Implementation of a Real-Time Human Activity Classifier Using a Triaxial Accelerometer and Smartphone

Na Li, Yibin Hou, Zhangqin Huang
Embedded Software and Systems Institute, Beijing University of Technology, nana439@emails.bjut.edu.cn; {ybhou, zhuang} @ bjut.edu.cn

Abstract
The real-time monitoring of human activity can provide valuable information regarding an individual’s degree of functional ability and lifestyle. This paper presents a real-time activities classification algorithm associated with the data acquired from a wearable triaxial accelerometer. The algorithm distinguishes steady state and non steady state. Periodic activities including walking, running and being still are recognized as steady state, and transition between different steady states is regarded as non steady state, from which we detect dangerous falling. Based on physical characterization of activities, we abstract the time domain features, such as change of SVM and continuous weightless time, then use Kalman filter to classify steady states, and a decision tree to detect falling event from the state transitions. All these processes run on an Android smartphone. The algorithm is unconstrained with the sensor orientation with respect to the body. Furthermore, targeting personal habits, the thresholds used for classifying periodic activities are automatically adapted by performing a short bout of processes. A set of trials involving five subjects during different days were undertaken, and the results indicate an overall accuracy of 97% across a series of tasks involving a variety of activities related to daily life. The algorithm exhibits excellent outcomes in activities recognition.

Keywords: Triaxial Accelerometer, Activities Recognition, Kalman Filter, Decision Tree

1. Introduction
Technical progress in microelectronics provides a new low-weight, low-power and miniaturized inertial sensors. Meanwhile, today’s mobile phones come equipped with an increasing range of computational, storage and communication resources. Many researches took advantage of these advancements in areas such as monitoring of daily life activities, which contributes to form a better habit for individual and gives elderly people help when emergency happens. In this scenario, a key challenge of these systems is to develop algorithms that synthesize the available data to compute higher level inferences, representations of human activities and context-possible in real-time, and communicate these higher level inferences to the cloud [1]. Now some products are emerging, e.g.: UbitFit [2], CenceMe [3], which all use accelerometer to infer the user’s activity.

Many researches on activity recognition have proved that good accuracy in inferring the daily activity can be gotten by using triaxial accelerometer [4]. These systems mainly employ multiple accelerometer units placed at various body sites to assist in their detection of activities such as walking, ascending stairs, descending stairs, falling and so on [5][6]. Most studies on this topic have carried out in two steps: features exaction and recognition process. As for abstracting features, such as mean, variance, SVM (signal vector magnitude), SMA (signal magnitude area), FFT coefficients and frequency spectrum are commonly used [7][8]. Every feature has its limitation, thus, multiple features (in both time and frequency) form feature vectors to use [9]. The recognition methods also exhibit diversity, such as fixed-threshold methods [5][6], pattern recognition strategies [10], conventional or fuzzy logic [11], and artificial neural networks [12][13]. In actual practice, there is no method that is superior to others well. For some periodic activities, such as: running, walking, cycling, studies usually use SMA, coefficient of FFT and power spectral entropy as features [14][15]. For unexpected aperiodic activity like falling, which always happens in a short time, some severe cases need to alert others and ask for help immediately, so it must meet real-time requirement. Some features such as virtual velocity, tilting angle were abstracted as features to detect the orientation change of body, which depend on the gyroscope or the accelerometer’s sensitive axis to sense orientation change [16][17]. For some activities, such as: having a bath, brushing teeth, driving car, some studies use other sensors to provide
background information, e.g.: microphone [1][3]. Because of complexity of human activities, there is lack of generalized approach to recognize all activities [18].

There are a few of issues to recognize human activity online by using accelerometer and smartphone. First, some features are relevant to sensor’s placement or orientation, which require fixing one sensing axis oriented toward the gravity. Second, it is difficult to choose the best features in advance; complex computing is always used to get a high dimension feature, which then leads to many parameters to be learned, and brittle training tires or resulting model. Third, complexity of the algorithm becomes an issue when continuous running on mobile phone.

Purpose of the paper is to recognize periodic activity (walking, running and being still) and detect dangerous falling event by using single triaxial accelerometer. Here, dangerous falling means that human can not return to his/her normal behaviors over a period of time after impacting on the ground.

This paper divides states into two classes: steady state and non steady state. Periodic activity is regarded as ‘steady state’, which is a general situation that the whole body is being for a relatively long period of time, for example: continuous standing, running, walking and so on. When there is a different activity, it means there is a state transition, which is regarded as ‘non-steady state’, for example: walking-to-running, running-to-falling. With respect to the previous work, the main strength of the algorithm includes three points: first, it divides human state into two classes according to human’s daily life and recognizes dangerous falling from non steady state, which tells us the context of falling; second, threshold of peak of SVM is used to recognize periodic activity and its value is auto-adaptive with user’s personal data; third, less calculating of features and using of decision tree are efficient in doing classification on the battery-powered smartphone.

The rest of paper is organized as follows. Section 2 presents detailed design of the recognition algorithm. Section 3 presents our experiment implementation. Section 4 presents an evaluation of the algorithm. Section 5 concludes the paper with a summary and the future work.

2. Methods

2.1. Feature Extract

The three-axle output of accelerometer must be transformed from the body reference frame to the inertial reference frame when human’s posture changes. In this paper, we use the signal vector magnitude (SVM), which is independent of the orientation of sensor. SVM reflects the instantaneous intensity of human movement at time $t$, which can be represented as equation (1) [4]:

$$ SVM(t) = \sqrt{a_{x,t}^2 + a_{y,t}^2 + a_{z,t}^2} $$  

(1)

Figure 1. (a) SVM when running; (b) SVM when walking

Figure 1 describes trend of SVM when subject is walking or running (Accelerometer was placed on the chest of body and sampling rate is 20HZ, all the following figures are the same setting.). Body gets an upward force from ground after foot impacts ground, which makes instantaneous acceleration increase and leads to bigger SVM. Impacting strength varies from different activities, so every activity has a specific range of SVM, but the range has tendency to stable during each periodic activity. Therefore, change of SVM (CSVM) is used as feature to recognize the steady state and non steady state, as equation (2).

$$ CSVM(t) = Max(SVM(t)) - Min(SVM(t-k)) $$  

(2)
We calculate CSVM by recognizing the monotone increasing and decreasing trend in change of SVM. For instance, when SVM is in a monotone increasing and a new sample is greater than former one, SVM is still increasing, we ignore this sample and wait for the next one and then repeat checking. But if it is less than the former one, we realize that the monotone increasing is end and mark the former sample as peak. By the counter way, we can collect the bottom of SVM, and then obtain the value of CSVM by calculating the difference between the adjacent peak and bottom in real time.

With regard to falling, patterns are different in daily life. Based on human’s health condition when falling, this paper divides it into two classes: falling with awareness or not. From the point of acceleration, major difference between them is number of peak. When falling with awareness, subject will reach part of body, e.g.: knee or arm, to impact on the ground first, then upper body impacts the ground. This process reflects subject’s instinct to protect him; but when falling without awareness, whole body impacts on ground, without instinct protection. Experiments demonstrate this process. Figure 2 demonstrates subject falling with awareness and not respectively (When falling, subject lay on a mattress with 15cm thick). In (a), two peaks correspond to action that subject’s knee knocks on the ground and upper body impacts the ground in sequence. In (b), one peak value corresponds to action that subject falls on the ground like a rigid body.

![Figure 2](image)

**Figure 2.** (a) SVM when falling forward with awareness; (b) SVM when falling backward without awareness;

From figure 2, we can find that falling always associates with strong impact on ground, which leads to bigger SVM instantaneously than normal activities. So feature (1) is abstracted:
(1): peak of SVM which is bigger than threshold;

In this paper, threshold of peak of SVM is simply written th(S) and set 2.5g; sample which satisfies feature (1) is written as $p_{SVM_{th}}$, shown in figure 2. According to appearance of change of SVM, jumping is similar to falling with awareness. Figure 3 describes change of SVM when subject jumps.

![Figure 3](image)

**Figure 3.** SVM when jumping awareness

In figure 3, two big peaks correspond to action that subject presses the feet into ground to jump, and lands on the ground again. Continuous weightless time before first peak corresponds to the process that subject bends legs to prepare bounce. Comparing figure 3 with figure 2 (a), we can find that they are different at the continuous weightless time between two peaks. When jumping, continuous weightless time between two peaks corresponds to the time period that human is on air. When falling, there is
almost no continuous weightless time between two peaks. The reason is that the second impact happens very quickly after the first impact, which leads to the second peak almost immediately. So we abstract feature (2):

(2): Continuous weightless time between $SVM_p(t)$ and its adjacent peak, which is written: $T_{weightless}(t)$.

Threshold of feature (2) is simply written as $th(T)$ and is set 0.2 seconds in this paper. In this study, we focus on the situation that subject falls and can not recover to normal himself/herself in a short time. So if subject impacts the ground and keeps motionless for a while, this subject probably needs to ask for help from others. So feature (3) is abstracted:

(3): Average value of SVM (t) during time T after $SVM_p(t)$, which is simply written: $ASVM(t)$;

Threshold of feature (3) is simply written $th(A)$. Its value is relevant with sensor’s offset. The calculating methods will be introduced in section 2.2. Here, T is set 5 seconds. Subject is usually from upright to horizontal position when falling. The basic technique employed to perform such situation relies on evaluation of human’s tilting angle $\theta$.

Suppose $SVM_p(t)$ appears at time $t$, we use tilting angle between acceleration vector at antecedent $T_1$ seconds ($t - T_1$) and acceleration at subsequent $T_2$ seconds ($t + T_2$), as shown in figure 2. Therefore, tilting angle is used as feature (4):

(4): Tilting angle $\theta$ between acceleration vector at time ($t - T_1$) and acceleration vector at time ($t + T_2$).

\[
\theta = \arcsin \left( \frac{\| A(t-T_1) \ast A(t+T_2) \|}{SVM(t-T_1) \ast SVM(t+T_2)} \right)
\]

$A(t-T_1) = [a_{x(t-T_1)}, a_{y(t-T_1)}, a_{z(t-T_1)}]$: acceleration vector at time $t - T_1$;

$A(t+T_2) = [a_{x(t+T_2)}, a_{y(t+T_2)}, a_{z(t+T_2)}]$: acceleration vector at time $t + T_2$; $\ast$: outer product of vector; $\cdot$: multiplication.

In this paper, $T_1$ is set 1 second and $T_2$ is set 2 seconds. Threshold of feature (4) is simply written $th(\theta)$ and set 50 degree. If subject’s tilting angle is from 0 to $th(\theta)$, body will be regarded still upright, whereas value more than $th(\theta)$, indicates a lying posture.

2.2 Classification Algorithm

This paper adopts second-by-second classification scheme, whereby activity is classified based on the data collected over a 1 second interval. An evaluation of the most suitable size for such a classification widow was presented in [19], where it was determined that a window of around 1 second (0.8 s-1.4 s) was optimal. Classification keywords are assigned to each second of time, according to the algorithm shown in figure 4.

1) Steady-state and Non-steady state: Value of CSVM reflects the trend of change of acceleration. The bigger one means change of acceleration is much larger, which always indicates a state transition. In order to fit the trends, we use Kalman filter to dispose CSVM. When processing, two CSVM are added, one of which corresponds to the falling edge and the other to the rising edge between the bottoms of SVM. Result is regarded as quantity of state ($X_n$):

\[
X_n = F(CSVM_i) = \sum_{i=1}^{\infty} CSVM_i \quad (n=2)
\]

$X_n$ is processed by Kalman filter and output value $X'_n$ is regarded as posterior of quantity, which is used to identify state of body. Figure 4 shows results dealt by Kalman filter.

Figure 4. (a) result of $X'_n$ dealt by Kalman filter when continuous running
(b) result of $X'_n$ dealt by Kalman filter when continuous walking
According to the experiments and analysis, we can get the following results:
(1) \( \mathcal{X}_1 \leq 0.2 \text{g}, \) body stays a static state;
(2) \( (X'_i - X'_{i-k}) < 0.5 \text{g}, \forall k \in [0...J], \) body stays a steady state, that means human repeats an activity;
(3) \( (X'_i - X'_{i-k}) > 0.5 \text{g}, \forall k \in (0...J], \) body stays a non-steady state, that means there is a state transition.

In this paper, we recognize walking, running, and being still from steady state. In experiment, range of \( \mathcal{X}_1 \) is different between walking and running, so we set different threshold to recognize them.

(1) if \( \mathcal{X}_1/2 \in [\text{th(W)}], \) then human is walking;
(2) if \( \mathcal{X}_1/2 \in [\text{th(R)}], \) then human is running;

For periodic activity (walking/running) recognition, we adapted the algorithm to personal characteristic automatically online after the subject performs a short bout of activities. In practice, when the subject acts in a steady state, we set initial value for the above four thresholds, which means each periodic activity having a broad range, for example, initial range of walking is set 0.5g~2g. After 10 seconds, we find the range is about 1.2g~1.8g, then this new range is marked and data is collected continuously for the next 10 seconds. These processes are carried out for about 120 seconds, if all \( \mathcal{X}_1 \) during this time satisfy the marked range, then this range will replace the initial value, that is: value (1.2g~1.8g) will replace initial value (0.8g~2g); otherwise, algorithm will return to the previous step and continue to look for better range of \( \mathcal{X}_1 \). By this process, threshold of these two activities can be adjusted according to the subject. This adaptive algorithm is able to improve reorganization accuracy.

2) Fall detection: Falling happens on state transition. A binary decision tree is used to detect falling, which requires only a few comparisons and thus consuming less battery power compared with complex classifiers [20], shown in figure 5. The tree is structured including four nodes, which corresponds to the features in section 2.1. Featured signal samples with larger values than the threshold fall into the right branch, while the smaller ones fall into left branch. Node 1 corresponding to feature (1) utilizes intensity of the higher peak to discriminate activities containing intense impact from normal activities. Node 2 corresponding to feature (3) utilizes active duration to discriminate situation that can return to normal himself/herself after falling. Node 3 corresponding to feature (4) uses tilting angle before and after falling to discriminate orientation change. Node 4 corresponding to feature (2) discriminates jumping and falling on sofa or bending down fast (in less than 2 s) with the help of continuous weightless time.

![Figure 5. Binary decision tree used for fall detection on smartphone](image-url)
3. Experimental Trials and Results

3.1 Instrumentation

The activity recognition system is made of three main components: sensor (a triaxial accelerometer), wireless transmission unit and processing unit. Although many smart phones equip with accelerometer, considering some parameters, e.g.: sample rate, resolution are different with model and vendor of phone, so we use the external sensor. In this paper, we choose ADXL345 (by Analog Devices Inc.), which is a digital 3-axis acceleration measurement sensor. Parameters are chosen as follows: measure range: -8g~+8g; I2C interface and 10-bit resolution. We regard body as an entire, so the sensor is located in the upper part of the trunk or at waist, where activity intensity is easier to detect without effect from dynamic four limbs.

Bluetooth chip’s model is BC5 (by CSR Inc.). There are two units in BC5: MCU and DSP. DSP reads data from accelerometer (there is a FIFO buffer in sensor, which can hold 32 groups of data), then transmits it to MCU. To balance realtime request and consumption of transmission, we pack four sets of data into one package with a prefix ‘A’, then transmit them. The specific prefix ‘A’ is necessary when we increase number of sensors. One set of data consists of 3-axes acceleration, which is 30 bits. So 128 bits are transmitted every 200 ms.

The central device to process data collection and activity recognition is a smartphone with Android system (HTC). Its performance parameter is following: CPU: 1.0GHZ, memory: 512 MB, vision of Android is 2.3. The smartphone receives data over Bluetooth, computes feature from raw data online, classifies the data in second-by-second basis online, and stores the data on a SD card.

3.2 Experimental Setup

A series of trials were carried out to assess the performance of the algorithm, ensuring all path of the flowchart (figure 4) were adequately evaluated. Five healthy adult subjects (25 to 35 years old, no obvious disease on legs) were asked to perform predefined activities, including daily activities, falling forward, backward, sideward. Because falling may cause seriously injures to the older, they were not taken into account in the experiments. All these activities required subject to repeat activities three...
times and finish them in different days. The sensor and Bluetooth were bound together and worn on the subjects’ belt or on the chest, without orientation requirement, smartphone was on subject’s hand or put in his/her pocket. During these trials, all raw data was transmitted to the smartphone for processing and display.

In this paper, we do not take rolling from bed, chair or down satire into account. The activities on steady state are: walking, running and still. For the walking tasks, subject was asked to walk with ordinary speed (about two steps one second) for about five minutes. For the running tasks, subject ran for about five minutes with speed 1.5 m/s~2.0 m/s. For falling task, mattress used was approximately 15cm thick with spring and was placed directly onto the floor. Before falling, subject was asked to do normal activities, e.g.: walking, running so on. For stumbling task, subject simulated this activity as if he was tripped by something. For falling into a sofa or chair, subject simulated it after standing. In experiment, subject jumped from ground and did different height.

3.3 Algorithm Performance

To measure the performance of our recognition approach, we choose accuracy rate, response time, and complexity to evaluate the algorithm. The definition of accuracy rate is as below:

\[
\text{Accuracy} = \frac{\text{tp}}{\text{all performed activities}}
\]  

True positive (tp): the number of correctly recognized cases for activities that were really performed;

For the walking and running tests, normal and fast-paced of them were classified at least 98% accuracy. There were problems with the recognition of walking when subject impacted ground very heavily on purpose. Under this situation, peak of SVM may attain threshold of running activity. In extreme situations, if subject skipped, this was recognized running.

In this paper, accuracy of falling means it is recognized as falling class, but can not identify what patterns it belongs to. When a recovery, e.g.: moving around, standing up or walking to a nearby seat was performed, we supposed the subject did not need help, and it was not recognized as dangerous falling. With regard to the situation of falling into a sofa/chair, the algorithm could not discriminate whether falling with dangerous or not. Because sometimes SVM could not attain threshold when subject fell on the chair, which were recognized as normal activity.

To measure the real-time performance, we measure response time when falling happened. In fact, there are several time intervals: falling event elapsed time, data transmitting time, data unpacking time, algorithm running time. Because there is not standard on beginning and ending of falling event, we define as follows (on 20HZ sample rate premises): Assuming impact happens and value of SVM reaches th(S) at \(t_{\text{fall-start}}\) time, 100 packages of data finish transmitting subsequent at \(t_{\text{fall-end}}\) time, running time of algorithm is \(t_{\text{algorithm-run}}\), response time is defined as equation (6) :

\[
t_{\text{response-time}} = t_{\text{fall-end}} - t_{\text{fall-start}} + t_{\text{algorithm-run}}
\]

We use timestamp method to get data transmitting and unpacking time. After the socket connection established, we set parameter \(t_{\text{start}}\) before reading one package and \(t_{\text{end}}\) after finishing reading and unpacking, and then we get transmitting time of 100 packages: \(t_{\text{data-transmit}}\), as shown in equation (7):

\[
t_{\text{data-transmit}} = \sum_{i=1}^{100} (t_{\text{end}}(i) - t_{\text{start}}(i))
\]

So the average transmitting and unpacking time of 100 sets data is 5300 ms, and average running time of algorithm is 3 ms, so response time is 5303 ms.

To evaluate the complexity of algorithm, we measure CPU load, memory footprint and battery life. Because we implemented a few functions such as user interface, sending alert to remote service, heart rate and temperature chart, there are several threads when recognition algorithm runs. About 37100KB main memory are occupied. The CPU load is shown in table 1:

<table>
<thead>
<tr>
<th>Thread name</th>
<th>CPU load</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data transmitting, processing and storing</td>
<td>6%</td>
</tr>
</tbody>
</table>
We evaluate the battery consumption in the following situation: all functions enabled (including functions in the table 3), luminance of screen of smartphone set to minimum, without incoming call. The battery of smartphone can last about 7 hours.

4. Discussion

A simple and effective online classifier for activity recognition is presented. Testing of the algorithm reveals its ability to recognize walking, running, being still, dangerous falling and jumping with a high degree of accuracy. There is occasional miss when detecting dangerous falling but the results are still acceptable. Overall, the algorithm provides a reasonable illustration of the activities performed by subject. The wearable devices can be a client of the intelligent community system, which serves a group of people and provides early warning etc.

Developing appropriate methods to deal with falls into chair/sofa or fall from downstairs requires further study. When the subject falls into a chair/sofa without awareness, sometimes algorithm cannot distinguish it from normal activities exactly. But if subject has awareness after falling, other methods can be used. e.g.: subject presses alarm button on the smartphone or calls for help via sound. In this situation, more functions are required to add, e.g.: background service.

Laboratory experiments only involve healthy participants, although elderly and chronically ill individuals need such aid. Experiments excluded these people for safety, so further experimentation is required to evaluate the performance of the activity classification algorithm on this target group.

Processor speed does not prove to be a constraint in the algorithm. This is simply because smartphone develops fast, and the ability of processor is enough for our algorithm. But memory size and ability of wireless transmission need to be considered. If we add other functions, e.g.: display, processing of other sensors, it was easy to result in unexpected termination of application before we tuned the application to balance the multi tasks. If the processor and memory was sufficiently, it would be possible to implement an algorithm to recognize more activities and achieve more accurate and/or detailed classifications, such as Bayesian Networks to classify various falls and specific state transition.

Power consumption is another major problem on mobile computing, but it could undergo a major reduction if we use optical algorithm. Currently the continually transmitting and processing raw data from sensor consumes much power. Further optimized implementation will use data-cycle pipeline, which abandons similar raw data and uses low power mode according to activity recognized.

5. Conclusion

A simple and effective activity classification algorithm is present based on the data collected from a single, triaxial accelerometer and implemented on smartphone, which is used in a real-time environment. Features extraction is independent of sensor’s orientation with respect to the body. Some activities have been recognized, such as, walking, running, staying still and dangerous falling. Experiments are carried out on five subjects and the algorithm recognized activities with accuracy higher than 90%. In the future, we will take other technique, e.g.: admission control, duty cycling, to facilitate long periods of data collecting and processing on mobile phone.

6. References

[2] Sunny Consolvo, David W. McDonald, Tammy Toscos, Mike Y.Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, “Activity sensing in


