A Fall Detection System based on SensorTag and Windows 10 IoT Core

Yuejiao Cheng[1, a], Chenglong Jiang[1, b], Jiong Shi [1, c]

[1] School of Electronic and Information Engineering, Zhejiang Wanli University, Ningbo, 315100, China

[a]email: 349513290@qq.com, [b]email: 389568639@qq.com, [c]email: shijiong@outlook.com

Keywords: Fall Detection; Wearable Sensors; Data Fusion

Abstract. Falls that lead to fatal injury have become a great challenge that cannot be neglected for elderly people. In this study, a surveillance system based on SensorTag and Windows 10 IoT Core for real-time fall detection is proposed. Raw data including three-dimensional accelerometer, gyroscope, and magnetometer are provided by SensotTag. Windows 10 IoT Core device makes use of these information to get the orientation of the subject by efficient data fusion and fall detection algorithms. Microsoft Azure services and Mobile/PC applications are also designed to achieve seamless data processing, analyzing, storing and acquiring at any time from any place as long as they have access to the Internet. Tests of the proposed system are performed according to experimental protocols including intentional falls and activities of daily lives. The results show that the proposed fall detection solution is reliable and effective.

Introduction

Population ageing has constituted an enormous threat to a wide range of economic, political and social processes since the 20th century in the western countries. And the situation in China is increasingly resembling that of western countries because of our birth-control policies. The change in age structure caused by population ageing will lead to the rising medical costs and increasing demands for elderly healthcare services. As a great challenge in elderly healthcare domain, fall detection has been deeply concerned by academic and medical researchers [1-3]. It's of great importance to develop an intelligent surveillance system to monitor the daily activities of the elderly and detect falls automatically.

In the last few years, different kinds of approaches have been proposed in fall detection area, which can be explained and categorized into three types: wearable device based, ambience sensor based and vision based [4]. First of all, wearable devices usually take advantages of embedded sensors to detect the motion and location of the body, such as accelerometer, magnetometer and gyroscope [5, 6]. And the cost of wearable device based approach is quite low, as well as the installation and operation is not complicated for the elderly. Secondly, ambience based approach always use pressure sensors to detect and track body. This solution is also cost-effective and easy-deployment [7, 8]. However, the possibility of sensing objects other than human bodies posts a remarkable challenge to the detection accuracy of this approach. Last but not least, vision based solution make full use of deployed cameras to monitor all the objects with-in the range, including human bodies [9, 10]. There is less intrusion into people's daily life than the above two approaches, while the observation space is limited and ubiquitous monitoring can't be achieved.

In this paper, a fall detection system based on Texas Instruments SensorTag and Windows 10 IoT Core is proposed. SensorTag is encapsulated with accelerometer, magnetometer and gyroscope sensors to overcome the limitation of a single accelerometer in existing solutions. A Raspberry PI 2 with Windows 10 IoT Core get the information of 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer from SensorTag via Bluetooth LE and perform the fall detection algorithm to provide real-time fall detection results of the people wearing it. Furthermore, a Microsoft Azure based schedule is proposed to achieve online data process, storage and management. The rest of this paper is organized as follows. The architecture of the proposed fall detection system is described first. Then, the hardware design and software design are represented. Experimental results are also demonstrated and analyzed. Finally, conclusions are made.

System Architecture

The architecture of the proposed solution is described in Fig.1. Generally, the system is composed of SensorTag, Windows 10 IoT Core device (Either Raspberry Pi 2 or Min-now Board MAX), Microsoft Azure, smart phone and PC/Tablet. The CC2541 TI SensorTag is used to collect real-time parameters of sensors which indicate the activity and posture of the target human body. Then all the parameters are sent to Windows 10 IoT Core device by Bluetooth LE. Windows 10 IoT Core device carry out the orientation filter and fall detection algorithm by means of the received data to achieve fall detection. Then both the parameters of sensors and the result of fall detection are packaged as AMQP (Advanced Message Queuing Protocol) messages for trans-mission to Microsoft Azure Event Hubs. Event Hubs is capable of collecting millions of event data per second, while Stream Analytics then process and analyze the massive amounts of data in time. And the filtered data are then stored in Azure Storage for further use. Universal applications for mobile phone, PC and Tablet based on Universal Windows Platform are designed to provide remote access to Azure Storage service to show the status of people who carry the SensorTag.

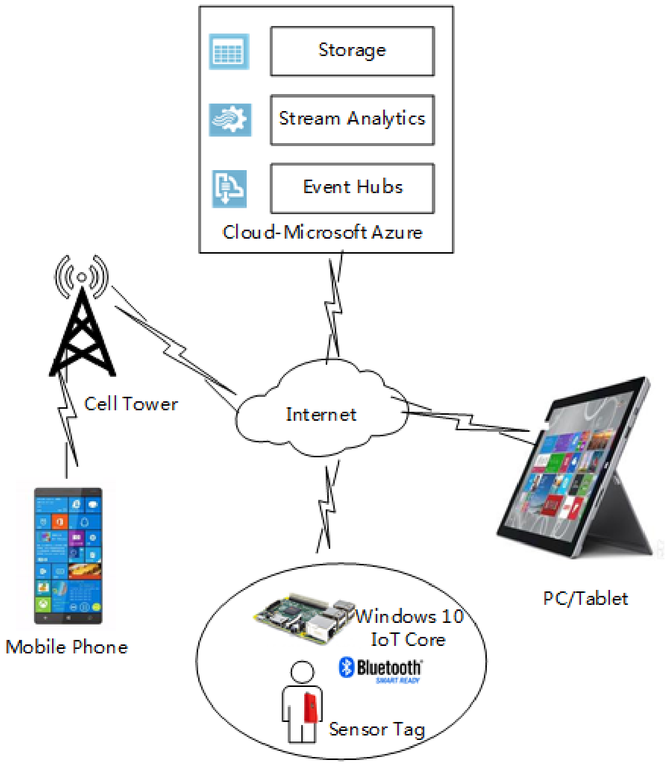


Fig.1. System architecture

Design of Windows 10 IoT Core Node

This study use Raspberry Pi 2 Model B as Windows 10 IoT Core Node. For the peripheral devices, ORICO BTA-403 Mini Bluetooth 4.0 USB Dongle is plugged into the USB port to communicate with SensorTag. In addition, Raspberry Pi 2 support WiFi Module to achieve wireless Internet access. So, the solution is easy to deploy in that both SensorTag and Windows 10 IoT Core device can be moved unrestricted to meet diversified requirements.

More and more wearable devices adopt Bluetooth Low Energy due to its key features as low power consumption, small size and low cost. The SensorTag is a BLE device powered with the TI CC2541 chip, which features a programmable Bluetooth 4.0 stack. There are six sensors on the device, including temperature, humidity, gyroscope, accelerometer, magnetometer and barometric pressure, which are exposing data through the GATT (Generic Attribute Profile). Windows 10 IoT Core Node use Bluetooth 4.0 USB Dongle to achieve the communication with SensorTag. The GATT profile of SensorTag is listed in Table 1.

Table 1. The GATT profile of SensorTag

GATT UUID Value (Hex)	GATT Server Permissions	Description
f000aa10-0451-4000-b000-000000000000	Read	Accelerometer Service
f000aa11-0451-4000-b000-000000000000	Read	Accelerometer Data
f000aa12-0451-4000-b000-000000000000	Read and Write	Accelerometer Configuration
f000aa30-0451-4000-b000-000000000000	Read	Magnetometer Service
f000aa31-0451-4000-b000-000000000000	Read	Magnetometer Data
f000aa32-0451-4000-b000-000000000000	Read and Write	Magnetometer Configuration
f000aa50-0451-4000-b000-000000000000	Read	Gyroscope Service
f000aa50-0451-4000-b000-000000000000	Read	Gyroscope Data
f000aa50-0451-4000-b000-000000000000	Read and Write	Gyroscope Configuration

Design of Fall Detection Algorithm

The fall detection algorithm is designed according to the variation of acceleration during an accidental fall. It's shown in Fig.2 that the whole process can be divided into four phases, that is, Start, Impact, Aftermath and Posture [11]. The sequence of four phases and the value of the acceleration are both considered to make the decision in this fall detection algorithm.

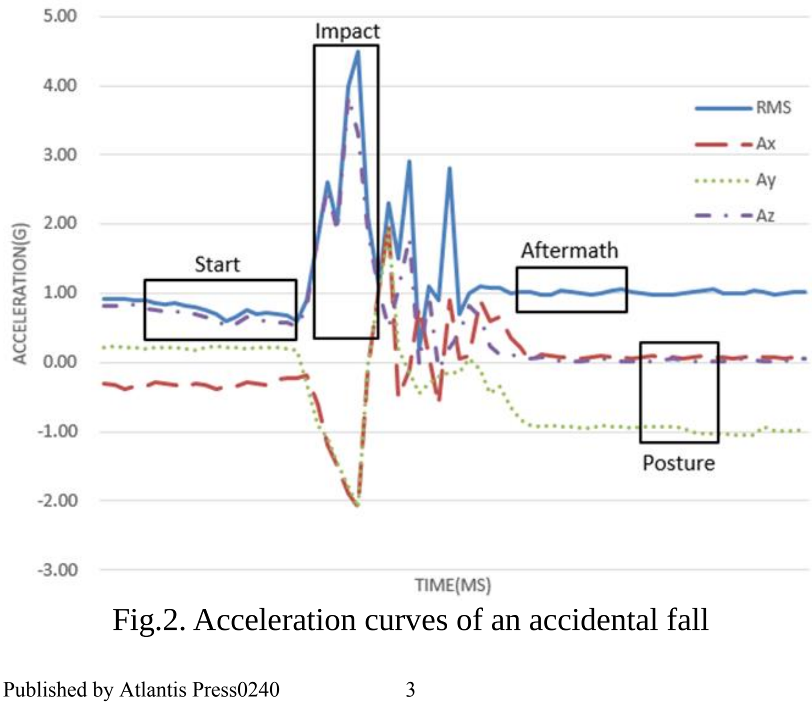


Fig.2. Acceleration curves of an accidental fall

First of all, an orientation filter is performed by the sensor arrays which employs a quaternion representation of orientation [5]. A quaternion $\hat{q}$ is a four-dimensional complex number which is defined as

$$\hat{q} = [q_1, q_2, q_3, q_4] \tag{1}$$

where q_1 is the scale part of the quaternion whereas q_2, q_3 and q_4 are the vector parts.

With the fusion of two different orientation calculations, the quaternion orientation of $q_{est,t}$ can be defined as

$$q_{est,t} = \gamma_t q_{\nabla,t} + (1-\gamma_t) q_{\omega,t}, \ \ 0 \le \gamma_t \le 1 \tag{2}$$

where $q_{\nabla,t}$ represents the orientation calculated from accelerometer and magnetometer, while $q_{\omega,t}$ represents the orientation calculated from gyroscope. γ_t is the factor given to each orientation calculation.

Secondly, Yaw, Pitch and Roll angles are calculated from quaternion data to depict another feature of orientation in three-dimensions. The three angles can be derived directly as

$$Yaw = atan2(2q_2q_3 - 2q_2q_4, 2q_1^2 + 2q_2^2 - 1) \tag{3}$$

$$Pitch = \text{-}sin^{-1}(2q_2q_4 + 2q_1q_3) \tag{4}$$

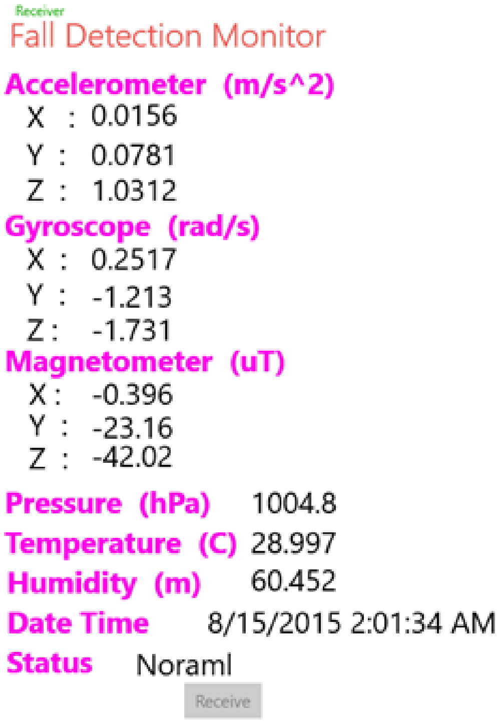
$$Roll = atan2(2q_3q_4 - 2q_1q_2, 2q_1^2 + 2q_4^2 - 1) \tag{5}$$

Compared with conventional Kalman-based solution, this orientation filter is suitable for real-time applications, since it has much lower computational complexity. Based on the acceleration and orientation data, the fall detection algorithm can be designed with the help of the recognition of Start, Impact, Aftermath and Posture Phase. On one hand, the RMS of 3-axis accelerometer can be used for Impact and Aftermath phase detection, on the other hand, Pitch and Roll angles of the orientation filter can be used for Posture phase detection.

Application Design

The application is developed based on the Windows Universal temple by Visual Studio 2015 Enterprise RTM version on Windows 10 Build 10240. The UI (User Interface) is described by MainPage.xaml file, while the user logic and data collection are located in MainPage.xaml.cs file. There are total seven GATT services on the SensorTag that we are interested in, and for each of those services we will need to create a GattDeviceService object in order to interact with them. Once we have a GattDeviceService object we can then obtain a GattCharacteristic object, which allows us to interact with GATT characteristics. We use these objects to write, read, and set up notifications with the SensorTag GATT characteristics. Then notification handlers are provided to read data from SensorTag. Furthermore, the data are used by Fall Detection Algorithm to get real-time status of the observed person. Finally, all the sensor data and the result of Fall Detection Algorithm are packaged as AMQP messages and sent to Windows Azure for further operation. The UI of the application which running on Windows 10 IoT Core is shown in Fig.3a.

(a) UI on Windows 10 IoT Core



(b) UI on Windows 10 PC

Fig.3. The UI of the solution

In this solution, we create a single app package that can be installed on both Mobile and PC. As the main goal of the application is to provide the real-time status information of the person who carry the SensorTag, so the UI designed here is quite simple and efficient. As shown in Fig.3b, only the interested sensor data and the result of fall detection algorithm are presented on the main page.

Experimental results

In most of the previous studies, intentional falls and activities of daily lives (ADL) are often tested to verify the reliability and efficiency of the fall detection algorithm. In this study, there are total 15 activities of the experimental protocol in Table 2.

Table 2. The Experimental Protocol

Category	**Test No.**	**Activities**	**Overall correct**	**Accuracy (%)**
Fall	1	Fall back with legs straight	28	93.3
	2	Fall forward with legs straight	29	96.7
	3	Fall back with knees bent	27	90
	4	Fall forward with knees bent	30	100
	5	Fall left with knees bent	30	100
	6	Fall right with knees bent	30	100
	7	Fall left with knees straight	27	90
	8	Fall right with knees straight	26	86.7
	9	Fall while sitting on a chair	28	93.3
	10	Trip over a small object	29	96.7
ADL	11	Sitting down and standing up from an armchair	30	100

	12	Lying down and standing up from a bed	30	100
	13	Walking a few meters	30	100
	14	Stretching while standing	30	100
	15	Picking up an object from the floor	30	100

Five volunteers performed the experimental protocol with SensorTag attached to the belt. The total number of simulated falls is 300, with 30 times of each fall activities. While the overall number of simulated ADL is 150, with 30 times of each ADL activities.

According to the result table, the proposed solution achieves an overall accuracy of 96.4%, which shows nearly the similar performance with the algorithm proposed in literature [5]. In addition, the real-time orientation filter has much lower computational load by the low complexity of the orientation estimation algorithm.

Conclusion

In this paper, a fall detection system based on SensorTag and Windows 10 IoT Core is proposed. The architecture of the system is first described. Then the hardware and soft-ware design details are represented, including the design of Windows 10 IoT Core node, the design of fall detection algorithm and the configuration of Windows Azure. Experimental results show that the proposed fall detection solution is reliable and effective.

Acknowledgements

This work was financially supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LY14F010007, National Natural Science Foundation of China under Grant No. 61471061, National College Students' Innovative Entrepreneurial Training Program of Zhejiang Wanli University under Grant No. 201410876005.

References

[1] S. Bo Yu, K. C. Ho, M. J. Rantz, and M. Skubic, "Doppler Radar Fall Activity Detection Using the Wavelet Transform," Biomedical Engineering, IEEE Transactions on, vol. 62, pp. 865-875, 2015.
[2] J. He and C. Hu, "A portable fall detection and alerting system based on k-NN algorithm and remote medicine," Communications, China, vol. 12, pp. 23-31, 2015.
[3] K. Lih-Jen and C. Chih-Sheng, "A Smart Phone-Based Pocket Fall Accident Detection, Positioning, and Rescue System," Biomedical and Health Informatics, IEEE Journal of, vol. 19, pp. 44-56, 2015.
[4] L. Jian and T. E. Lockhart, "Development and Evaluation of a Prior-to-Impact Fall Event Detection Algorithm," Biomedical Engineering, IEEE Transactions on, vol. 61, pp. 2135-2140, 2014.
[5] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A High Reliability Wearable Device for Elderly Fall Detection," Sensors Journal, IEEE, vol. 15, pp. 4544-4553, 2015.
[6] R. Freitas, M. Terroso, M. Marques, J. Gabriel, A. Torres Marques, and R. Simoes, "Wearable sensor networks sup-ported by mobile devices for fall detection," in SENSORS, 2014 IEEE, 2014, pp. 2246-2249.
[7] Z. Xiaodan, H. Jing, G. Potamianos, and M. Hasegawa-Johnson, "Acoustic fall detection using Gaussian mixture models and GMM supervectors," in Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, 2009, pp. 69-72.

[8] L. Yun, K. C. Ho, and M. Popescu, "Efficient Source Separation Algorithms for Acoustic Fall Detection Using a Microsoft Kinect," Biomedical Engineering, IEEE Transactions on, vol. 61, pp. 745-755, 2014.
[9] M. Yu, S. M. Naqvi, A. Rhuma, and J. Chambers, "One class boundary method classifiers for application in a video-based fall detection system," Computer Vision, IET, vol. 6, pp. 90-100, 2012.
[10] E. E. Stone and M. Skubic, "Fall Detection in Homes of Older Adults Using the Microsoft Kinect," Biomedical and Health Informatics, IEEE Journal of, vol. 19, pp. 290-301, 2015.
[11] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," Information Technology in Biomedicine, IEEE Transactions on, vol. 10, pp. 156-167, 2006.

