

A Effective Feature Construction Method for Fall Detection using Smartphone

Chunshan Li¹, Tianyu Dai², Xin Hu³, Dianhui Chu⁴
{lics@hit.edu.cn¹, daity@hit.edu.cn², hux@hit.edu.cn³,
chudh@hit.edu.cn⁴ }
School of Computer Science and Technology
Harbin Institute of Technology (Weihai)^{1,2,3,4}

Abstract. Recent years, smartphone based fall detection solutions have become research hotspots. These previous algorithms always analyze two types of data (accelerometer and gyroscope) and detect fall event on activities of daily life (ADL) of people which does not consider the case on physical exercise, such as, running etc. In this paper, we propose an effective feature construction method to convert a continuously device motion record to a feature vector which can define the occurrence of a fall event accurately. Base on those feature vectors, a heuristic fusion approach is adopted to extract the fall events on ADL with running. Our method runs on four types of refined and unbiased data (Attitude, RotationRate, Gravity and UserAcceleration) providing by iPhone's Core Motion framework. And 15 volunteers were employed to simulate fall events. The empirical results have demonstrated that the proposed method is effective and reliable on ADL with physical exercise.

Keywords: Fall Detection; Device Motion; Smartphone; Feature Construction

1 Introduction

Fall-related injuries among the elderly people (age 65 and older) are the cause of nearly 750,000 hospitalizations and 25,000 deaths per year in the United States [1]. More than one third of elderly fell each year [2]. It was established that the earlier the fall is detected, the lower is the rate of morbidity-mortality [3,4]. According to type of methods, the available fall detection system can be divided into computer vision-based method, audio-based method, infrared sensor-based method and wearable device-based method. [6]. With the development of smartphones' sensor, the fall detection solutions become research hotspots [5]. Lee's work [7] shows that fall detection using smartphones is a feasible and highly attractive technology for elderly, especially those who live alone. Erik et al. [13] uses an ensemble of decision trees to compute a confidence that a fall preceded on a ground event. Meanwhile, many studies proposed some fall detection system by built-in tri-accelerometer of smartphone[8,9]. However, previous studies often focused on analyze fall event on accelerometer data and gyroscope data, and it does not consider the case on physical exercise. In this paper, we can obtain Device Motion data from the Core Motion framework on iOS, which uses sensor fusion algorithms to refine the raw data and generate information for a device's Attitude, Rotation rate, the direction of Gravity and the User-generated Acceleration. The advantages of Device Motion data are as follows: firstly, researchers do not need to do the filtering method on the smartphones since the data derived from the Core Motion is unbiased; secondly, the Device

Motion provide four data types, which is more than the employed raw data in wearable sensors or smartphones by traditional researches.

The aim of our work is to design a feature construct method to convert a continuously device motion record to a feature vector, which can define the occurrence of a fall event. Then, we adopted a heuristic fusion approach to extract the fall events on ADL with running. At last, we design experiment to evaluate the reliability of the Device Motion data that detects falls with respect to traditional classification methods such as KNN, SVM, Naive Bayes and Decision Tree. Based four data types of Device Motion, we exploit different feature combination pattern, and verify which pattern can get the best performance. At last, we run fall detection algorithms on ADL with running dataset and find best feature combination pattern.

Due to the fact that there is no available Device Motion data on the internet, we invited 15 volunteers to simulate falls and recorded Device Motion data during their activities of daily life. At last, we got the validation dataset that contains 186 fall events and 1920(contain running data) ADL events. After a comparison of diverse traditional classification methods, we assessed the sensitivity, specificity and accuracy of different methods.

The remainder of the paper is organized as follows. Sec 2 introduces the motivation of our work and discusses related work. Sec 3 outlines our work. Sec 4 presents the results of our experimentation. Sect. 5 discusses the achieved results. Sect.6 provides some thought about the future directions.

2 Related Work

The number of elderly people will continuously grow in the near future. Indeed, the World Population Ageing Report (2013) states that the global share of elderly people will reach more than 21% by 2050 (more than 2 billion people) [10]. Intensive related researches have been and focused on the wearable device especially based on the smartphones, since in many parts of the world, almost everyone has a smartphone. Hsu et al. [14] construct a set of six features and use support vector machine to detect fall on smartphone. Literature [11] shows that almost 19 previous works employed features from accelerometer or gyroscope data to detect falls using threshold method, SVM or other methods. Literature [15] incorporates a threshold based algorithm and implements the method on Andorid system. Luque et al. [12] investigated more than 50 fall detection systems based on Android. Most of those systems employed built-in accelerometer or gyroscope sensor, and seldom of those considered orientation sensor to help the system detect falls better. What's more, all of these researches mentioned above have not compared and analysed the impact from different data type combination pattern. Therefore, this article compares the result when exploit different data combination pattern to detect falls using the several classification methods.

3 Methods

3.1 Data Collection on Smartphon

A good fall detection methods rely on high quality data acquired by sensing devices. In this paper, we focus on Device Motion data derived from iPhone particularly, which contains four data types, and each type includes 3 components. They are Attitude.pitch, Attitude.roll,

Attitude.yaw; Rotationrate.x, Rotationrate.y, Rotationrate.z, Gravity.x, Gravity.y, Gravity.z; Useracceleration.x, Useracceleration.y, Useracceleration.z. Among those components, the pitch, roll and yaw refer to the Euler angles. Since these data filtering and deviation correction process is completed by the iPhone's Core Motion framework, we can focus on the analysis of these data.

In our work, we set the data sampling interval to 0.015ms, and marked with only 2 activity types, ADL and fall. The program will obtain Device Motion data and save them to the application's file system with the text format continuously as the data collection program is running. When the volunteers end the collection program, they can submit the data to us by email. Fig. 1 shows the data collection flow chart.

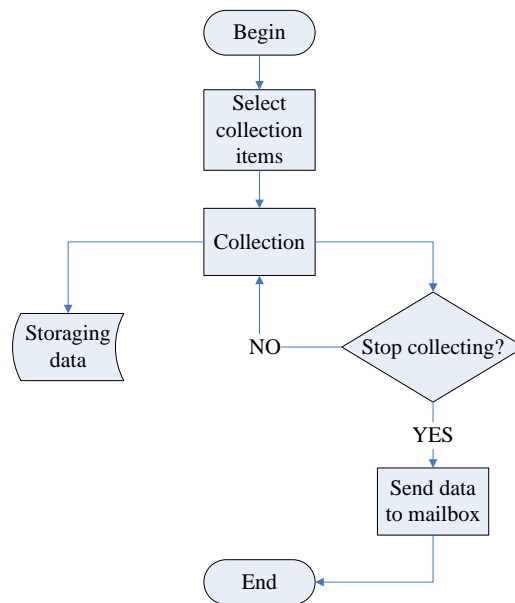


Figure 1. Data collection flow scheme

3.2 Data set construction

In order to support the validation of the proposed questions, Device Motion readings were recorded during ADL and simulated falls. Since inviting elderly people to simulated falls is unsafe, we invited 15 healthy young volunteers to simulate falls as well as recorded their activities of daily life. All the tests were carried on using one iPhone4s in the trousers left front pocket and another in the right front pocket in approximately vertical position.

The volunteers were composed of 15 males, with average height of 175cm and average weight of 65kg. Considering the cushioning mattress is not very soft and the volunteers' safety, they performed only 3 different fall types and each fall type was repeated up to 3 times. The fall types include lateral fall to the right ending lying flat, lateral fall to the left ending lying flat and forward fall ending lying flat. The collection of the ADL samples was chosen randomly from the ADL derived from the 15 volunteers. We chose 6 time periods, a total of about 960 thousand records as ADL samples, and then utilized Matlab to extract continuously Device Motion records randomly from these 6 subsets according to their proportion. It is

noteworthy that one of the sample sets was collected during the running activity from one volunteer.

3.3 Description of Fall Event

Fig. 2 shows an example record during fall simulating. We can clearly see the behaviour of the volunteer from the gravity.y's changes with the timestamp in it.

In Fig. 2, the area between two red dotted lines represented the volunteer was falling, the green area represented the volunteer was standing, the purple area represented the volunteer was recovering from the falling, the left blue area represented the volunteer was putting the smartphone in his pocket and the right blue area represented the volunteer was taking out the smartphone from his pocket. We selected the area in the two red dotted line as a sample of a fall event, as marked with ①–⑨.

We found that the number of gravity.y records between two red dotted lines in Fig.3 is approximately equals 60 after 186 fall events' mark work, thus we selected 60 consecutive Device Motion records to represent a fall event. In order to maintain consistency, we selected 60 consecutive Device Motion records to represent an ADL event too.

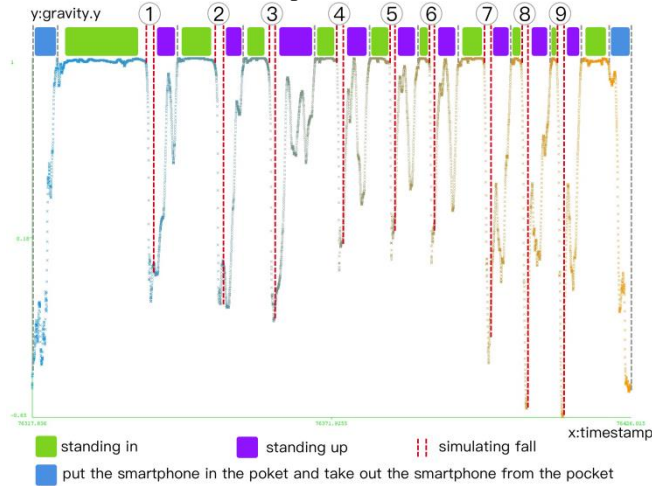


Figure 2. Example record during fall simulating

3.4 Fall Feature Construction

The single activity was represented by a continuously 60 Device Motion records when we first extract it. We adopt a feature extraction method to convert 60 consecutive records into a feature vector to facilitate the comparison of the traditional classification algorithms. As mentioned above, we chose 60 consecutive Device Motion records as a sample activity. Therefore, an activity sample set is a two-dimensional matrix which is represented as $Record_{n,m}$ ($n = 60, m = 12$), and the new feature vector is recorded as $Feature_{1,m} = (Feature_{1,1}, Feature_{1,2}, \dots, Feature_{1,j}, \dots, Feature_{1,m})$ ($1 \leq j \leq 12, m = 12$). $Feature_{1,j}$ can be calculated by the formula (1).

$$Feature_{1,j} = \sqrt{\frac{\sum_{i=1}^{n-1} |R_{i+1,j} - R_{i,j}|^2}{n-1}} \quad (1 \leq i \leq n) \quad (1)$$

After the activity's feature extraction process, we obtained 1920 ADL events (contain 560 running events) and 186 fall events.

4 Result

4.1 comparison classifiers and Evaluation Metric

In this section, we have implemented four traditional classification methods as fall detector, which are k-Nearest Neighbour (KNN) classifier, the SVM classifier, the Naive Bayes classifier and the Decision Tree classifier. These classifiers have been trained and tested using 10-fold cross-validation method based on both ADL and Fall instances.

Considering the output of the traditional classification method and the dataset context, there are 4 possible situations:

True Positive (TP): a fall occurs and the classifier properly detects it;

False Positive (FP): the classifier detects fall but actually it does not occur;

True Negative (TN): an ADL occurs and the classifier detects it;

False Negative (FN): a classifier detects ADL but actually it does not occur.

We use sensitivity, specificity and accuracy to evaluate the performance of the traditional classification algorithms. They are calculated as formulas (2), (3), (4).

$$Sensitivity = \frac{TP}{TP+FN} \quad (2)$$

Sensitivity represents the precision rate of the fall event's classification results.

$$Specificity = \frac{TN}{TN+FP} \quad (3)$$

Specificity represents the precision rate of the ADL event's classification results.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

Accuracy represents the precision rate of true results among all classification results.

4.2 Dectection Experiments on ADL with Physical Exercise

The total number of combination pattern equals 15 ($C_4^1 + C_4^2 + C_4^3 + C_4^4$) according to 4 types of data. The results obtained from the execution of four traditional classification algorithms using the previously transformed feature vectors which contains 1920 ADL events (contain 560 running events) and 186 fall events are presented in Table 1 for both left pocket and right pocket samples.

Regarding the specific number of different combination pattern, we divided the 15 combination pattern into four categories. The first category chose four data types, consists only one combination pattern; the second category chose three data types, consists of four combination pattern; the third category chose two data types, consists of six combination pattern, the fourth category chose only one data type, consists of four combination pattern. Because the first category only contains one combination pattern, it has no comparability under the different combination of four data types. We found that the Naive Bayes classifier achieved best sensitivity with the combination of Attitude, RotationRate, Gravity and UserAcceleration. The SVM classifier presented the best specificity with the combination of Attitude and Gravity, and the KNN classifiers presented the best accuracy with the combination of RotationRate, Gravity and UserAcceleration.

Table 1. Results of the Detection Algorithms with 560 Running Events)

No	Combination Pattern	Measures	Naive Bayes	SVM	KNN	J48
1	Attitude RotationRate Gravity UserAcce	Sensitivity	0.973	0.613	0.828	0.871
		Specificity	0.859	0.992	0.996	0.985
		Accuracy	0.869	0.959	0.981	0.975
2	RotationRate Gravity UserAcce	Sensitivity	0.968	0.796	0.839	0.855
		Specificity	0.871	0.992	0.996	0.989
		Accuracy	0.879	0.975	0.982	0.977
3	Attitude Gravity UserAcce	Sensitivity	0.984	0.586	0.833	0.855
		Specificity	0.865	0.995	0.989	0.986
		Accuracy	0.875	0.959	0.975	0.974
4	Attitude RotationRate UserAcce	Sensitivity	0.941	0.505	0.747	0.844
		Specificity	0.851	0.993	0.995	0.986
		Accuracy	0.858	0.95	0.973	0.974
5	Attitude RotationRate Gravity	Sensitivity	0.962	0.532	0.801	0.844
		Specificity	0.868	0.993	0.99	0.984
		Accuracy	0.877	0.953	0.973	0.972
6	Attitude RotationRate	Sensitivity	0.828	0.484	0.785	0.769
		Specificity	0.855	0.993	0.989	0.983
		Accuracy	0.853	0.948	0.971	0.964
7	Attitude Gravity	Sensitivity	0.957	0.199	0.72	0.78
		Specificity	0.895	0.999	0.973	0.982
		Accuracy	0.9	0.928	0.951	0.964
8	Attitude UserAcce	Sensitivity	0.946	0.565	0.694	0.833
		Specificity	0.855	0.995	0.983	0.984
		Accuracy	0.863	0.957	0.958	0.971
9	RotationRate Gravity	Sensitivity	0.946	0.71	0.79	0.86
		Specificity	0.88	0.99	0.991	0.983
		Accuracy	0.886	0.965	0.973	0.972
10	RotationRate UserAcce	Sensitivity	0.887	0.812	0.753	0.731
		Specificity	0.861	0.991	0.991	0.988
		Accuracy	0.864	0.975	0.97	0.965
11	Gravity UserAcce	Sensitivity	0.978	0.591	0.844	0.806
		Specificity	0.878	0.993	0.99	0.987
		Accuracy	0.887	0.958	0.977	0.971
12	Attitude	Sensitivity	0.694	0.188	0.645	0.667
		Specificity	0.927	0.997	0.973	0.983
		Accuracy	0.906	0.926	0.944	0.955
13	RotationRate	Sensitivity	0.694	0.72	0.715	0.672
		Specificity	0.876	0.989	0.98	0.978
		Accuracy	0.86	0.965	0.957	0.951
14	Gravity	Sensitivity	0.957	0.36	0.688	0.849
		Specificity	0.919	0.977	0.974	0.961
		Accuracy	0.922	0.941	0.949	0.952
15	UserAcce	Sensitivity	0.882	0.586	0.591	0.591
		Specificity	0.869	0.993	0.967	0.982
		Accuracy	0.87	0.957	0.934	0.948

However, the SVM presented the worst performance with extremely unbalanced sensitivity and specificity. Its sensitivity is intensively worse than the others. Although its specificity is better than the others, we don't choose it as detection method of fall event. Although the Naive Bayes classifier obtains best performance on sensitivity metric, which is 97.3%, it also gets worse specificity and accuracy. The KNN classifier achieves best performance on

accuracy metric (98.2%). Since the accuracy metric represents the precision rate of true results among all classification results and other two metrics of KNN are slightly less than best results, we choose the KNN methods as our fall detection method with feature construction on combination 2 for the scenario of the ADL with running.

5 Discussion and Conclusion

In this study we propose an effective feature construction method to detect the occurrence of a fall event accurately. A dataset containing 1920 ADL events (with 560 running samples) and 186 fall events was constructed by 15 volunteers. Taking advantage of this dataset, we evaluated the robustness of fall detection with four traditional classification algorithms based on different combination of Device Motion data type. Extensive experiments have demonstrated that:

- The propose feature construction method is reliable to detect falls;
- Under the combination of different Device Motion data types, the performance of these four different classifiers is complicated, and there is no absolute data combination pattern making the four classifiers all have the best performance. But after an overall comparison, we found that KNN classifier presented a best performance with the combination of Attitude, Gravity and UserAcceleration on ADL with physical exercise;
- The proposed method is simple, robustness and implemented on smartphone applications easily.

Overall, the work of this paper is to provide a reference for developing a more accurate fall detection system based on smartphones. This article can help other researchers to develop the more valuable fall detection system which has low energy and high accuracy. Furthermore, we can only focus on the algorithm and combination of the Device Motion data types which only has highest sensitivity and low computing resource, because Device Motion data is not the unique parameter to judge the happen of a fall event, we can use other parameters such as human-computer interaction to reduce the false alarm rate of fall events.

References

- [1] Crandall M, Duncan T, Mallat A, et al. Prevention of fall-related injuries in the elderly[J]. Journal of Trauma & Acute Care Surgery, 2017, 81:1.
- [2] Organization W H. WHO global report on falls prevention in older age.[J]. Who Global Report on Falls Prevention in Older Age, 2007.
- [3] Gurley RJ, N Lum, M Sande, B Lo, MH Katz “Persons found in their homes helpless or dead”. N Engl J Med.; 334(26):1710-6, 1996.
- [4] Wild, D.U.S. Nayak and B. Isaacs “How dangerous are falls in old people at home?” Br Med J (Clin Res Ed). 282(6260) : 266-8, 1981.

- [5] M. Yu, S. M. Naqvi, A. Rhuma, and I. Chambers, "Fall Detection in a Smart Room by Using a Fuzzy One Class Support Vector Machine and Imperfect Training Data," in *Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1833-1836.
- [6] Maglogiannis and C. Doukas, "Emergency Fall Incidents Detection in Assisted Living Environments Utilizing Motion, Sound and Visual Perceptual Components," *IEEE Transactions on Information Technology in Biomedicine*, , pp. 1-10, 2018.
- [7] R. YW Lee and A. I. Carlisle, "Detection of falls using accelerometers and mobile phone technology," *Age and Ageing*, pp.1-7, 2011.
- [8] Aguiar B, Rocha T, Silva J, et al. Accelerometer-based fall detection for smartphones, *IEEE International Symposium on Medical Measurements and Applications*. IEEE, 2014:1-6.
- [9] He Y, Li Y, Bao S D. Fall detection by built-in tri-accelerometer of smartphone, *IEEE International Conference on Biomedical and Health Informatics*. IEEE, 2012:184-187.
- [10] United Nations (2013) *World Population Ageing Report 2013*. <http://goo.gl/z410Ju>.
- [11] Micucci D, Mobilio M, Napoletano P, et al. Falls as anomalies? An experimental evaluation using smartphone accelerometer data, *Journal of Ambient Intelligence & Humanized Computing*, 2015:1-13.
- [12] Luque R, Casilari E, Morón M, et al. Comparison and Characterization of Android-Based Fall Detection Systems, *Sensors*, 2014, 14(10):18543.
- [13] Stone E E, Skubic M. Fall Detection in Homes of Older Adults Using the Microsoft Kinect[J]. *IEEE Journal of Biomedical & Health Informatics*, 2017, 19(1):290-301.
- [14] Hsu Y W , Chen K H , Yang J J , et al. Smartphone-based fall detection algorithm using feature extraction, *International Congress on Image & Signal Processing*, 2017.
- [15] Tsinganos P, Skodras A. A smartphone-based fall detection system for the elderly. *10th International Symposium on Image and Signal Processing and Analysis*, 2017:53-58.