2016 IEEE First Conference on Connected Health: Applications, Systems and Engineering Technologies

A Real-time Low-complexity Fall Detection System On The Smartphone

Weihao Qu, Feng Lin and Wenyao Xu
Department of Computer Science and Engineering
University at Buffalo, Buffalo, US
Email: {weihaoqu,flin28,wenyaoxu}@buffalo.edu

***Abstract*—The importance of detecting falls in real-time is more and more emphasized recently. In this paper, we develop a real-time fall detection system on the smartphone. It is based on our low-complexity fall detection algorithm [1] that can detect the fall after the fall event happens. The system can distinguish dangerous falls by setting a monitoring time. The system is implemented on the Android platform and targeted on low energy consumption and fast processing, which can be seamlessly applied into wearable products.**

Keywords—Fall detection, Android, Real-time

I. Introduction

The fall accident is one of the major health risks, especially for the elderly (aged 65+). Thirty three percent of the elderly are reported having experienced at least one fall per year and 68% hospitalization of the elderly are fall-related [2]. The direct annual medical cost of elderly falls rises to about 20 billion dollars [3]. Even worse, about 10,000 deaths per year among people aged 65 years and older in the U.S. occur due to fall accidents [4].

Fall detection is urgently demanded. Wearable sensors such as inertial sensors play a key role in fall detection. Most fall detection products need multiple sensors such as the gyroscope and accelerometer working together to make a prediction. However, handling multiple sensors' information makes these algorithms too complex and the detection procedure energy-consuming. It has become a challenge when integrating multiple sensors into wearable products.

The proposed real-time fall detection system is based on our low-complexity fall detection algorithm [1] to implement. The system only uses acceleration, which requires less energy consumption compared to other multiple-sensor-based fall detection products. The sensor data are collected and processed by an Android App in real-time. The App will alarm, post the user's location and ask for help on certain social media once it detects fall events.

II. DESCRIPTIONS

A. Real-time Algorithm Adaption

The original algorithm in our previous research [1] needs to be adapted for handling the real-time sensor data. The median filter is needed to filter the noise of the raw sensor data. As shown in Fig. 1, in State 0, when sensor data (acceleration) update in real-time, a buffer is used to store the most recent acceleration value. In our case, we set the buffer size the same as the window size of the median filter. Every time the sensor data update, the buffer will delete the oldest element in the tail and add the newly coming data into the head. The middle element of the sorted buffer is the result after applying the real-time median filter.

Since the accelerometer provides 3-axis accelerations, we have three temporary buffers correspondingly to store the median filtered accelerations x, y, and z. The overall acceleration O and the overall differential of acceleration D will be calculated. The differential along one direction is calculated by temporarily storing the last acceleration along

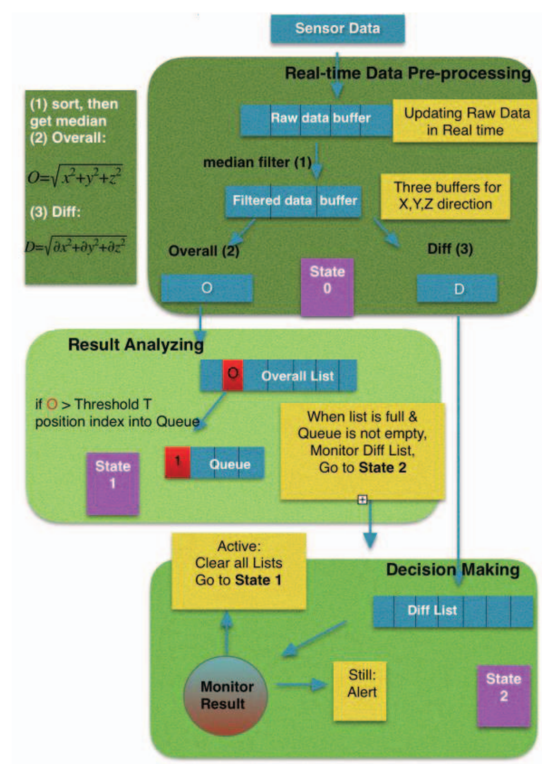

Fig. 1. Real-time fall detection data flow diagram.

the direction. For instance, the differential along x axis dx is

978-1-5090-0943-5/16 $25.00 © 2016 IEEE
DOI 10.1109/CHASE.2016.73

calculated by using last acceleration x' in (1). Likewise, differential along y, z axis *d*y, *dz* will be calculated.

$$dx = x - x' \tag{1}$$

$$O = \sqrt{x^2 + y^2 + z^2} \tag{2}$$

$$D = \sqrt{dx^2 + dy^2 + dz^2} \tag{3}$$

Equation (2) and (3) show how to calculate the real-time overall acceleration O and overall differential D.

As shown in Fig. 1, in State 1, O and D will be stored into Overall List and Diff List separately. The two lists have the same capacity and are designed to store data for five seconds. Five seconds is long enough for the system to monitor the possible fall. Before acceleration O is put into the Overall List, it is compared to a threshold T. If O is larger than T, it is a possible fall candidate, the position index in the Overall List is put into a Queue Q, which helps us find the candidate position in Overall List when starting analyzing data. There is another constraint before putting into Q that any two position indexes in the Queue cannot be too close to each other. The reason is that the acceleration peak caused by a fall will normally last for 0.5 second, which means all the data during the fall process may be larger than T. To decrease the unnecessary check, the constraint is necessary.

The system will analyze data when the Overall List is full. There are two cases:

Case 1: If no peak is detected and Q is empty, the system clears the lists and keep on working.

Case 2: If Q is not empty, it goes to State 2 as shown in Fig. 1, the system gets the element E in the queue, find the peak position P in Overall List, which can be found by finding the max value position in a certain time range (0.5s) starting from position E. Then the system starts monitoring the Diff List from position P. It divides the monitor period (3s) into 100 sub-periods. If there are over 95 still periods among 100 sub-periods, the system alerts. The system has another threshold T_D for monitoring user state, if the values in Diff List during a sub-period do not reach T_D, the sub-period is regarded as a still period. If the system does not alert, the system finds next element in Q until Q is empty or the monitor time is smaller than the necessary duration (3s). Then the system clears the lists before last element in the queue.

This design ensures the real-time data processing. If the fall happens, the system will detect the fall and alert after the monitor period, no later than three seconds.

B. *Alert Design*

Considering many dangerous fall events happen when the elderly live alone, our App is designed to connect to certain social media such as Twitter, Facebook and Weibo. The user can login to his/her account. When the fall is detected, the alarm of the smartphone will first alarm, the GPS of the phone is used to report the location of the possible fall as well as the messages for help on the social media.

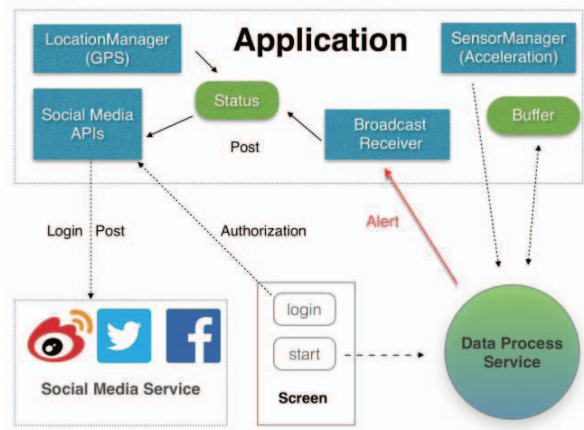

Fig. 2. Android fall detection app component structure.

C. *Android Application Implementation*

The system is implemented on the Android platform. It reads data from the motion sensors of LG Nexus 4 smartphone with Android version 4.4.3. The GPS service of the smartphone is also used.

The data processing process will be implemented in the format of service, which will run in the background. The component structure of the application is shown in Fig. 2. LocationManager provides the GPS information, the Social Media APIs connects to social media such as Twitter or Weibo. The SensorManger provides the real-time acceleration data to the service. When the system detects dangerous falls, it will broadcast an alert intent. The App registering a broadcast receiver receives the alert and asks for help on social media. The main GUI will give user the access to log in to its social media and access to start fall detection service or stop it.

III. CONCLUSION

We build our real-time fall detection system on the smartphone based on the low-complexity algorithm in our previous research, which only uses accelerometer. In its Android-based implementation, we add the alert design to connect to social media. In the future, the real-time system can be applied into smaller-size products to detect falls in real-time.

REFERENCES

1. W. Qu, F. Lin, A. Wang and W. Xu, "Evaluation of a low-complexity fall detection algorithm on wearable sensor towards falls and fall-alike activities," in 2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), pp. 1-6, December, 2015. IEEE.
2. G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955.
3. N. Noury, "A smart sensor for the remote follow up of activity and fall detection of the elderly," in Microtechnologies in Medicine & Biology 2nd Annual International IEEE-EMB Special Topic Conference on. IEEE, 2002, pp. 314–317.
4. J. Davis, M. Robertson, M. Ashe, T. Liu-Ambrose, K. Khan, and C. Marra, "International comparison of cost of falls in older adults living in the community: a systematic review," Osteoporosis International, vol. 21, no. 8, pp. 1295–1306, 2010.