A classifier based approach to real-time fall detection using low-cost wearable sensors

Nguyen Ngoc Diep, Cuong Pham, Tu Minh Phuong
Computer Science Department
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
{diepnguyenngoc, cuongpv, phuongtm}@ptit.edu.vn

Abstract— **In this paper, we present a novel fall detection method using wearable sensors that are inexpensive and easy to deploy. A new, simple, yet effective feature extraction scheme is proposed, in which features are extracted from slices or quanta of sliding windows on the sensor's continuously acceleration data stream. Extracted features are used with a support vector machine model, which is trained to classify frames of data streams into containing falls or not. The proposed method is rigorously evaluated on a dataset containing 144 falls and other activities of daily living (which produces significant noise for fall detection). Results shows that falls could be detected with 91.9% precision and 94.4% recall. The experiments also demonstrate the superior performance of the proposed methods over three other fall detection methods.**

Keywords- fall detection, feature extraction, wearable sensors, SVM

I. Introduction

Each year, one in every three adults age 65 and older falls. Falls can cause moderate to severe injuries, such as hip fractures and head injuries, and can increase the risk of early death. People age 75 and older who fall are four to five times more likely than those age 65 to 74 to be admitted to a long-term care facility for a year or longer [22]. In addition to the elderly, disable people such as people with dementia, Parkinson or poor motor control significantly contribute to falls. For example, there are as many as 400 falls per 100 dementia people [9]. Injuries caused by falls can be obstacles on the elderly's living independent at homes, and increase the risk of early death.

Previous studies also show that many elderly people experience falls and they cannot stand up, even from non-injurious falls, and as result remain on ground for even longer than an hour [22]. Furthermore, half of those elderly who have been in such situations die within 6 months, even if no direct injuries from the fall have occurred [26]. Therefore, detecting falls as early as possible is a most critical as this can reduce the time between the fall and the arrival of medical care. An automatic, real-time fall detection system to send alerts to the caregivers would be extremely needed.

Most previous works approach the fall detection problem with expensive technologies [7, 28]. This would not be suitable for poor people. For example, using sensor or smartphone costs hundreds of US dollars is not highly relevant to poor people. This is particular true in Vietnam where up to 29% of Vietnamese population is classified as poor or below (according to the UNDP standard). Therefore, the development of a fall detection using low-cost sensors and easy deployment technology would be very useful for the majority of population even low-income people. Therefore, in this paper we develop a fall detection system using low-cost sensing devices (i.e. less than $25) and easy deployment: Wii Remotes that have been used in our previous studies [9, 21]. The device is available in the market and easy to buy at the electronic/game stores. In addition to our approaches presented in [9, 21], we propose a new feature set extracted from real-time acceleration data stream and the use of support vector machines (SVM) for the classification and detection stage which significantly improve the accuracy of the fall detections. In brief, our main contributions are as follows.

First, we propose a simple yet effective new feature set based on the distribution of differences of angles and directions of data points within slices/quanta (small segments) segmented from sliding windows. When using with support vector machines (SVM) [8], these features lead to significant improvements in detection accuracy as shown by experimental results.

Second, we rigorously evaluate our proposed method on a dataset collected from 12 subjects at the Posts and Telecommunications Institute of Technology with significant noise included in the dataset.

II. Related Work

Existing approaches to fall detection belong to one of three main categories: wearable sensing [11,14,16,21,27,30], ambient sensing [2,11,32] and computer vision [12,13,23]. In wearable sensing approaches, one or more devices are worn on the parts of human body to detect posture and/or motion, while ambient sensing requires multiple sensors installed in the surrounding environment. In computer vision approaches, digital cameras installed in the environment are used to provide video sequences of motion, which are analyzed to recognize the event of a fall. It is noticed that the second and third approaches are typically more accurate but they lack flexibility as these often require the pre-settings of the environments (i.e. the camera positions and calibrations) and are case specific and dependent on different scenarios. Therefore, only small areas can be tracked. Moreover, those approaches possibly raise issues concerning confidentiality and privacy invasion. Meanwhile, for wearable sensing, the tracking area is not limited as the device is worn on the human body.

Few fall detection systems utilize the fusion of accelerometer and other sensors to achieve high accuracies of fall detection. For example, Hwang et al. [14] used a tri-axial accelerometer and gyroscope, both placed on the chest or Lai et al. [16] combined several tri-axial acceleration sensors for joint sensing of injured body parts, when an accidental fall occurs. Inertial sensors and the data logging unit are combined by Wu et al. [27] to develop a portable pre-impact fall detection system. However, such approaches are often inconvenient for the users.

Initial approaches to fall detection often use thresholds in continuous measurements. Noury et al. [18, 19], for example, designed an autonomous sensor, which can be attached under the armpit to detect the combination of the following events: the velocity exceeds a specific threshold, the sequence from a vertical posture to the lying posture, and the absence of movements after the fall (velocity is less than a threshold). They achieved a sensitivity and a specificity approximately 85%. Lindeman et al. [17] placed a 3D accelerometer in an implant behind the wearer's ear lobe and proposed 3 thresholds to trigger a fall, including: the sum-vector of acceleration in the xy-plane higher than 2g; the sum-vector of velocity of all spatial components right before the impact higher than 0.7 m/s; and the sum-vector of acceleration of all spatial components higher than 6g. The system designed by Wu et al. [27] also used a threshold for vertical velocity to detect the fall prior impact. Bourke et al. proposed fall algorithms separately based on thresholds on both signals from a tri-axial accelerometer [6] and a biaxial gyroscope [5]. Moreover, our previous works [9, 21] approached to fall detection in real-time using pre-defined thresholds by computing the similarity of the trained sequence models and the observation sequence. Although significant results are achieved, using threshold-based methods often lack of adaptability. In particular, thresholds are often calibrated on simulated falls but possibly are not suitable for real-world falls. For instance, the root sum vector is considered to be a feature for impact detection in almost algorithms, but each author used a different threshold to detect the impact [3].

In contrast, some prior works approach to fall detection using classification methods. Doukas et al. [10] proposed an SVM to detect falls from running activity. Jantaraprim et al. [15] also used SVM with a proposed feature called short time min-max to distinguish fall from ADL. Zhang et al. [31] developed a 1-class SVM classifier deployed on a cell phone. In general, these systems achieve high fall detection rates (i.e. over 95%), however, comparing to our dataset [9, 21] their datasets are significantly simpler and less (or no) noisy. In this work, we also propose a classifier method for fall detection problem, but our work is distinct from others as our dataset collected from 12 subjects and each subject perform 12 falls with various postures and other 13 daily activities in which some activities are pretty similar to falls such as standing-to-sit, sitting-to-lie, etc., and we extract a new feature set effectively for considerably improving detection rates.

Fig. 1. Wii Remote (left) and its fixed position on human body (right)

III. HARDWARE AND DATA PREPROCESSING

Our approach relies on the same hardware settings as used in our previous work [21]. For the completeness of the paper and the convenience of explaining the detection method, in this section, we briefly describe the hardware and the data acquaintance step. The reader is referred to [21] for more details.

A. Hardware

The proposed system detects falls by processing data returned from a Wii Remote device worn on the hip of human body as shown in Fig. 1. Designed as part of a popular game controller, the Wii Remote has the ability to sense acceleration along three axes through the use of an ADXL330 accelerometer. Based on Micro Electro Mechanical System (MEMS) technology, the ADXL330 is a small, thin, low power, inexpensive, complete 3-axis accelerometer with signal conditioned voltage outputs, all on a single monolithic integrated circuit. This accelerometer measures acceleration with a minimum full-scale range of ±3g. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration [1]. Wii Remote also integrate an onboard 8051 microprocessor, random access memory/read only memory, human interface device profile (HID), application, and Bluetooth protocol stack which allows sensing data to be streamed to personal computer or mobile phone via Bluetooth. This helps to reduce the complexity in installation and setup.

Fig. 2 shows two samples of continuous acceleration data streams. The first sample is a fall and the second one is a non-fall activity. The accelerations are measured in X, Y, and Z axes (relative to the accelerometer). Values for acceleration are transmitted with a sampling frequency of 100Hz (100 samples per second) to a personal computer (PC) via a Bluetooth dongle and a logging application running on the PC records the acceleration data for fall analysis.

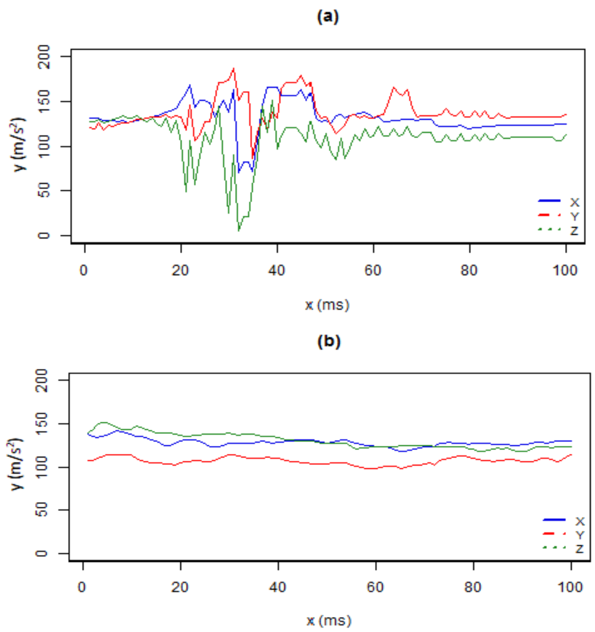

Fig. 2. A fall sample (a) and a non-fall activity sample (b)

B. Data preprocessing

Sensor data streams present complex issues related to data quality. Data are often missing, and when not missing are subject to potentially significant noise and calibration effects [4]. To alleviate the undesired of effects of missing and noisy data, we estimate nearby and historical data to fill in missing values. But before doing that gap-filled process, low-pass and high-pass filters are applied on the data to reduce noise (abnormal sample value). See [21] for details of these steps.

IV. FALL DETECTION ALGORITHM

Our proposed algorithm detects falls by classifying a window of a signal stream into "*fall*" and "*not-fall*". Given as input three streams of acceleration data along three axes X, Y, Z, the algorithm detects falls using the following steps:

- Segmentation: the input signal stream is segmented into frames or windows by using a sliding window of a fixed length.
- Feature extraction: in this step, we extract features from each frame which we use as input for the classification step.
- Classification: the system classifies each frame into "*fall*" and "*not-fall*" by using an SVM classifier with features extracted in the previous step.

The following sections give the details of each step.

A. Segmentation

The most important parameter of this step is the window size. Empirical studies with different window sizes [9, 21] have shown that the best performance of the fall detection algorithm can be achieved with sliding window size of 1.8 seconds and an overlap of 0.6 seconds. Therefore, in this study, we follow the same settings for the segmentation step.

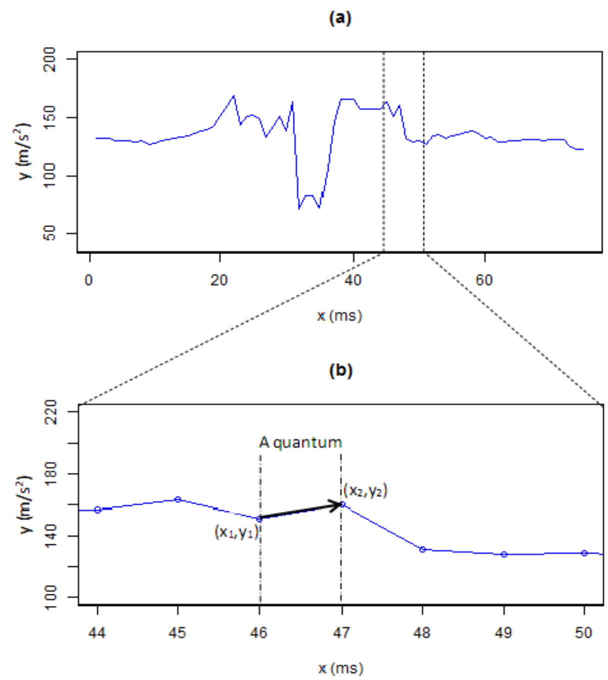

Fig. 3. Example of quantum and its direction

B. Feature extraction

Choosing features that are predictive of falls is crucial for the success of any fall detection system. Here we propose a novel type of features for the problem at hand. Briefly, for each frame (or window), we first divide the frame into small slices or quanta. Each quantum has length l, where l is a predefined parameter. Then, we compute the direction of each quantum. Finally, we count the number of quanta belonging to each bin. Here, a bin is defined to be a group of quanta with similar directions. We use the counted numbers as features and form a feature vector of size 3xM (from three streams X, Y, and Z) for the given frame, where M is number of bins and is another parameter of the algorithm. Below we describe in detail this feature extraction process.

Recall that the signal stream is a time series represented in xy-coordinates where x is in millisecond, y is in m/s^2. Fig. 3(a) shows an example frame. After the given frame is divided into slices of quanta of size l, we extract features as follows.

i) Compute the direction of each quantum. The direction $d(q)$ of a quantum q is defined as the angle between the vector connecting its start (x_1, y_1) and end (x_2, y_2) points and x-axis, as shown in Fig. 3(b). Since the tangent of this angle is $(y_2 - y_1)/(x_2 - x_1)$, direction $d(q)$ is computed as:

$$d(q) = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \tag{1}$$

The value of $d(q)$ is in range [-90^0, 90^0].

ii) Map the direction $d(q)$ into one of M bins, where a bin is a group of quanta with similar directions. In other words, $d(q)$ is quantized into M groups. There are several ways to select the best number of bins and the bins themselves, for example by means of clustering. In this paper, we select bins manually and leave the automated bin selection problem for future work. Fig. 4 shows an example of quantization setting with five bins: -90 degrees (in the negative vertical direction), -45 degrees (along the negative diagonal), 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the positive vertical direction).

iii) Count the number of quanta belonging to each of M bins and form a vector of size M, elements of which are the counted numbers. The right part of Fig. 4 shows an example of such a vector as a histogram.

We concatenate three vectors counted this way for the three signal streams (along the X, Y, Z axes) to form a vector of size 3xM, which we will use as the feature vector of the given frame.

The intuition behind using this kind of features is that frames corresponding to similar activities or falls would have similar curves of signal streams and thus have similar number of quanta with close directions. Therefore, these features provide an approximate representation of signal streams, which can be computed efficiently and is suitable for a wide range of classification algorithms.

There are two important parameters of the feature extraction process, namely the quantum length l and the bins. Using short quanta allows capturing changes of signals over finer grain time periods. However, too short quanta may make the method sensible to noisy signals. Similarly, a successful bin selection allows separating signals that are predictive of falls apart from irrelevant ones.

C. *Classification using SVM*

The features extracted in the previous step can be used with various classification algorithms to separate falls from non-fall activities. In this study, we use support vector machines (SVM) as the classifier due to its superior prediction accuracy in a number of application domains. SVMs achieve good generalization on unseen data by relying on two techniques: i) mapping input features into a new feature space often of higher dimensions using a so called kernel function; and ii) finding in the new feature space a hyperplane with max-margin that separate negative examples from positive ones. Before training and prediction with SVM, we normalize feature vectors so that all feature values are within range [0, 1].

V. EXPERIMENTS

A. *Dataset*

We used the dataset from [21] for evaluating our proposed method. The data were collected from 12 subjects, wearing Wii Remote on their hips. The subjects were asked to perform 144 falls and other 12 daily activities (non-falls). There were no constraints on the order of activities as well as time to perform the activities. Since many of non-fall activities produce signals similar to falls, for example jumping, standing-to-sit, and lots of noise, this dataset is challenging for accurate fall detection.

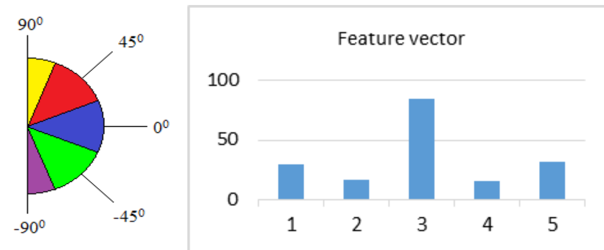

Fig. 4. An example of quantization with five bins (left) and the feature vector (right)

B. *Experimental settings*

The performance is evaluated in terms of precision and recall values, defined as:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

where TP = true positives (detected falls), FP = false positives (non-fall samples giving false fall alarm), FN = false negatives (undetected falls).

We use two experimental settings to evaluate the proposed method. In the first setting, we combine all the data collected from the 12 subjects and measure precision and recall using stratified 10-fold cross-validation. The folds are selected so that the mean response value is approximately equal in all the folds. In the case of a dichotomous classification, this means that each fold contains roughly the same proportions of the two types of class labels. We use 9 subsets for training and the remaining subset for testing. The process is repeated 10 times and the mean accuracy is taken as the result. In the second setting, the *leave-one-subject-out* protocol is used for evaluation. We use 11 subjects for training and the remaining one for testing. We average the results over 12 subjects and report average precisions and recalls. Note that the second setting is more difficult since the classifier has no information about the subject, for which it should make predictions. This setting is also more similar to practical conditions, in which a system should detect falls for new people.

We try several types of kernel functions and select the best parameters by using the popular grid search procedure. The RBF kernel and linear kernel achieve the highest accuracy. In what follows, we report the performance achieved by using linear kernel with $C = 0.1$.

C. *Results*

Effects of bin selection and quantum length

In the first experiment, we investigate the influence of bin selection (including the number of bins and the range of

TABLE I. Results for 10-fold cross-validation evaluation (different quantum lengths and bin settings)

Bin settings		Quantum length (*l*)					
		l = 1			*l = 2*		
M	***Range (degree)***	***Precision (%)***	***Recall (%)***	***F-Measure (%)***	***Precision (%)***	***Recall (%)***	***F-Measure (%)***
5	(-90,-45,-15,15,45,90)	85.7	83.3	84.5	80.6	80.6	0.806
6	(-90,-45,-15,0,15,45,90)	85.3	80.6	82.9	80.6	80.6	0.806
7	(-90,-60,-45,-15,15,45,60,90)	86.5	88.9	87.7	85.7	83.3	0.845
8	(-90,-60,-45,-15,0,15,45,60,90)	86.5	88.9	87.7	85.7	83.3	0.845
9	(-90,-60,-45,-15,15,45,60,90)	86.8	91.7	89.2	83.8	86.1	0.849
9	(-90,-65,-45,-25,-10,10,25,45,65, 90)	**91.9**	**94.4**	**93.2**	**84.2**	**88.9**	**0.865**
10	(-90,-80,-65,-45,-15, 0,15,45,65,80,90)	91.9	94.4	93.2	84.6	91.7	0.88
16	(-90,-85,-80,-70,-55,-30,-15, 0,15,30,55,70,80,85, 90)	91.7	91.7	91.7	82.1	88.9	0.853

TABLE II. Results for leave-one-subject-out evaluation (different quantum lengths and bin settings)

Bin settings		Quantum length (*l*)					
		l = 1			*l = 2*		
M	***Range (degree)***	***Precision (%)***	***Recall (%)***	***F-Measure (%)***	***Precision (%)***	***Recall (%)***	***F-Measure (%)***
5	(-90,-45,-15,15,45,90)	85.14	78.72	81.06	85.06	86.5	85.38
6	(-90,-45,-15,0,15,45,90)	82.64	80.5	80.96	85.56	90.5	87.82
7	(-90,-60,-45,-15,15,45,60,90)	85.5	82.5	83.56	84	90.5	83.56
8	(-90,-60,-45,-15,0,15,45,60,90)	85.14	80.5	82.12	85.52	92.5	88.72
9	(-90,-60,-45,-15,15,45,60,90)	86.18	88.5	87.04	85.52	92.5	88.72
9	(-90,-65,-45,-25,-10,10,25,45,65, 90)	**91.14**	**87.1**	**89**	**89.04**	**91.8**	**90.34**
10	(-90,-80,-65,-45,-15, 0,15,45,65,80,90)	85.5	80.5	86	85.52	92.5	90.16
16	(-90,-85,-80,-70,-55,-30,-15, 0,15,30,55,70,80,85, 90)	88.9	84	85.56	85.52	92.5	88.72

each bin) and quantum length on the detection accuracy. We vary the number of bins from 5 to 16 and change the range of each bin. For each bin selection setting, we computed the prediction accuracy using 10-fold cross-validation and leave-one-subject-out settings with quantum length $l = 1$ and $l = 2$. We do not use longer quanta because $l = 3$ or higher means that the sampling rate is smaller than 34 Hz, which is not fast enough to react to acceleration changes, and therefore, it is inadequate for fall detection purposes.

Tables 1 and 2 show the results for 10-fold cross-validation and leave-one-subject-out settings respectively (only combinations with the highest results are shown to save space). As expected, the method achieved more accurate results for the first setting, confirming that the second setting is more difficult.

The results show that both number of bins and bin ranges have influence on the accuracy. The optimal number of bins is around 9, and the bin sizes from 10 to 25 degrees give the best results.

While the quantum length of 1 yields the best results for 10-fold cross-validation (see Table 1) with 91.9% precision and 94.4% recall, these results are consistent to the best results (91.14% precision and 87.1% recall) in the leave-one-subject-out setting (see Table 2). A minor difference is that the recall with the quantum length of 2 (91.8%) is higher than the recall with the quantum length of 1 (87.1%) under the leave-one-subject-out setting. A possible explanation for this difference in the best value of quantum length is that shorter quanta would possibly lead to over-fitting, the effect of which is more significant one making prediction for completely unseen data, as in the second setting.

Comparing to our previous study [21], these fall detection results are significantly improved. For instance, the new method improves 7.11% precision (from 84.03% in [21]) and 4.46% recall (from 82.64% in [21]) under leave-one-subject-out training. We also compared our method with the SVM based methods proposed by Jantaraprim et al. with short time min-max features [15] and Zhang et al.'s method [30] on our dataset using 10-fold cross-validation tests (we tested their methods on our dataset [21] which consists of 144 falls and 12 other daily activities). Although [21] includes less falls than [15], by including 12 other activities, [21] is significantly noisier than [15] which only consists of fall sequences. All test results were aggregated to precision and recall which show that our proposed method considerably outperform both [15] (i.e. 94.4% compared to 88.9% recall) and [30] (i.e. 94.4% compared to 83.3% recall). As whose methods are based SVM, the test results have demonstrated that our proposed feature extraction scheme is more effective than [15] and [30].

VI. CONCLUSION

We have presented a novel method for detecting falls from low-cost wearable sensor data. By combining a new type of features, which can be computed very efficiently, with a discriminative kernel based classification algorithm (SVM), the method achieves over 90% precision at high recall in both cross-validation and leave-on-subject-out settings on a noisy dataset. Experiments also show that the method outperforms three other methods in terms of detection accuracy. Together with inexpensive hardware, our method has potential to be a good complement for existing approaches to the fall detection problem. The proposed method can be improved in several ways, for example by adding automated selection of bin number and ranges, which we plan to do in future work.

REFERENCES

[1] ADXL330: http://www.analog.com/static/imported-files/data_sheets/ADXL330.pdf (accessed on 5/25/2013)

[2] M. Alwan, P.J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, R. Felder, “A Smart and Passive Floor-Vibration Based Fall Detector for Elderly,” IEEE International Conference on Information & Communication Technologies (ICITA) (2006) 1003–1007. pp.

[3] F. Bagalà et al., "Evaluation of accelerometer-based fall detection algorithms on real-world falls," PloS one 7.5 (2012): e37062.

[4] M. Balazinska et al., "Data management in the worldwide sensor web," Pervasive Computing, IEEE 6, no. 2 (2007): 30-40.

[5] A.K. Bourke and G.M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor,” Medical Engineering and Physics 30, no. 1 (2008): 84.

[6] A.K. Bourke, J. V. O'brien and G. M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," Gait & posture 26, no. 2 (2007): 194-199.

[7] Brickhouse Alert: http:/www.brickhousealert.comlpersonal-emergency-medical-alarm.html (accessed on 5/25/2013)

[8] N. Christianini and J. Shawe-Taylor, “An introduction to support vector machines,” Cambridge University Press, 2000.

[9] K.Q. Dinh, C. Pham, N.D. Nguyen, M.P. Tu, “Real-Time Fall Detection Using an Accelerometer”, Journal of Science and Technology 51(1A) 2013. 108-116 (in Vietnamese)

[10] C. Doukas et al., "Patient fall detection using support vector machines," Artificial Intelligence and Innovations 2007: from Theory to Applications. Springer US, 2007. 147-156.

[11] C. Doukas, I. Maglogiannis, F. Tragkas, D. Liapis and G. Yovanof, “Patient Fall Detection Using Support Vector Machines,” International Federation for Information Processing (IFIP), SpringerLink 247 (2007) 147–156.

[12] Z. Fu, E. Culurciello, P. Lichtsteiner and T. Delbruck, “Fall detection using an address-event temporal contrast vision sensor,” IEEE Trans. Biomed. Circuits Syst. 2 (2008) 88–95.

[13] L. Hazelhoff, J. Han, P.H.N. de With, “Video-based fall detection in the home using principal component analysis,” Adv. concepts Intell. Vision Syst. (ACIVS), SpringerLink 5259 (2008) 298–309.

[14] J.Y. Hwang, J.M. Kang, Y.W. Jang, H.C. Kim, “Development of Novel algorithm and real-time monitoring ambulatory system using Bluetooth module for fall detection in the elderly,” in Proceedings of the 26th annual international conference of the IEEE-EMBS; 2004

[15] P. Jantaraprim, P. Pornchai, L. Chusak and W. Booncharoen, "Fall Detection for the elderly using a Support Vector Machine," International Journal of Soft Computing and Engineering 2, no. 1 (2012): 484-490.

[16] C.F. Lai, S.Y. Chang, H.C. Chao, Y.M. Huang, “Detection of cognitive injured body region using multiple tri-axial accelerometers for elderly falling,” IEEE Sens. J. 11 (2011) 763–770.

[17] U. Lindemann, A. Hock, M. Stuber, W. Keck, and C. Becker, "Evaluation of a fall detector based on accelerometers: a pilot study," Medical and Biological Engineering and Computing 43, no. 5 (2005): 548-551.

[18] N. Noury et al., “Monitoring behavior in home using a smart fall sensor and position sensors,” in Microtechnologies in Medicine and Biology, 1st Annual International, Conference On. 2000, pp. 607-610. IEEE, 2000.

[19] N. Noury et al., “A smart sensor for fall detection in daily routine,” in SICICA2003, Aveiro-Portugal, 9-11 Jul 2003.

[20] C. Pham and P. Olivier, "Slice&dice: Recognizing food preparation activities using embedded accelerometers," Ambient Intelligence. Springer Berlin Heidelberg, 2009. 34-43.

[21] C. Pham, T.M. Phuong, “Real-Time Fall Detection and Activity Recognition Using Low-Cost Wearable Sensors,”, in Proc. of 13th ICCSA 2013, Part I, LNCS 7971, pp. 673--682. Springer, Heidelberg (2013).

[22] A.C. Reece, J.M. Simpson, “Preparing older people to cope after a fall,” Physiotherapy 1996;82(4):227–35.

[23] C. Rougier, J. Meunier, A. St-Arnaud, J. Rousseau, “Robust video surveillance for fall detection based on human shape deformation,” IEEE Trans. Circuits Syst. Video Technol.(CSVT) 21 (2011) 611–622.

[24] J.A. Stevens, and A.M. Dellinger, "Motor vehicle and fall related deaths among older Americans 1990–98: sex, race, and ethnic disparities," Injury Prevention 8.4 (2002): 272-275.

[25] Wii Remote: http://en.wikipedia.org/wiki/Wii_Remote (accessed on 5/25/2013)

[26] D. Wild, U.S. Nayak and B. Isaacs, "How dangerous are falls in old people at home?" British medical journal (Clinical research ed.) 282, no. 6260 (1981): 266.

[27] G. Wu, S. Xue, “Portable preimpact fall detector with inertial sensors,” IEEE Trans. Neural Syst. Rehabil. Eng. 16 (2008) 178–183.

[28] W. Wu et al., "The smartcane system: an assistive device for geriatrics," in Proceedings of the ICST 3rd international conference on Body area networks, p. 2., 2008.

[29] Y. Xinguo, “Approaches and principles of fall detection for elderly and patient,” in Proc. 10th IEEE Int. Conf. e-Health Netw., Appl. Serv., 2008, pp. 42–47.

[30] T. Zhang, J. Wang, L. Xu, P. Liu, "Fall detection by wearable sensor and one-class SVM algorithm," Intelligent Computing in Signal Processing and Pattern Recognition. Springer Berlin Heidelberg, 2006. 858-863.

[31] T. Zhang, J. Wang, L. Xu, P. Liu, “Using Wearable Sensor and NMF Algorithm to Realize Ambulatory Fall Detection,” Springer-Verlag, 2006 Vol. 4222, pp. 488–491.

[32] Y. Zige, D. Litvak and I. Gannot , “A method for automatic fall detection of elderly people using floor vibrations and sound proof of concept on human mimicking doll falls,” IEEE Trans. Biomed. Eng. 56 (2009) 2858–2867.