31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

Off-the-Shelf Mobile Handset Environments for Deploying Accelerometer based Gait and Activity Analysis Algorithms

Martin Hynes[1], Han Wang[1] and Liam Kilmartin[1 2], Member, IEEE

*[1]Communication and Signal Processing Research Unit,
School of Engineering and Informatics,
[2] National Centre for Biomedical Engineering Science,
National University of Ireland, Galway
mhynes@nuigalway.ie*

Abstract — **Over the last decade, there has been substantial research interest in the application of accelerometry data for many forms of automated gait and activity analysis algorithms. This paper introduces a summary of new "of-the-shelf" mobile phone handset platforms containing embedded accelerometers which support the development of custom software to implement real time analysis of the accelerometer data. An overview of the main software programming environments which support the development of such software, including Java ME based JSR 256 API, C++ based Motion Sensor API and the Python based "aXYZ" module, is provided. Finally, a sample application is introduced and its performance evaluated in order to illustrate how a standard mobile phone can be used to detect gait activity using such a non-intrusive and easily accepted sensing platform.**

I. Introduction

THE study of human movement has been an active area of research over many years with one of the main aims being the identification of different variables that could be used to identify and characterise gait patterns. Other research has focused on the effect that aging has on walking parameters such as swing length and swing speed. Various studies have suggested the use of accelerometer based hardware to measure the acceleration due to the motion of the body. Most modern accelerometers are tri-axial in nature which means that the forces being measured can often, with careful physical placement of the accelerometer based sensing platform, represent physiologically relevant components e.g. anteroposterior, vertical and lateral. This type of force measurement methodology forms the basis of many proposals for monitoring human gait and activity in general. These applications include automated fall detection systems targeted at elderly populations[1] and energy expenditure estimation algorithms to support monitoring of adherence to prescribed exercise regimes[2] in the case of various disorders such as diabetes, cardio-vascular diseases etc. All of the systems that have been proposed to date have relied on the user wearing or carrying some external accelerometer based hardware platform in order to measure the forces that are exerted [3]. In many cases, such platforms also contained some form of wireless communication functionality to allow the system to download recorded or processed data to a remote server application. However, a common issue that has arisen in many trials using such systems is that the subjects in the studies have found that this external apparatus is quite uncomfortable and cumbersome to wear [4]. However, a recent trend in the area of consumer electronics involves the deployment of accelerometers embedded within certain classes of devices. For example, in the area of gaming, the Nintendo Wii controller uses accelerometers as a primary functionality to measure the physical actions of the player. Other devices, such as laptop computers, have used accelerometer devices to detect when the device has been dropped in order to implement actions which will protect sensitive subsystems (e.g. hard disk drive) within the device. Most interestingly, a very recent trend has seen the deployment of accelerometers within "off-the-shelf" cellular/mobile phone handset. In many of these handsets, the accelerometer functionality is not well known or utilised but because of the ubiquitous nature of the cellular handset in modern life, such a platform could provide an ideal and unobtrusive platform for deploying certain real time gait analysis and physical activity estimation applications based on accelerometry. This paper seeks to introduce the background to these new "software-only" based acceleromter environments which require absolutely no additional hardware customisation. In addition, because of the nature of the device, there is the added benefit of a built in data communication functionality to support real time downloading of raw or processed acceleromter data. Section 2 of this paper will review the area of accelerometer based gait and activity analysis . Section 3 of this paper will provides an overview of the mobile phone based acceleometer functionality and the software development environments available for developers wishing to implement applications which utilise the acceleromter sensor data in real time. Section 4 of this papers describes a basic sample application which has been implemented and evaluated using this mobile handset based platform. Section 5 summarises the main conclusions which have been drawn from this work and outlines further research which is currently being undertaken in this area.

II. Related Research

In this section of the paper, a review is provided of some of the application spaces in which gait\activity analysis has been applied in recently reported research. The main applications to which gait analysis algorithms have been deployed include: Biometric applications, Activity monitoring, Fall detection and Rehabilitation

The use of gait analysis has been proposed as a less intrusive method for implementing biometric authentications compared with other techniques such as voice, fingerprint or iris analysis. For example, in [5] it was shown how a combined acceleration signal could be used for a gait based biometric authentication procedure. This analysis was based on determining a statistical model relating to subject dependent gait parameters which was recorded in a database. During an authentication procedure, the equivalent parameter statistics were calculated and compared to the previously stored reference template. A basic similarity metric was used to determine whether to accept or reject the "walker" as the claimed identity.

Another area of significant recent research interest is the automated monitoring of daily activity using accelerometer platforms. In some cases, such research has also focused on the estimation of the surface slope when walking activity is taking place. Much of this research is based on the extraction of spectral features (e.g. Wavelet Packet Decomposition (WPD) [6] or Discrete Cosine Transform (DCT) [8]) from the accelerometer data and the use of various classification algorithms, including neural networks, Support Vector Machines etc. Another WPD based method was outlined in [7] which was implemented using a mobile phone handset using a custom accelerometer platform connected to the phone via an external port. The accelerometer data was then sent to a remote server where a gait pattern classification algorithm based on a WPD analysis followed by a Bayesian Classifier was implemented. This algorithm was shown to be 80% successful in pattern classification. Another alternative method outlined in [8] using custom accelerometer hardware used a DCT feature extraction stage followed by a Gaussian Mixture Model (GMM) to classify the walking patterns. This method was shown to be 86% successful in subject dependent gait pattern classification.

A final noteworthy area of significant research relates to monitoring the activity of elderly populations, particularly those suffering from degenerative disorders which affect gait such as Parkinson's Disease (PD). The impact of PD on gait patterns was reported as being clearly identifiable using accelerometer data in [3]. Also, by using an FFT analysis on accelerometer data, the "freezing effect" on the gait pattern due to PD was reported as identifiable in [9]. Such classifications can be very useful for clinicians when it comes to measuring the progression of the disease. Fall detection is another area in which the use of accelerometer platforms has become commonplace. Kangas et al. [10] applied a simple threshold technique to accelerometer data and when applied in combination with some posture techniques it leads to a 100% detection of falls from a recorded database.

III. Mobile Phone Accelerometer Support

Accelerometers are increasingly being incorporated into the hardware of many modern cellular/mobile phone handsets. A non-exhaustive list of currently available mobile handsets which contain embedded accelerometer devices includes: Sony Ericsson W760i. Nokia 5500, N95 and N97,Samsung Omni, Blackberry Storm, Apple iPhone and Google G1. Since each of the mobile phone manufacturers may utilise different embedded accelerometer devices, it is essential that a standardised and portable development environment exists for software developers wishing to deploy applications on the handsets that utilise such accelerometer hardware. A mobile sensor Application Programming Interface (API) facilitates the development of applications which require access to data from any embedded sensors on the mobile handset (e.g. accelerometers). The two main development environments which allow mobile handset applications to interface with the mobile handset's sensors use either the Java based JSR 256 API or alternatively the Symbian Sensor API. Some of the handsets, most notably the Apple iPhone, use proprietary software APIs which are in no way cross-platform portable and hence these will not be discussed in any more detail.

A. *JSR 256 API*

The Java based JSR 256 API [11] is a generic API which can be used on any type of device supporting Java Micro Edition (Java ME). In terms of mobile handsets, this API is currently supported by both Samsung and Sony Ericsson mobile handsets with certain upcoming handsets from Nokia also likely to support the API. The JSR 256 API provides a generic architecture to allow J2ME developers to interact with the mobile devices' sensors. The JSR 256 API provides interfaces that the sensor's manufacturers can implement which will then allow a developer to design a single Java ME application which is compatible with any JSR 256 compliant handset. The JSR 256 Mobile Sensor API (using the javax.microedition.sensor package) allows Java ME application developers to access sensor data easily and uniformly. The class structure of the JSR 256 API is defined in the JSR specification[11]. It can be seen in the javax.microedition.sensor class structure that the only class actually implemented is the Sensor Manager which returns a list of all available sensors. All of the interface definitions in this specification are indicative of methods which must be implemented by the mobile handset manufacturer in order to allow third party Java ME developers to access their sensors in the standardised manner.

In the case of handsets that include tri-axial accelerometers (some handsets currently only support dual axis accelerometers), each of the three data channels are managed using two different interfaces, namely: ChannelInfo and Channel. The ChannelInfo interface defines the properties of the data, for example, the accuracy of the measurement or the units of the measurement. The Channel interface maintains condition objects attached to each channel of the sensor, for example the valid range of accelerometer signal values, with these conditions being used to trigger an application to read new accelerometer data.

B. Mobile Sensor API

The Mobile Sensor API is an alternative programming interface available to application developers wishing to access the accelerometer functionality and it is most commonly used in Nokia mobile devices which are based on the Symbian operating system. The Sensor plug-in, which is available in the S60 Software Development Kit (SDK) [12], is used to access data from the embedded sensors on such devices. However there are several complications associated with the use of the Mobile Sensor API with Nokia Mobile Devices. For example, in the case of early versions of the N95 (firmware earlier than 20.0.015), the developer can only access one sensor and it is only possible to access the 4 values of tilt for that sensor. A possible solution to this problem involves the use of the N95 RD Accelerometer plug-in. More recent versions of the N95 handset, allows the developer to use the sensor API to access data for all 3 axis with no limitations on the number of samples which can be read. The S60 SDK (3rd edition) saw the introduction of the new Sensor Framework which supports a new set of APIs offering more versatility and extendibility as well as new possibilities relating to sensor data formats and usage scenarios. It must be stated that use of this API requires the application to be developed in Symbian C++ which has a slightly different structure to ANSI C++ and it is widely accepted as being a very challenging development environment even for expert software developers.

An alternate development option exists for less accomplished developers who do not wish to develop an application using C++. Python for the S60 SDK brings the power and productivity of the Python programming language to the S60 platform. These tools enable rapid application development and prototyping, and the ability to create stand-alone S60 applications written in Python. The "aXYZ" module is a sensor extension for Python which gives developers access to the N95 accelerometer. However, the primary disadvantage of this approach is that a developer must first install the Python Platform along with the Nokia RD Accelerometer Plug-in before installing the “aXYZ” module. In summary, the Mobile Sensor API has some major disadvantages namely:

- The mobile device must be Symbian based.
- In order to develop applications in either C++/Python the developer must install additional plug-ins.

Therefore the J2ME based JSR 256 API provides a more “platform independent” portable solution which is also more developer friendly. Indeed, Nokia handsets will support the JSR 256 API in future S60 SDK (5th Edition) based mobile phones, such as the N97 model.

IV. SAMPLE ALGORITHM IMPLEMENTATION

A basic Java ME gait analysis application was developed using the JSR 256 API for execution on a Sony Ericsson W580i handset in order to both investigate and illustrate the ease of implementation and to investigate the performance of the overall application. The algorithm, which implements a simple gait activity detection algorithm, is shown in figure 1. The application executing on the handset processes the three accelerometer channels at a sample rate of 20Hz without any significant impact on the overall performance of other software on the handset. In addition to implementing the algorithm, the sample application also automatically downloads the results of the gait analysis in real time (along with GPS data on the subject’s location) over a GPRS/3G connection to a remote database enabled server.

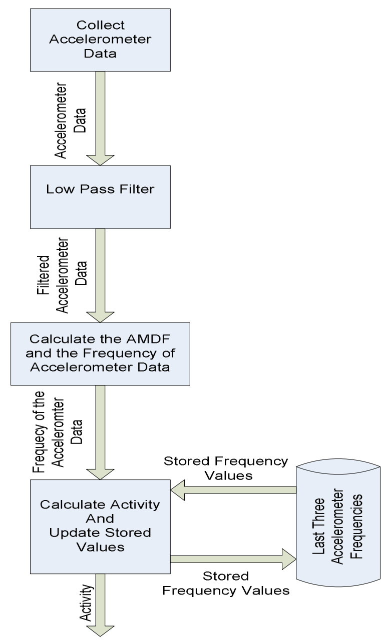

Figure 1: Algorithm Design

Firstly for each accelerometer channel a frame of 60 samples (3 seconds) of accelerometer data is collected and passed through a fifth order FIR filter with a cut-off frequency of 0.5 radians/sec. An Average Magnitude Difference Function (AMDF), as given by equation (1), is calculated for the resultant frames of filtered data in order to facilitate detection of gait activity through identification of the presence of a strong periodicity on one or more of the channels

$$D(k) = \frac{1}{N}\sum_{n=0}^{N-1}\left|s(n) - s(n+k)\right| \quad \textbf{(1)}$$

where**:**

N: Frame size in samples
S(n): Filtered Accelerometer Data
D(k): AMDF value for a delay of k samples

Figures 2 and 3 show examples of the raw accelerometer data stream and an example AMDF for both walking and non-walking activities.

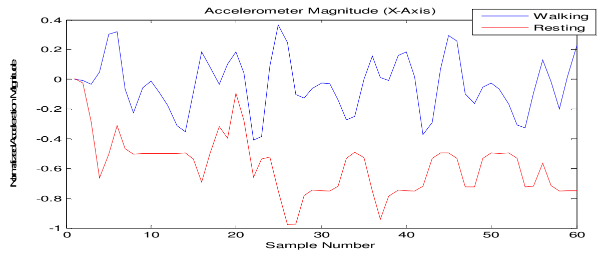

Figure 2: Accelerometer Data

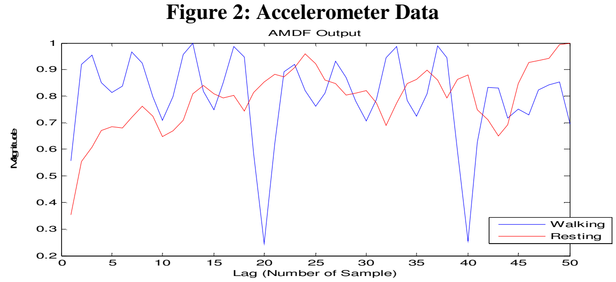

Figure 3: AMDF Output

The graphs clearly illustrate the expected periodicity of the walking trace while the resting trace displays no obvious periodicity. The last block in the overall algorithms design was to implement a basic two-state state machine (implementing a "walking" and "non-walking" state). In order to make the algorithm more robust, transitions from one state to the other are not based on the result of analysing a single frame and detecting periodicity on any of the channels. Instead transitions only occur when four out of the "last" five frames indicate that a transition in state should take place. Performance evaluation tests were carried out with subjects carrying the phone handset in a "normal" fashion (e.g. in a jacket pocket). Each of five healthy subjects were asked to carry a handset supporting the application during parts of their normal daily activities. Such activities would typically consist of both walking and non-walking periods. The subjects were accompanied during these tests to allow the output of the classification algorithms to be compared with actual activity being undertaken. Table 1 summarises the results of these tests in the form of a confusion matrix and illustrate a surprisingly good performance given the relative simplicity of the algorithm and the lack of any restrictions in terms of where the mobile handset was stored during activities.

	Walking	Resting
Walking	89.2%	10.8%
Resting	1.18%	98.82%

Table 1: Confusion Matrix for Algorithm's Recognition Rate

V. CONCLUSIONS

This paper has focused on providing an overview of how "off-the-shelf" mobile handsets equipped with embedded accelerometers could potentially provide a convenient and natural device for implementing certain automated gait and activity monitoring for "out of the laboratory" deployments. A summary and brief critique of the most commonly used programming techniques to allow custom accelerometry based applications to be developed for such handsets was provided. The structure and performance of a sample gait activity detection application which was deployed on one such handset was summarised. It is clear that this type of platform could potentially provide a very convenient and easily accepted platform as a means of implementing many forms of automated gait and activity analysis. However, there are some issues which require further research. These include issues relating to the placement and likely unpredictable movement of handsets during "normal" activity, particularly when the application requires knowledge of the accelerometer orientation. Other issues relate to the impact which "normal phone activity" (e.g. making calls, texting etc.) would have on overall algorithm performance. The latter issues can be addressed to some degree as it would be possible to "suspend" any accelerometer data analysis when such events are detected by the application but studies need to be completed with this form of platform to determine whether such action need actually take place or perhaps whether the underlying gait or activity would still be detectable in the presence of much more complex "noise" in the accelerometer data.

REFERENCES

[1] T. Yoshida, F. Mizuno, T. Hayasaka, K. Tsubota, S. Wada, and T. Yamaguchi, "Gait analysis for detecting a leg accident with an accelerometer," *1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, 2006.*

[2] R. Kauw-A-Tjoe, J. Thalen, M. Marin-Perianu, and P. J. M. Havinga, "Sensorshoe: Mobile gait analysis for parkinson's disease patients," in *UbiComp 2007 Workshop Proceedings, Innsbruck, Austria*, 2007.

[3] Jong Hee Han, Hyo Sun Jeon and K. S. Park, "Gait detection from three dimensional acceleration signals of ankles for the patients with parkinson's disease," in *International Conference on Technology and Applications in Biomedicine, 2008. ITAB 2008.*, May 2006, pp. 349–352.

[4] J. A. DeLisa and K. Casey, *Gait analysis in the science of rehabilitation*. DIANE Publishing, 1998.

[5] D. Gafurov, K. Helkala, and T. Søndrol, "Biometric gait authentication using accelerometer sensor," *JCP*, vol. 1, no. 7, pp. 51–59, 2006.

[6] N. Lovell, N. Wang, E. Ambikairajah, and B. Celler, "Accelerometry based classification of walking patterns using time-frequency analysis," in *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007. EMBS 2007.*, Aug. 2007, pp. 4899–4902.

[7] T. Iso and K. Yamazaki, "Gait analyzer based on a cell phone with a single three-axis accelerometer," in *MobileHCI '06: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. New York, NY, USA: ACM, 2006, pp. 141–144.

[8] R. K. Ibrahim, E. Ambikairajah, B. Celler, N. H. Lovell, and L. Kilmartin, "Gait patterns classification using spectral features," in *IET Irish Signals and Systems Conference, 2008*, June 2008, pp. 98–102.

[9] J. H. Han, W. J. Lee, T. B. Ahn, B. S. Jeon, and K. S. Park, "Gait analysis for freezing detection in patients with movement disorder using three dimensional acceleration system," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2003.*, vol. 2, Sept. 2003, pp. 1863–1865 Vol.2.

[10] M. Kangas, A. Konttila, I. Winblad, and T. Jamsa, "Determination of simple thresholds for accelerometry-based parameters for fall detection," in *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007. EMBS 2007.*, Aug. 2007, pp. 1367–1370.

[11] JSR 256: Mobile Sensor API, "http://jcp.org/en/jsr/ec."

[12] Symbian S60 SDK, "www.symbian.com."