



# Collaborative Fall Detection Using Smart Phone and Kinect

Xue Li<sup>1</sup> · Lanshun Nie<sup>1</sup> · Hanchuan Xu<sup>1</sup> · Xianzhi Wang<sup>2</sup> 

Published online: 18 January 2018  
© Springer Science+Business Media, LLC, part of Springer Nature 2018

## Abstract

Humanfall detection has attracted broad attentions as sensors and mobile devices are increasingly adopted in real-life scenarios such as smart homes. The complexity of activities in home environments pose severe challenges to the fall detection research with respect to the detection accuracy. We propose a collaborative detection platform that combines two subsystems: a threshold-based fall detection subsystem using mobile phones and a support vector machine (SVM)-based fall detection subsystem using Kinects. Both subsystems have their respective confidence models and the platform detects falls by fusing the data of both subsystems using two methods: the logical rules-based and D-S evidence fusion theory-based methods. We have validated the two confidence models based on mobile phone and Kinect, which achieve the accuracy of 84.17% and 97.08%, respectively. Our collaborative fall detection approach achieves the best accuracy of 100%.

**Keywords** Fall detection · Collaborative detection · Kinect · Smart phone

## 1 Introduction

The world has entered the ageing society where many old people live in solitary with diseases or difficulty getting about. Home accidents such as falls surge in recent years and account for a large portion of medical cost worldwide. The lack and high expense of aged care call for automatic detection of home accidents like falls. Fortunately, the advances of network and embedded technologies have made it possible to detect accidental falls of elderly people via portable electronic devices. Generally, a system detect human falls by implementing fall detection algorithms that analyze the signals of those devices: when the elderly exhibits an abnormal behavior, the system

immediately pushes an alert with the collected information to the monitoring personnel or rescue center through wireless networks. In this way, the system can request for treatment of any injuries caused by falls automatically and thereby reduce damages to the elderly regardless of their locations and living conditions.

Although the concept of fall is a common sense of humans, describing falls for machines can be difficult, making the detection of falls by machines even more difficult. Currently, most fall detection systems use a single device or a number of homogeneous sensors to monitor human activities. For this reason, the fall detection accuracy of these systems is strictly confined by the limitations of a single device type (e.g., a Kinect might be occluded by an object and an intelligent device can be restricted by its deployed location). In contrast, collaborative sensing, i.e., fusing the data from multiple types of devices, can potentially yield more reliable detection results. A typical collaborative sensing procedure may involve collecting data from different types of sensors, fusing these data, feeding data to an integral algorithm, and finally, applying some predefined threshold is to detect falls based on the algorithm's results.

Among the existing fall detection approaches, non-wearable devices have attracted increasing attentions in the field of collaborative fall detection. Non-wearable devices avoid people from the inconvenience of wearing various devices on bodies and address the low accuracy of fall detection caused by using only a single type of devices: it is

---

✉ Xue Li  
lixuecs@hit.edu.cn

Lanshun Nie  
rhine.nie@foxmail.com

Hanchuan Xu  
xhc@hit.edu.cn

Xianzhi Wang  
xzwang@smu.edu.sg

<sup>1</sup> Harbin Institute of Technology, Harbin, China

<sup>2</sup> Singapore Management University, Singapore, Singapore

uncommon for people to wear multiple types of devices all serving the same purpose whereas not the case for non-wearable devices in practice. Based on the above insights, we aim to leverage the advantages of different types of devices for more accurate fall detection in complex home environments. In particular, we propose a collaborative fall detection approach that collects data and extract features from different types of devices. The approach builds confidence models to give the data from each sensor type a confidence value and detects falls by combining the confidence values from different sensor types at the decision layer. To put this idea into reality, we have established a collaborative detection platform, which includes two subsystems: a threshold-based fall detection subsystem using mobile phones and a SVM-based fall detection subsystem using Kinects. Each subsystem has its own confidence model and the platform uses data fusion methods to detect fall based on the results of the two subsystems collaboratively. In a nutshell, we make the following contributions in this paper:

1. We proposed a collaborative system that leverages two types of devices, namely mobile phones and Kinects, to address the fall detection problem in a home environment;
2. We evaluate our proposed approach via experiments in real-world scenarios and show the improved fall detection accuracy and reduced false alarm rate of the proposed approach when compared to the approaches using single types of devices.

The remainder of this paper is organized as follows. We review the related work in Section 2. Section 3 presents our collaborative fall detection approach that use confidence models to leverage the merits of multiple sensor types. Section 4 reports the experiments and discusses the experimental results. Finally, Section 5 gives the concluding remarks.

## 2 Related work

A typical fall detection system has three parts: sensor data collection, fall detection, and action after successful fall detection. A real-time fall detection system works as follows: first, devices send motion data to the processing unit; next, the fall detection system uses algorithms to capture falls; finally, the system takes actions such as playing certain tones to attract attention from nearby or sending alert information to the family members or caregivers. This alert information should include the event, location, direction, falling state (conscious/unconscious), and timestamp [1]. The performance of a fall detection system is usually measured by sensitivity (SE) and specificity (SP) [2–4].

## 2.1 Devices

There are three categories of devices commonly used for fall detection: wearable or carried sensors, vision-based sensors, and environment sensors. Wearable and carried devices (e.g., smart mobiles) use embedded three-axis acceleration sensors to monitor static state and postures [5] or to detect falls [6, 7]; vision-based sensors perceive the change of an object's state as well as the space and time to analyze whether a person falls [8]. At present, cameras [9–11] and Kinect [12, 13] are two most popular types of visual devices used to detect falls; environment sensors generally use vibration [14], audio [15, 16], or radio frequency technologies (such as RFID [17]) for fall detection. For example, the work in [18, 19] analyzes the signal strength of the RFID devices to recognize activities of elderly people.

## 2.2 Fall detection methods

There are several types of multi-sensor data fusion algorithms: missing data fusion, associated data fusion, inconsistent data fusion, and heterogeneous data fusion [20, 21]. The work in [22] summarizes the challenges faced in multi-sensor based fall detection and analyzes the current detection methods and data fusion methods. The method in [23] first judges if the angular velocity is greater than the threshold, and then fuse the acceleration sensor data and angular velocity sensor data (in the form of 0 or 1) to distinguish falls from other daily activities. Multi-wavelet pixel level fusion and D-S evidence theory decision level fusion are proposed and applied to recognition systems in [24]. A new method based on multi-sensor fusion is also proposed based on [25] and applied to measure soil moisture content. The work in [26] studies analyses massive, heterogeneous, real-time, and uncertainty networking data and puts forward an algorithm based on the weighted D-S evidence theory for Internet of Things heterogeneous sensor data fusion. An improved D-S theory is applied to the field of water quality monitoring and fire monitoring [27, 28].

## 2.3 Collaborative fall detection

Our surveys show that existing fall detection systems are based on either a single device or multiple homogeneous sensors and most of the fall detection system experiments are carried out in laboratory environments. Generally, these systems fuse the multiple sensor data into a single algorithm and use the threshold method to carry out activity recognition [29, 30]. The work in [31] proposes to use three types of devices, 3D camera, wearable MEMS acceleration sensor, and microphone to detect falls in smart homes, but it gives no details about the fusion of data from the different devices. The work in [32] combines the three-axis acceleration sensor and an atmospheric pressure sensor for fall detection, which uses

the threshold method to combine the sensor data. The work in [23] propose a multi-sensor data fusion method that determines whether the value of the angular velocity exceeds a given threshold before fusing the acceleration sensor and angular velocity sensor data to classify daily activities. In [20], multi-sensor data fusion algorithms are classified into fusion of defect data, fusion of associated data, fusion of non-uniform data, and fusion of different types of data. The work in [33] detect falls using multiple devices, but the devices used are all of the same type rather than of different types. For this reason, the work cannot overcome the short comings of a single device type.

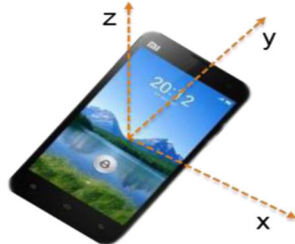
### 3 Method

As afore mentioned, a single type of device have limitations for fall detection. Therefore, we proposed to collaboratively use two types of devices, namely mobile phones and Microsoft Kinects, for fall detection in our approach.

Our multi-device collaborative fall detection system consists of two subsystems: the smart phone-based fall detection subsystem and the Kinect-based fall detection subsystem. The two subsystems collect and analyze their respective data using different algorithms. After that, the calculated data and confidence values are sent to the Netty server through the Internet. Finally, the system uses these data and the confidence model for data fusion and judges whether the elderly has encountered an abnormal behavior and whether to take emergency measures. Figure 1 shows the workflow of our proposed collaborative fall detection system.

#### 3.1 Threshold based fall detection using smartphone

The smart phone-based fall detection algorithms collect human activity data from smart phones’ built-in three-axis acceleration sensors. The collected data are smoothed using media filters to reduce noises, external shocks, and collision. The algorithm calculates three feature values using the acceleration data: Signal Magnitude Area (SMA), Signal Magnitude Vector (SMV), and Tilt Angle (TA) [34, 35].



The details as following:

SMA: this feature depicts the magnitude of the change in human activity and is used to distinguish the time interval between user activity and rest. The greater the value, the greater the change of movement.

$$SMA = \frac{1}{t} \left( \int_0^t |x(t)| dt + \int_0^t |y(t)| dt + \int_0^t |z(t)| dt \right)$$

where X (T), y (T) and Z (T) represent the sampling values of X, y, and Z axes, respectively.

SMV: the characteristic test results determine the most suitable parameter as the threshold of SMV, which represents the instantaneous activity intensity.

$$SMV = \sqrt{x_i^2 + y_i^2 + z_i^2}$$

TA: the angle between the Y axis and the vertical direction. If TA is below the threshold of 40 degrees, it is classified as falling or lying.

$$TA = \arcsin \left( \frac{y_i}{\sqrt{x_i^2 + y_i^2 + z_i^2}} \right)$$

We use the MATLAB simulation software to simulate the values of SMA, SMV and TA, and statistical methods to determine their thresholds. The algorithm uses the thresholds for real-time fall detection. Since the objective is to detect falls, other activities such as standing, lying or vertical activities are considered false positives.

#### 3.2 SVM-based fall detection using Kinect

##### 3.2.1 SVM-based fall detection

We use support vector machine (SVM), the frequently used classification method in the machine learning field, for fall detection using Kinect. For a given training set label pair  $(x_i, y_i)$ , where  $i = 1 \dots n, x_i \in R_n, y_i \in \{1, -1\}$ , there are the following optimization problems:

If the training set is linearly separable, the hyper plane can be directly solved by the maximum distance. Otherwise, the data needs to be mapped from low-dimensional space to high-dimensional space to

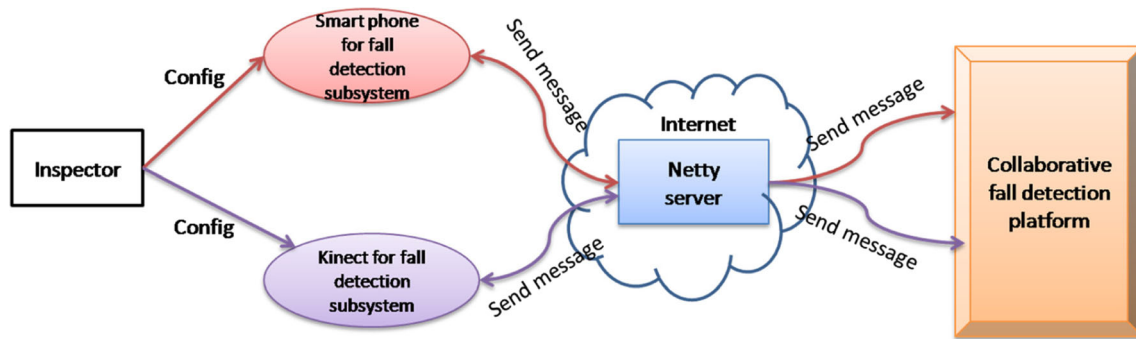


Fig. 1 Workflow of the collaborative fall detection system

complete the data conversion process and to solve the optimal hyper plane.

$$\min_{w,b,\varepsilon} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (1)$$

$$\begin{aligned} s.t. & y_i(w^T \varphi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \quad (2)$$

where  $C > 0$  is the penalty parameter for the misclassification and  $\xi$  is the slack variable.

There are four basic functions for mapping low-dimensional spatial data to high-dimensional spatial data for the SVM:

- (1) Linear kernel function:  $K(x_i, x_j) = x_i^T x_j$ .
- (2) Polynomial kernel function:  $K(x_i, x_j) = \exp(-\gamma x_i^T x_j + r)$ ,  $\gamma > 0$
- (3) Radial Basis Function (RBF):  $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$ ,  $\gamma > 0$
- (4) Multi-layer perceptual kernel function (sigmoid):  $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

We use the libSVM library developed by Professor Lin Zhiren [36] and SVM's default radial basis function as the kernel function. The radial basis function has the following advantages: 1) it can well handle the non-linear relationship between the sample labels and attributes; 2) it has a relatively small number of parameters and requires little effect and complexity to calculate; 3) the numerical constraints have less influence on the use of the radial basis kernel function due to  $0 < K(x_i, x_j) \leq 1$ , the polynomial kernel function value may not converge ( $\gamma x_i^T x_j + r > 1$  or  $\gamma x_i^T x_j + r < 0$ ), and the multilayer perception kernel function is restricted due to trigonometric function definition; otherwise, it would be meaningless at some point [37].

### 3.2.2 Fall detection using Kinect

The fall detection method uses a USB interface to connect to a PC to process the sensor data from Kinect. To detect the head movement and analyze the head speed, the data from Kinect must be converted into geographical coordinates. Since the Kinect can convert the color and the depth information to geographical coordinates based on the camera coordinate system, the Kinect-based detection method can directly use the head coordinate information extracted from the RGB image and the depth image. Initially, the coordinates of the head are the same as the coordinates of the camera coordinate system. Based on the coordinates of the head, the coordinate system performs behavior recognition according to the human head's movement speed. The calculation of speed is based on tracking the trajectory of the head of the human skeleton. Therefore, the system continuously detects both heads to calculate the speed and then use this speed to judge whether the elderly person has fallen.

The calculation of head speed follows the following steps:

- (1) Coordinate transformation: this step extracts the coordinates of the camera coordinate system from the Kinect depth information, thereby converting the coordinates of the camera coordinate system to geographical coordinates. Suppose the camera head under the frame of reference coordinates are  $[x, y, z]$ , then the geographical coordinates are  $[X, Y, Z]$ . Depending to the camera position, there might be two coordinate system transformations.
- (2) Speed calculation: the speed is calculated by comparing the displacement between the two head coordinates divided by the time difference between the two head coordinates. First, the system obtains data of the converted coordinate system and determines whether there are previous data. If true, two coordinate system data will be compared and calculated through the z-axis and xy plane displacement data and divided by the time difference between the two datasets. After that, we obtain the speed

value for the z-axis and xy axis, and use the SVM algorithm to judge the behavior of the elderly according to the calculated speed data.

The system adopts SVM to analyze speed characteristics and identify the human behavior to tell whether the human body is falling. The overall functional block diagram of the Kinect detection system is shown in Fig. 2.

### 3.3 Data fusion model

The data fusion aims to process data from multiple data sources to overcome the unreliability issue caused by one-side observation of a single data source. Such unreliability may cause instability and inaccuracy of the processing system. Multi-sensor data fusion can avoid these problems or reduce the impact of these problems. Therefore, the decision-making system based on multi-sensor data generally have better accuracy and have a more comprehensive judgment to monitor the target. The following describes the principles and levels of data fusion, as well as common data fusion algorithms.

#### (1) The principles of data fusion

Data fusion collects sensor data from different sources, extracts features from these data, and correlated the features related to the same observation target to fuse the original data using data fusion algorithms [38].

#### (2) The level of data fusion

A typical multi-sensor data fusion technology has three levels of integration: data layer integration, feature layer integration, and decision-making integration.

#### (3) Common data fusion algorithm

There are several categories of multi-sensor data fusion methods: the statistical inference method, the decision theory

method, signal processing and the estimation theory method, the geometric method, the information theory method, and the artificial intelligence method. Statistical inference methods include Bayesian inference, D-S evidence reasoning, etc. The methods of signal processing and estimation include the weighted average method and the Kalman filter. The information theory method includes the entropy method and the minimum description length method. Typical artificial intelligence methods include neural network and rule-based reasoning.

Considering the difference between the two kinds of sensor data, we adopts the decision-making data fusion method to integrate the data. In this paper, two concrete methods of decision-making fusion are used. The first is the result-oriented, logical, rule-based, collaborative data fusion method. The second is to use the D-S evidence theory to calculate fall data.

### 3.4 Confidence model

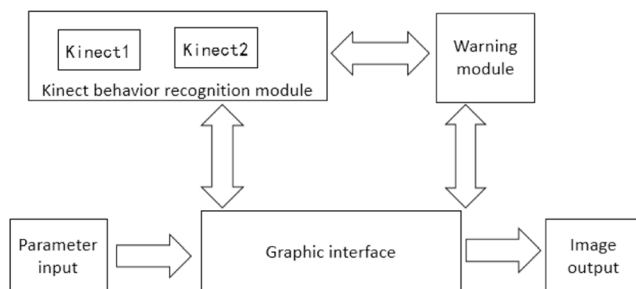
Using a single device has limitations in fall detection. For example, the Kinect can be restricted by shade objects and smart phones can be restricted by location restrictions. Since different fall detection algorithms can be affected by varied factors, we define confidence models to combine them. In our proposed approach, each device extracts the features of the collected data and obtain a result according to the respective algorithm. After then, the decision-making data fusion based on the evidence theory or logical rule is used to fuse the statistics and the result data of each sensor.

Since the data sent from each subsystem to the PC monitoring side may not be always credible, we establish a confidence model to calibrate the collect data. A confidence value is represented as a percentage value in the range of [0, 1]. A higher conference value indicates the higher performance of the device. Devices in different data collection environments may have different confidence because the sensors have different efficacy and the confidence model enables the data from different sensors to compensate the vulnerability of one another for the higher overall credibility.

### 3.5 Collaborative fall detection

#### 3.5.1 Logical rules-based collaborative fall detection

According to the physical condition of the elder in the family and the specific needs of the corresponding technical services, our collaborative fall detection system are designed with two service models: “fall dully” and “fall sensitively”, and the



**Fig. 2** The overall functional block diagram of the Kinect detection system

collaborative fall detection system should meet the needs of various family members.

**Fall sensitively:** this situation may avoid omission but mistake daily activities as falls. Guardians in this situation need only lift the alarm. This model is suitable for those with osteoporosis, a fear of falling, or poor physical condition.

**Fall dully:** This situation may reduce the false alarm rate but miss falls. This model is suitable for the elder whose physical condition is better or the elder who is more able to exercise independently.

Suppose we derive the fall detection result of each device as one of two status values, “fall” and “non-fall”. The results of different devices can be fused using simple logical operations. Suppose there are  $n$  devices, the result of the fall test for the  $i$ -th device is  $R_i$ , with a fall being set to True,

and a non-fall being set to False. The  $i$ -th device performs the fall detection accurate rate as  $Ad_i$ , and descends with the  $Ad_i$ . The results are fused using two devices with high accuracy of fall detection, assumed to be  $Ad_1$  and  $Ad_2$ . For the user chooses ‘Fall sensitively,’ the result of the  $t$ -th result  $R_t = R_{Ad_1} | R_{Ad_2}$  is the final fusion result; conversely, if the user chooses ‘Fall dully,’ the result of the  $t$ -th result is  $R_t = R_{Ad_1} \& R_{Ad_2}$ . If the fusion result is False, the result whose detection accuracy which is the highest will act as the  $t$ -th test results, and will otherwise output fusion results. The system will be based on the user’s choice to serve the user.

The confidence model is applied to the rule-based collaborative fall detection algorithm. The algorithm is described in the following.

The rule-based collaborative fall detection algorithm:

---

**Input:** The user inputs *serviceMode*, Device set  $D$ , fall detection result set  $R$ ;

**Output:** collaborative fall detection results

---

For each  $d_i \in D$

Calculate the  $d_i$  fall detection accuracy as  $Ad_i$ , according to the confidence model

End for;

Sort  $\langle Ad_1, Ad_2, \dots \rangle$  in descending order

If *serviceMode* == “Fall sensitively” then

$$R_t = R_{Ad_1} | R_{Ad_2}$$

Else if *serviceMode* == “Fall dully” then

$$R_t = R_{Ad_1} \& R_{Ad_2}$$

If  $R_t == \text{False}$  then

$$R_t = R_{Ad_1}$$

End if;

End else if;

End if;

Return fusion results  $R_t$ .

---

According to the logic rules of the collaborative fall detection method, the confidence model is able to identify and predict falls based on the real-time monitoring results of the situation, calculate the accuracy of the fall, change the corresponding confidence, and to forecast for the next step.

### 3.5.2 D-S evidence theory based collaborative fall detection

The D-S evidence theory is a decision-making data fusion method. It defines a series of operations and concepts to enable the system to work strictly under uncertain conditions. The method can combine the confidence

values from various sources [39] to fuse data in a collaborative system. In view of the advantages of the D-S evidence theory in handling uncertain data information, we uses this method to carry out the fall data fusion. The following describes the general method of D-S evidence theory.

Suppose each sensor has two state values A and B, and all possible state values are denoted by  $\Theta$ , then  $\Theta$  can be expressed by the following Eq. (3):

$$\Theta = \{A, B, \{A, B\}, \{\phi\}\} \tag{3}$$

where the subset  $\{A, B\}$  represents either A or B. Each subset gives a certain weight, representing the probability that the state is correct. For the sensor data, the probability value is related to the sensor itself. The probability of A is represented by  $m(A)$ , as defined in Eq. (4).

$$\begin{cases} m(\Phi) = 0 \\ \sum_{A \in \Theta} m(A) = 1 \end{cases} \tag{4}$$

There are two other computational methods, beliefs and plausibility for the elements in  $\Theta$ . The specific calculation is shown by Eq. (5) and (6).

$$Belief(A) = \sum_{E_k \subseteq A} m(E_k) \tag{5}$$

$$Plausibility(A) = \sum_{E_k \cap A = \phi} m(E_k) \tag{6}$$

where the probability falls in the range of  $[Belief(A), Plausibility(A)]$ .

In the data fusion environment, each sensor has three measurement methods, for each element A in  $\Theta$ , there are probability  $m(A)$ , belief(A), and plausibility(A). The values of the different sensors can be fused with the authority of the Dempster-Shafer rule, as shown in Eq. (7).

$$(m_i \oplus m_j)(A) = \frac{\sum_{E_k \cap E_{k'} = A} m_i(E_k) m_j(E_{k'})}{1 - \sum_{E_k \cap E_{k'} = \Phi} m_i(E_k) m_j(E_{k'})} \tag{7}$$

Since Eq. (7) is controversial in validity and produces counter-intuitive results in face of conflicts among sources, we defile rules in Eq. (8) to alleviate the limitations of the Dempster-Shafer theory.

$$\begin{cases} m'(A) = \frac{1 - (1 - m_i(A))(1 - m_j(A))}{1 + (1 - m_i(A))(1 - m_j(A))} \\ (m_i \oplus m_j)(A) = \frac{m'(A)}{\sum m'(A)} \end{cases} \tag{8}$$

Based on the experimental results, the confidence model takes the accuracy of the device fall detection as the initial value. Then it updates the device’s fall detection preparation rate and finally uses the D-S evidence theory to carry out the fall test results of data fusion and make the final decisions If the user changes the location of the device after the monitor is started, the update confidence module can still work according to the set process to obtain the new detection accuracy rate.

The workflow of the D-S evidence theory combined with the confidence model is described in Fig. 3.

### 4 Evaluation

In this section, we report our experiments in a smart home environment to test the accuracy of the collaborative platform in detecting falls. The following subsections will introduce the experimental settings using students as testers, the evaluation of different fall detection subsystems, and the evaluation of different methods for collaborative fall detection, respectively.

#### 4.1 Experiment settings

We simulated a smart home environment using smart phones and Kinects to verify the effectiveness of our approach in real-world scenes. The information of sensors is shown in Table 1. Sensor information. (we show all the tables in the Appendix for better clarity). For the experiments, the same person fell

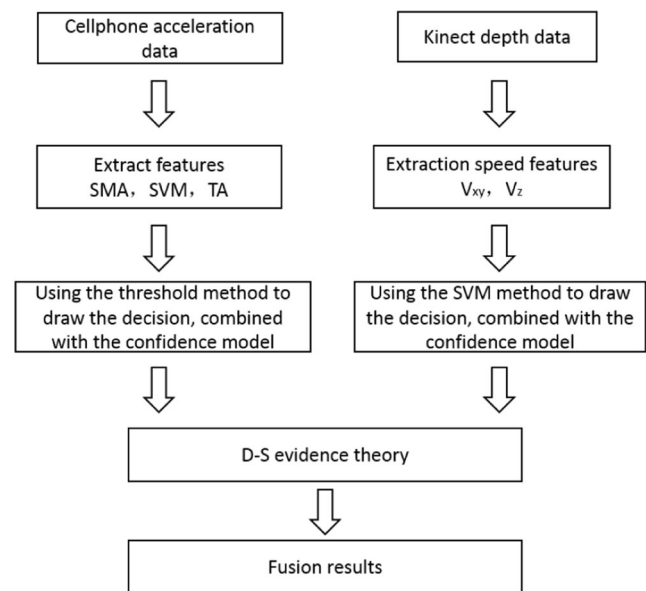


Fig. 3 The D-S evidence theory combined with the confidence model

and performed other daily activities about 500 times, and these activities included *Forward fall*, *Backward fall*, *Left falls*, *Right falls*, *Squat stand up*, and *Walking*.

## 4.2 Evaluation methods

We use the criteria ([https://en.wikipedia.org/wiki/Evaluation\\_of\\_binary\\_classifiers](https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers)) to evaluate the performance of the fall detection method.

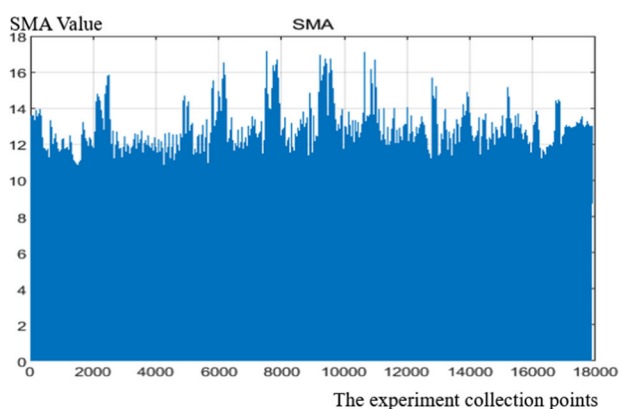
### 4.2.1 Threshold based fall detection using the smartphone subsystem

The smart phone was installed in the pocket and tied to the waist, and the program would be done for each action twice. The acceleration information was stored in the SQLite database, and we used the data in the database and MATLAB for simulation. In this experiment, we set the sampling frequency of the mobile phone acceleration sensor which was 50Hz,

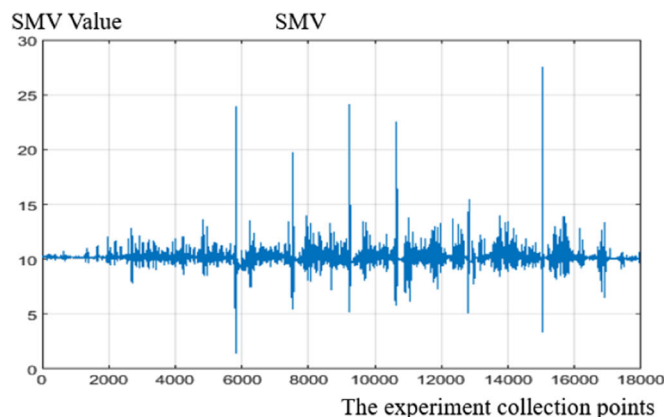
because the daily main operating frequency was lower than 20 Hz.

The experimenter was asked to complete two consecutive movements within 10s. Using the result of MATLAB simulation, we calculated the Signal Magnitude Area (SMA), SMV, and Tilt Angle (TA), which are shown in Fig. 4a, b, and c, respectively. The results of the following simulation are the number of data acquisition points, where the vertical axis is the specific value and the Tilt Angle is the radian value. Both the Signal Magnitude Area and the Signal Magnitude Vector are calculated.

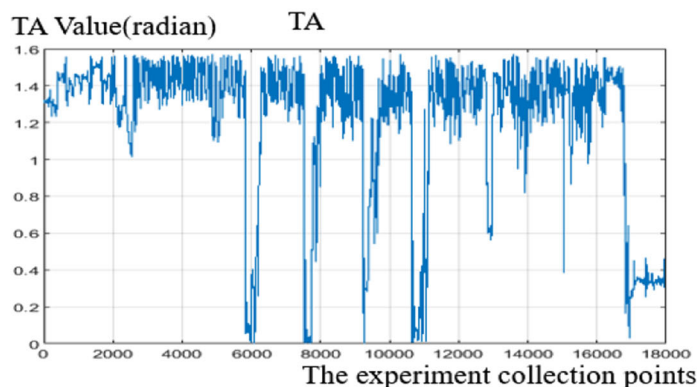
Figure 4 is in accordance with the experimental program to do the normal action simulation map, from these figures can be roughly differentiated behavioral action. To better determine the threshold of each eigenvalue, we tried to sample the two actions of fast squat and quick sit-down because of the similarity of the eigenvalues of fast squatting, fast sitting and falling. The sampling results were simulated using MATLAB, as shown in Fig. 5. The results can be seen from Fig. 4, the angle of inclination of fast squat and fast sit



(a) Signal Magnitude Area (SMA)



(b) Signal Magnitude Vector (SMV)



(c) Tilt Angle (TA)

Fig. 4 a Signal Magnitude Area (SMA). b Signal Magnitude Vector (SMV). c Tilt Angle (TA)



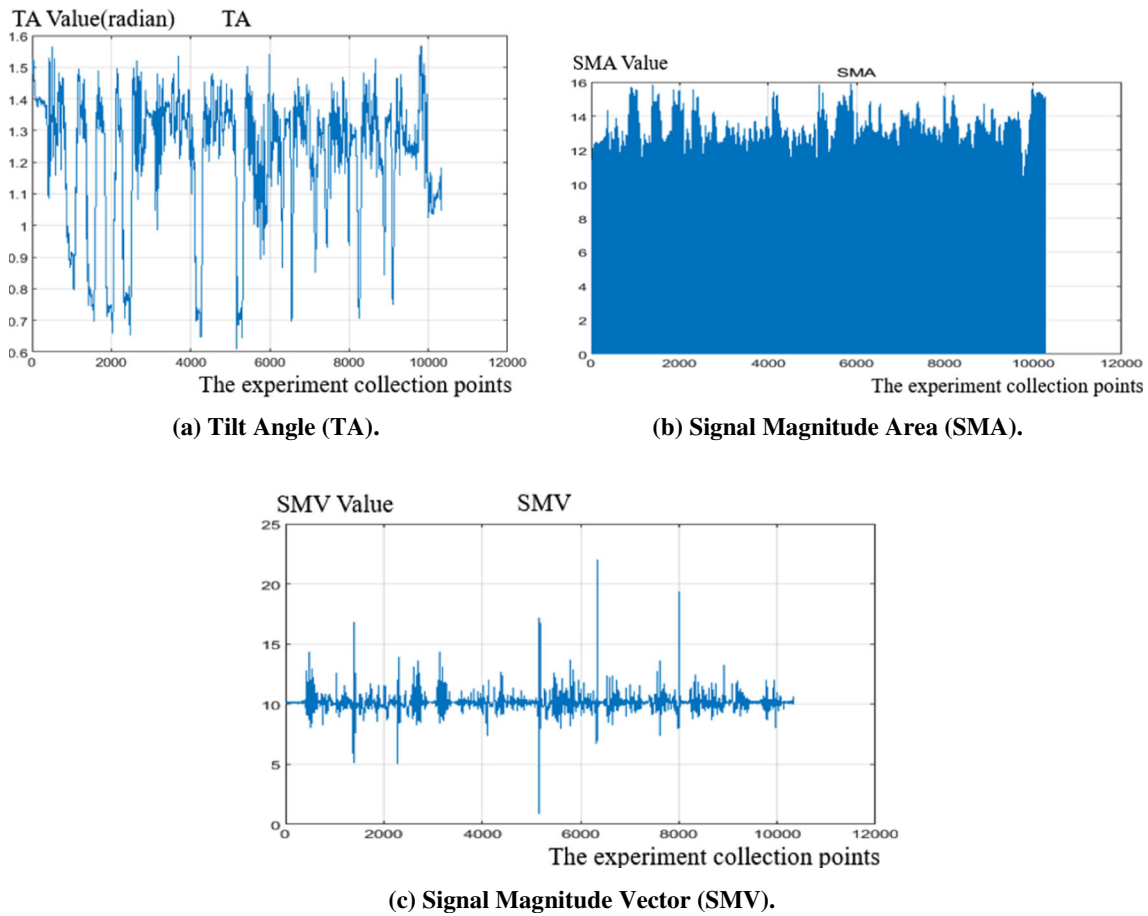


Fig. 5 a Tilt Angle (TA). b Signal Magnitude Area (SMA). c Signal Magnitude Vector (SMV)

is greater than 0.6 (radian). From the experimental data, we can see that the SMV value of the fast squat and fast sit down will be relatively large, which is similar with the SMV value when falling, so it may produce miscarriage of judgement.

The threshold of the three eigenvalues is obtained by statistics over the simulation results and used to carry out

the fall test and to test the performance of the method. In accordance with the experimental design for four different falls, each action was done 30 times. Recording the actual fall but did not detect the number of times, and then perform a single performance test according to the performance of the evaluation algorithm.

Kinect and smart phones test the fall detection at the same time, the phone was placed on various parts of the body in the experiment, and the results of the fall test were counted respectively. Table 2. Fall detection results (smart phone tied to the waist). shows the statistics of the mobile phone on the waist. The statistics of the mobile phone placed in the loose trousers pocket are shown in Table 3. The statistical results of the mobile phone placed in the tight trousers pocket are shown in Table 4. The statistics of the mobile phone placed in the pocket are shown in Table 5.

#### 4.2.2 SVM based fall detection using the Kinect subsystem

This paper captured the fall and non-fall data of the 2000 times respectively, the figure was drawn by MATLAB as shown in Fig. 6, which counts the fall and non-fall speed

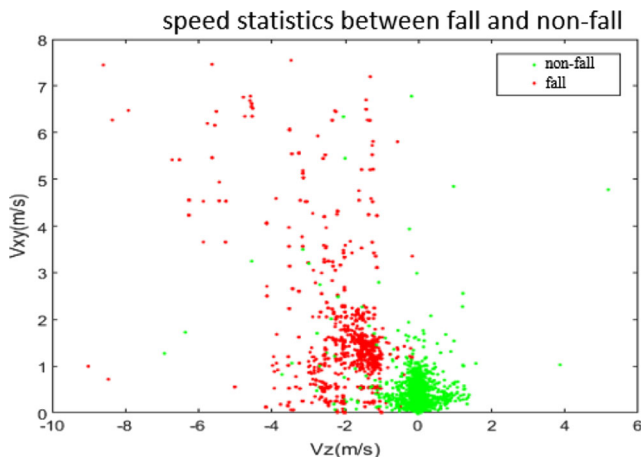
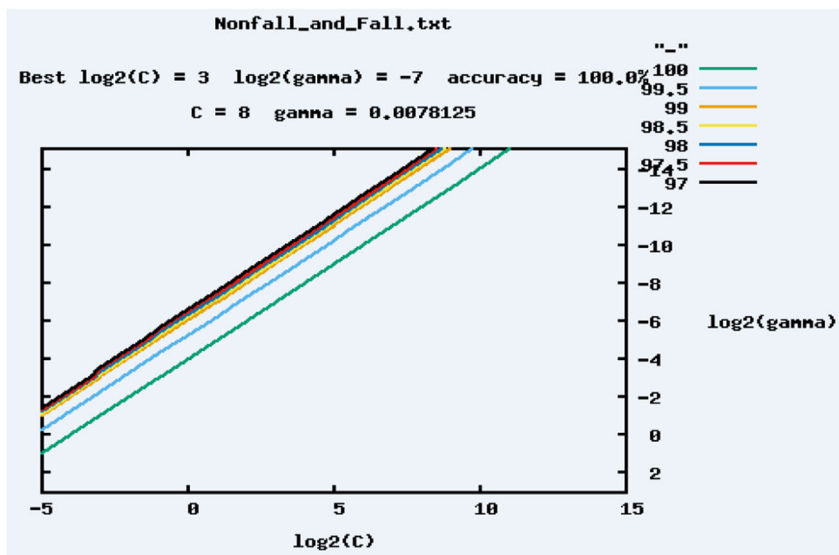


Fig. 6 Fall and non-fall speed statistics

Fig. 7 Fall and non-fall C and  $\gamma$  values and recognition rates



characteristics. The figure shows that the fall and non-fall speed characteristics are relatively obvious, so it can use the support vector machine method to carry out the classification of behavior.

First, this experiment recorded the speed of the experimental staff movement of the head according to the experimental staff doing different actions, the characteristics of this experiment are Z-axis speed  $V_z$  and X-axis and Y-axis of the combined speed  $V_{xy}$ , the classification was judged according to the two features. The type of action has four types of labels that fall and non-fall. And then this system uses SVM to study these two features, the model was in accordance with behavior prediction. Because this system uses support vector machines and Radial kernel function, it is necessary to determine the parameters C and  $\gamma$  through experiments.

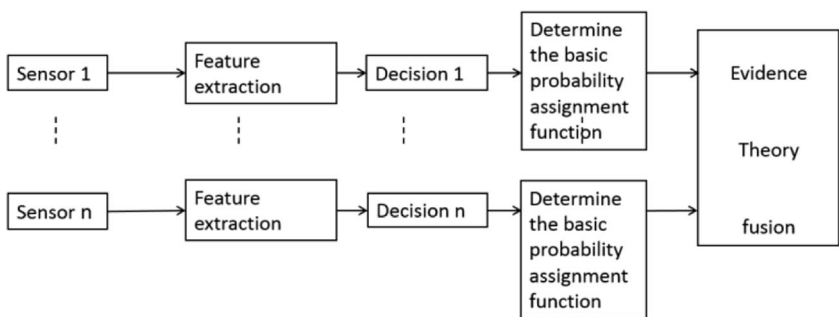
This paper captured the fall and non-fall data of the 2000 times respectively, the non-fall data as a group, fall data as a group, using libSVM svm scale to normalize the data. Through the grid.py to optimize the training set Parameter selection, and

get the fall and non-fall C,  $\gamma$  values and recognition rate, the result is shown in Fig. 7. The system uses the optimization parameters, and they are used in SVM training the model. At last, we use the model to predict real-time behavior.

Using the model developed to predict results, the fall experiment was conducted according to experimental design. Forward falls, backward falls, left falls, right falls and Squat stand up were conducted 30 times respectively when the Kinect is within the recommended range in 4 min, and we record the results. And then we performed the performance test according to the method of the evaluation algorithm performance. The performance test results are shown in Table 6. This table is the statistics of Kinect overall fall detection accuracy, because the mobile phone was used together with the fall detection in the experiment, so when the phone placed in various locations, they are corresponding to the different fall detection performance.

From the table also can be seen, the walking behavior has the phenomenon of false positives, on one hand, the reason is for the SVM algorithm, the amount of data trained by smaller

Fig. 8 Data fusion procedure based on the D-S evidence theory for fall detection



data, on the other hand, in the experiment, due to experimental equipment placed, Kinect can only shoot to Upper body.

#### 4.2.3 Logical rules based collaborative fall detection

The data are processed according to the collaborative method described below.

If the user selects the ‘Fall sensitively’ mode, the fusion results in the four cases are shown in Table 7.

If the user selects the ‘Fall dully’ mode, the fusion results in the four cases are shown in Table 8.

The above fusion results show that when the fall sensitively mode is selected, the accuracy of the fall detection is obviously improved, the equipment with higher accuracy is improved, the false alarm rate is improved, the equipment with lower accuracy rate, the false alarm rate reduces. However, when the ‘Fall dully’ is chosen, the method assumes higher detection accuracy of equipment, leading to the improved accuracy and reduced false alarm rate.

#### 4.2.4 D-S evidence theory based collaborative fall detection

In this section, we present the fall detection method based on D-S evidence theory to fuse the data from various devices and to improve the accuracy of the fall detection. The data fusion process based on the D-S Evidence Theory for fall detection is shown in Fig. 8.

First, the mobile phone’s built-in acceleration sensor collects acceleration data and extracts features, followed by the threshold method determining whether the user falls. In particular, the system gives the threshold of a decision value while calculating the confidence of the phone in a specific position, where this confidence is the mobile phone fall down detection accuracy. Meanwhile, Kinect begins to collect the movement of human skeletal head node speed and extract the speed  $V_{xy}$  and  $V_z$  these two eigen values. The two eigen values are used for training. Finally, SVM is used to predict the two kinds of features, and the results of the fall detection are obtained, and the accuracy of Kinect fall detection is calculated. Table 9 shows the detection accuracy when the mobile phone is placed at different parts of the body and Kinect corresponding to the detection accuracy of the fall. This accuracy rate can provide evidence to the evidence theory for applying this method for data fusion.

For the fall detection, this paper is only concerned about the falls of home users, so the identification of home users is  $\Theta = \{A, B\}$ , where A is the fall behavior, and B is the non-fall behavior. According to the D-S evidence theory formula, the final fusion result is shown as Table 10.

### 4.3 Discussion

The comparison shows the fusion method based on logical rules and D-S evidence theory achieve good accuracy and false positive rate. Considering the fusion results of our experimental data and fusion methods, the rule-based fusion results have higher fall detection accuracy and lower false alarm rate. The premise is that these two devices can have better complementarity in the fall detection process, i.e., if a device does not detect the fall, while another device does. The accuracy of data fusion based on the D-S evidence theory is determined by not only this complementarity but also the accuracy of each device itself.

In summary, by differentiating the confidence on the two types of devices according to their detection results, the collaborative approach is able to fuse their results and achieve better performance. In the next step, we want to create the model, which can use confidence values (smart phone, Kinect, or both) adaptively according to location of the elderly.

### 5 Conclusion

In this paper, we present a novel collaborative fall detection method that uses different types of sensors, namely smart phones and Kinects, based on confidence models. Given a complex home environment with isolated devices, we address the deficiencies of existing approaches by establishing a platform for collaborative fall detection, which can collaborate with a variety of devices to detect falls and to provide users more convenient and reliable fall detection services. Our proposed approach has the following advantages:

- Reasonable fusion of multiple sensors can significantly improve the accuracy and reduce the false alarms.
- The fusion is straightforward and thus has low complexity (we only use the basic decision results to fuse).
- Our approach represents a promising approach for multi-sensor integration and collaboration under the circumstance of the increasing use and diversity of sensors.

For the future work, we will consider overcoming the limitations of devices by selecting the appropriate types of devices and data fusion methods. Besides, fall detection is only the first step towards a collaborative complex family environment. We plan to further improve the system to enable adaptive settings and fall detection.

**Acknowledgments** Funded by the 2014 Microsoft Research Asia Collaborative Research Program.

## Appendix

**Table 1** Sensor information

Name	Version	Application	Description	Quantity
Smart phone	Mi2	Collect human movement acceleration information	Using TCP communication	1
Microsoft Kinect	Kinect 1517	Collect the head of the person movement speed	Using TCP communication	1

**Table 2** Fall detection results (smart phone tied to the waist)

Action	TP	FN	FP	TN	SE	SP	AC
Forward fall	26	4	0	0	86.67%	0	86.67%
Backward fall	25	5	0	0	83.33%	0	83.33%
Left falls	26	4	0	0	86.67%	0	86.67%
Right falls	24	6	0	0	80%	0	80%
Squat stand up	0	0	2	28	0	93.33%	93.33%
Walking	0	0	0	Walking 1 min	–	–	–

**Table 3** Fall detection results (smart phone placed in a loose trousers pocket)

Action	TP	FN	FP	TN	SE	SP	AC
Forward fall	15	15	0	0	50%	0	50%
Backward fall	30	0	0	0	100%	0	100%
Left falls	15	15	0	0	50%	0	50%
Right falls	22	8	0	0	73.33%	0	73.33%
Squat stand up	0	0	3	27	0	90%	90%
Walking	0	0	0	Walking 1 min	–	–	–

**Table 4** Fall detection results (smart phones placed in a tight trousers pocket)

Action	TP	FN	FP	TN	SE	SP	AC
Forward fall	22	8	0	0	73.33%	0	73.33%
Backward fall	30	0	0	0	100%	0	100%
Left falls	19	11	0	0	63.33%	0	63.33%
Right falls	11	19	0	0	36.67%	0	36.67%
Squat stand up	0	0	2	28	0	93.33%	93.33%
Walking	0	0	0	Walking 1 min	–	–	–

**Table 5** Fall detection results (smart phone placed in a coat pocket)

Action	TP	FN	FP	TN	SE	SP	AC
Forward fall	20	10	0	0	66.67%	0	66.67%
Backward fall	23	7	0	0	76.67%	0	76.67%
Left falls	19	11	0	0	63.33%	0	63.33%
Right falls	16	14	0	0	53.33%	0	53.33%
Squat stand up	0	0	7	23	0	76.67%	76.67%
Walking	0	0	0	Walking 1 min	–	–	–

**Table 6** Fall detection results using the Kinect

Action	TP	FN	FP	TN	SE	SP	AC
Forward fall	116	4	0	0	96.67%	0	96.67%
Backward fall	120	0	0	0	100%	0	100%
Left falls	116	4	0	0	96.67%	0	96.67%
Right falls	114	6	0	0	95.00%	0	95.00%
Squat stand up	0	0	14	106	0	88.33%	88.33%
Walking	0	0	0	Walking4min	–	–	–

**Table 7** Fall sensitively collaborative statistical results

Methods and their statistical indicators	Waist	Loose trousers pocket	Tight trousers pocket	Coat pocket
Kinect accuracy	97.5%	96.67%	95.83%	98.33%
Kinect fallfalse rate	3.33%	4.44%	6.11%	1.67%
The smart phone accuracy	84.17%	68.33%	68.33%	65.00%
The smart phone fall false rate	11.67%	22.78%	22.22%	27.22%
Logical rules accuracy	100%	98.33%	99.17%	100%
Logical rulesfall false rate	2.78%	5.00%	4.44%	4.44%

**Table 8** Fall dully collaborative statistical results

Methods and their statistical indicators	Waist	Loose trousers pocket	Tight trousers pocket	Coat pocket
Kinect accuracy	97.5%	96.67%	95.83%	98.33%
Kinect fallfalse rate	3.33%	4.44%	5.56%	1.67%
The smart phone accuracy	84.17%	68.33%	68.33%	65%
The smart phone fall false rate	11.67%	22.78%	22.22%	27.22%
Logical rules accuracy	97.5%	96.67%	95.83%	98.33%
Logical rulesfall false rate	3.33%	4.44%	5.56%	1.67%

**Table 9** The accuracy of the two methods in different situations

Methods and their statistical indicators	Waist	Loose trousers pocket	Tight trousers pocket	Coat pocket
The Kinect	97.08%	97.08%	97.08%	97.08%
The smart phones	84.17%	68.33%	68.33%	65%

**Table 10** The accuracy of D-S evidence theory fusion results

Methods and their statistical indicators	Waist	Loose trousers pocket	Tight trousers pocket	Coat pocket
Based on D-S evidence fusion results accuracy	97.5%	96.67%	95.83%	98.33%
Based on D-S evidence fusion results erroneous judgement	3.33%	4.44%	5.56%	1.67%

## References

1. Pannurat N, Thiemjarus S, Nantajeewarawat E (2014) Automatic fall monitoring: a review. *Sensors* 14(7):12900–12936
2. Igual R, Medrano C, Plaza I (2013) Challenges, issues and trends in fall detection systems. *Biomed Eng Online* 12(1):66–66
3. Jian H, Chen H (2015) A portable fall detection and alerting system based on k-NN algorithm and remote medicine. *China Commun* 12(4):23–31
4. Kau L, Chen C (2015) A Smart phone-based pocket fall accident detection, positioning, and rescue system. *IEEE J Biomed Health Inform* 19(1):44–56
5. Rakhman AZ, Nugroho LE (2014, November) u-FAST: ubiquitous fall detection and alert system for elderly people in smart home environment. In: 2014 Makassar international conference on electrical engineering and informatics (MICEEI), IEEE, pp 136–140
6. He Y, Li Y, Yin C (2012) Falling-incident detection and alarm by smartphone with Multimedia Messaging Service (MMS). *E-Health Telecommun Syst Netw* 1:1–5
7. He Y, Li Y, Bao SD (2012, January) Fall detection by built-in tri-accelerometer of smartphone. In: 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), IEEE, pp 184–187
8. Mubashir M, Shao L, Seed L (2013) A survey on fall detection: principles and approaches. *Neurocomputing* 100:144–152
9. Rougier C, Meunier J, St-Arnaud A, Rousseau J (2011) Robust video surveillance for fall detection based on human shape deformation. *IEEE Trans Circ Syst Video Technol* 21(5):611–622
10. Auvinet E, Multon F, Saint-Arnaud A, Rousseau J, Meunier J (2011) Fall detection with multiple cameras: an occlusion-resistant method based on 3-d silhouette vertical distribution. *IEEE Trans Inf Technol Biomed* 15(2):290–300
11. Pan Z, Liu S, Fu W (2017) A review of visual moving target tracking. *Multimedia Tools Appl* 76(16):16989–17018
12. Stone EE, Skubic M (2015) Fall detection in homes of older adults using the Microsoft Kinect. *IEEE J Biomed Health Inform* 19(1):290–301
13. Mastorakis G, Makris D (2014) Fall detection system using Kinect's infrared sensor. *J Real-Time Image Proc* 9(4):635–646
14. Pierleoni P, Belli A, Maurizi L, Palma L, Pernini L, Paniccia M, Valenti S (2016) A wearable fall detector for elderly people based on ahrs and barometric sensor. *IEEE Sensors J* 16(17):6733–6744
15. Li Y, Banerjee T, Popescu M, Skubic M (2013, July) Improvement of acoustic fall detection using Kinect depth sensing. In: 2013 35th annual international conference of the IEEE, engineering in medicine and biology society (EMBC), pp 6736–6739
16. Li Y, Ho KC, Popescu M (2012) A microphone array system for automatic fall detection. *IEEE Trans Biomed Eng* 59(5):1291–1301
17. Torres RLS, Wickramasinghe A, Pham VN, Ranasinghe DC (2015, June) What if your floor could tell someone you fell? A device free fall detection method. In: Conference on artificial intelligence in medicine in Europe, Springer, Cham, pp 86–95
18. Yao L, Sheng QZ, Li X, Gu T, Tan M, Wang X, Zou W (2017) Compressive representation for device-free activity recognition with passive RFID signal strength. *IEEE Trans Mob Comput PP*(99):1–1
19. Yao L, Sheng QZ, Ruan W, Gu T, Li X, Falkner N, Yang Z (2015, August) Rf-care: device-free posture recognition for elderly people using a passive rfid tag array. In: Proceedings of the 12th EAI international conference on mobile and ubiquitous systems: computing, networking and services on ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp 120–129
20. Khaleghi B, Khamis A, Karray FO, Razavi SN (2013) Multisensor data fusion: a review of the state-of-the-art. *Inf Fusion* 14(1):28–44
21. Liu S, Pan Z, Song H (2017) Digital image watermarking method based on DCT and fractal encoding. *IET Image Process* 11(10):815–821
22. Koshmak G, Loutfi A, Linden M (2016) Challenges and issues in multisensor fusion approach for fall detection: review paper. *J Sensors* 2016:6931789
23. Andò B, Baglio S, Lombardo CO, Marletta V (2016) A multisensor data-fusion approach for ADL and fall classification. *IEEE Trans Instrum Meas* 65(9):1960–1967
24. Kannan R, Garg A (2015, November) Adaptive sensor fusion technology for mobile and wearable applications. In: *Sensors*, 2015 IEEE, pp 1–4
25. Pugh M, Brewer J, Kvam J (2015, April) Sensor fusion for intrusion detection under false alarm constraints. In: *Sensors applications symposium (SAS)*, 2015 IEEE, pp 1–6
26. Zhou B, Yao Y (2015, November) Decision-level sensor-fusion based on DTRS. In: *International conference on rough sets and knowledge technology*, Springer International Publishing, pp 321–332
27. Peng H, Cao X (2010, December) Research conflict problems of DS evidence and its application in multi-sensor information fusion technology. In: 2010 I.E. international conference on IEEE, information theory and information security (ICITIS), pp 747–750
28. Liu S, Pan Z, Cheng X (2017) A novel fast fractal image compression method based on distance clustering in high dimensional sphere surface. *Fractals Complex Geom Patterns Scaling Nat Soc* 25(23):1740004
29. Liu S, Pan Z, Fu W, Cheng X (2017) Fractal generation method based on asymptote family of generalized Mandelbrot set and its application. *J Nonlinear Sci Appl* 10(3):1148–1161
30. Liu S, Fu W, He L, Zhou J, Ma M (2017) Distribution of primary additional errors in fractal encoding method. *Multimedia Tools Appl* 76(4):5787–5802
31. Leone A, Diraco G, Distante C, Siciliano P, Malfatti M, Gonzo L, Libal V (2008, October) A multi-sensor approach for people fall detection in home environment. In: *Workshop on multi-camera and multi-modal sensor fusion algorithms and applications-M2SFA2 2008*
32. Tolkiehn M, Atallah L, Lo B, Yang GZ (2011, August) Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor. In: 2011 annual international conference of the IEEE engineering in medicine and biology society, EMBC, pp 369–372
33. Nadee C, Chamnongthai K (2015, December) Multi sensor system for automatic fall detection. In: *Signal and information processing association annual summit and conference (APSIPA)*, 2015 Asia-Pacific, pp 930–933
34. Madansingh S, Thrasher TA, Layne CS, Lee BC (2015, October) Smartphone based fall detection system. In 2015 15th international conference on control, automation and systems (ICCAS), IEEE, pp 370–374
35. Gimpel T, Kiertscher S, Lindemann A, Schnor B, Vogel P (2015) Evaluation of threshold-based fall detection on android smartphones. In: *HEALTHINF*, pp 598–604
36. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. *ACM Trans Intell Syst Technol (TIST)* 2(3):27
37. Hsu CW, Chang CC, Lin CJ (2003) A practical guide to support vector classification
38. Mitchell HB (2007) *Multi-sensor data fusion: an introduction*. Springer Publishing Company, Incorporated. <https://doi.org/10.1007/978-3-540-71559-7>
39. Wu H, Siegel IM, Stiefelhaber R, Yang J (2002) Sensor fusion using 819 Dempster-Shafer theory [for context-aware HCI]. In: *Proceedings 820 of the 19th IEEE instrumentation and measurement technology 821 conference, 2002. IMTC/2002 vol. 1, IEEE*, pp 7–12