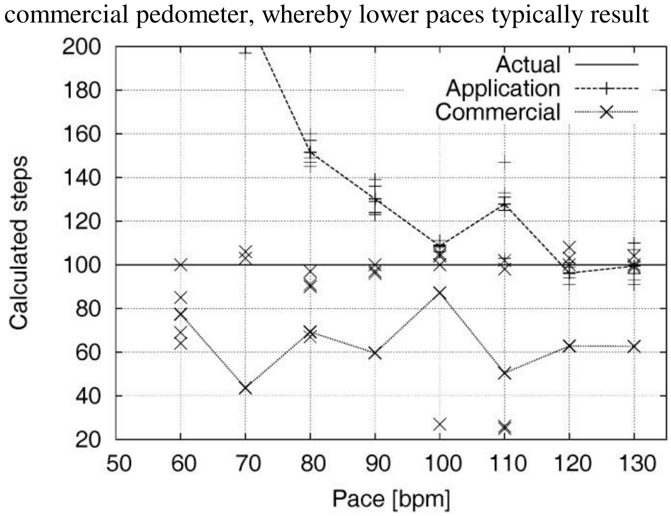

Figure 5. Steps are calculated at different paces from 60 bpm to 130 bpm. A total of 100 steps were recorded for each trial within this condition. The sensor units were loose in the side pockets of the subject's cargo pants.

in larger deviations from the actual number of steps and almost independently of the placement scenario (fixed vs. loose).

IV. CONCLUSION

In this project, a smart phone based step monitoring algorithm is proposed. The algorithm is simply based on peak detection on the acceleration information. The sensor platform was chosen to be a smart phone that would ultimately be the initial step for a wireless physiological activity monitoring system. The algorithm was tested for different pace of walk for various step numbers and the smart phone platform performed better than a commercial pedometer in terms of step count.

One of the biggest challenges is to process the sensor data consistently even if the sensor device's orientation is random. Initial tests were shown to be promising. Authors are now working on the improvement of the algorithm by performing a running averaging scheme in order to accurately determine the peaks. This work will be the base for a physiological activity monitoring system where activities can be classified and ultimately falls could be detected.

REFERENCES

[1] Mostarac,P., Hegeduš,H., Jurcevic,M., Malaric,R., Lay-Ekuakille, A,"Fall Detection of Patients Using 3-Axis Accelerometer System", in: IEEE International Workshop Medical Measurements and Applications, Torino, Italy, 2011, pp.456-459.

[2] Perolle, G., Fraisse P., Mavros,M., Etxeberria,, I., Fatronik, S., Lirmm, F., Zenon, G., Ingema,S. "Automatic Fall Detection and Activity Monitoring for Elderly", in Proc. MEDETEL,2006.

[3] Stevens, J.A., "Fatalities and Injuries from Falls Among Older Adults – United States, 1993–2003 and 2001–2005," MMWR 2006; 55(45).

[4] Atallah,L., Lo,B., King,R., Yang,G.Z. "Sensor positioning for activity recognition using wearable accelerometers" in Proc. Int. Workshop Wearable Implantable Body Sens. Netw., Los Alamitos, CA, 2010, pp.24–29.

[5] Li, Q., Stankovic, J.A., Hanson, M.A., Barth, A.T., Lach, J., Gang Zhou. "Accurate, Fast Fall Detection Using Gyroscopes and Accelerometer-Derived Posture", in BSN '09 Proc. of the Sixth International Workshop on Wearable and Implantable Body Sensor Networks, Berkeley,CA, 2009, pp.138-143

[6] L. Bao and S. S. Intille, "Activity Recognition from User-annotated Acceleration Data," in Proc. 2nd Int. Conf. Pervasive Comput., 2004, pp.1–17.

[7] Bourke, A.K. and Lyons, G.M., "A Threshold-based Fall-detection Algorithm using a Bi-axial Gyroscope Sensor," Medical Engineering and Physics, 30(1), pp. 778-998.

[8] "Samsung Admire User Manual," Web site: www.samsung.com/us/support/downloads/SCH-R720ZRAXAR Aug. 23, 2011.

[9] "iPhone User Guide For iPhone OS 5.0 Software," Web site: manuals.info.apple.com/en_US/iPhone_iOS3.1_User_Guide.pdf Oct. 20,2011, [Jan.19,2012]

[10] Maier, R., "Professional Android Development Book," Crosspoint,IN: WILEY, 2009.

[11] Kokshenev, V.B., "Dynamics of Human Walking at Steady Speeds," Physical Review Letters, 98, 208101-4, 2004.