Embedded System for Fall Detection Using Body-worn Accelerometer and Depth Sensor

Michal Kepski[1], Bogdan Kwolek[2]
[1] Interdisciplinary Centre for Computational Modelling, University of Rzeszow, 35-959 Rzeszow, Poland, mkepski@ur.edu.pl
[2] Department of Computer Science, AGH University of Science and Technology, 30-059 Krakow, Poland, http://home.agh.edu.pl/~bkw/contact.html

Abstract— **This paper presents an embedded system for fall detection using accelerometric data and depth maps. A real-time processing of motion data and depth maps is realized on a low-cost PandaBoard platform. In order to achieve detection of human falls with low computational cost the system performs a depth-based inferring about the fall event when person's movement is above some preset threshold. The performance of the system has been evaluated on our publicly available dataset consisting of synchronized depth maps and motion data. To investigate the detection accuracy in depth maps from different camera views the image sequences were simultaneously recorded by two Kinect sensors, where one of them was placed in the front of the scene, whereas the second one was located on the ceiling. The motion data were acquired by a body-worn accelerometer and transmitted wirelessly to the processing unit, responsible for both synchronization and recording or processing of the data.**

Keywords—Embedded Systems, Assistive Technologies, Fall Detection.

I. INTRODUCTION

Falls are a well-known cause of morbidity from injury and mortality in the elderly. They are the leading reason of injury-related hospitalisation in persons aged 65 years and over and account for significant fraction of all hospital admissions in this age-group [1]. Even falls that do not lead to physical injuries can result in the so called post-fall syndrome, which typically manifests itself in a loss of confidence, wobble, tentativeness with resultant loss of mobility and independence. The reason for this is that the elderly fear of lying after the fall on the floor in solitude and without help for a long time [2]. Therefore, falls should be detected as early as possible. In consequence, the development of low cost and reliable fall detection system has received considerable attention in recent years [3]. Thanks to automatic fall detection, the system can issue an alert without needing to press the emergency button. As a result, the injured person can be delivered to a hospital in order to receive timely medical care.

Fall detection methods can be divided into two major groups depending on how the information is acquired, that is, methods using vision sensors and methods based on non-visual sensors. The main limitation of systems based on typical RGB cameras [4] is that they cannot achieve satisfactory fall detection accuracy in poor illumination conditions. Besides the privacy issues, the lack of depth information may lead to poor fall detection performance. Moreover, typically such systems have considerable computational demands. In the second groups of the fall detectors the inertial sensors are used most frequently. Usually, they use a body-worn accelerometer and a threshold based algorithm to examine if a person's movement is above some preset threshold [5]. However, as demonstrated in [6], such systems generate a large number of false alarms, which in turn lead to frustration of the seniors. Recently, Kinect's depth camera has been proposed to be utilized in fall detection systems [7, 8]. In the discussed work it has also been demonstrated that the depth maps are sufficient to detect the person being monitored. Since the Kinect uses infrared light sensors to illuminate the viewed scene and an infrared camera to observe them in invisible light, the fall detection can be done any time. A recent survey on the use of Kinect in fall detection systems can be found in [9]. Although these solutions are promising, they still have insufficient accuracy of fall detection as well as generate too large number of false alarms [10].

In this work we present an embedded system for fall detection on the basis of accelerometric data and depth maps. We show how motion data and depth maps are processed in real-time on a low-cost PandaBoard platform to achieve reliable fall detection. To attain the fall detection with low computational cost the system performs depth map-based inferring about the fall event only when person's movement is above some preset threshold. The detection accuracy has been evaluated on our publicly available dataset consisting of synchronized depth maps and motion data. In order to investigate the detection accuracy in depth maps from different camera views the image sequences were simultaneously recorded by two Kinect sensors, where one of them was placed in the front of the scene, whereas the second one was mounted on the ceiling. The motion data were acquired by a body-worn accelerometer and transmitted wirelessly to the processing unit, responsible for both synchronization and recording of the data.

II. PERSON DETECTION IN DEPTH MAPS

In this Section we discuss algorithms for person detection in depth map sequences. In the below Subsection we explain how person is detected in depth images acquired by a Kinect facing the scene, whereas in subsequent Subsection we describe a method for person detection in depth maps acquired by a Kinect mounted on the ceiling, i.e. providing the top view of the scene.

A. *Person Detection in Frontal Depth Maps*

The frontal maps were acquired by a static Kinect that was placed at the height of 1 m from the floor. The person has been detected through differencing the current depth image from a depth reference image. The depth reference image represents the scene depth and it is accommodated on-line to reflect the scene changes. Each pixel in the depth reference map is a temporal median of the fifteen depth pixels. For each depth pixel a fifteen element circular buffer is utilized to store continuously the acquired depth values. Every fifteen depth map acquired by the Kinect is stored in the depth circular buffers. In practical terms this means that for Kinect sensor acquiring the images at 30 Hz, the depth reference image is entirely refreshed in 7.5 seconds. The person can be delineated with 30 fps through differencing the current depth image from the depth reference image accommodated in such a way. Figure 1. demonstrates delineation of the person in an example depth image.

Fig. 1. Person extraction in frontal depth maps. Depth reference image (left), current image (middle), image with the extracted person (right).

B. *Person Detection in Overhead Depth Maps*

The observation area for an overhead Kinect mounted on the height of 2.6 m is about 5.5 m^2. In order to increase the field of observation we utilized a homemade pan-tilt head to rotate the Kinect sensor. Thanks to the use of such a pan-tilt head the field of the covered view is far larger and in effect the Kinect can observe a typical room. During the person movement the controller rotates the camera in order to keep his/her in the central part of the image. The person is detected in real-time on the basis of depth region growing [11]. The person's position is expressed as the centroid of the delineated area. The algorithm detects the floor with low computational cost in order to decrease the number of pixels that can be potentially included into the person blob. Figure 2 demonstrates the extracted person blob by the discussed algorithm together with the corresponding depth map.

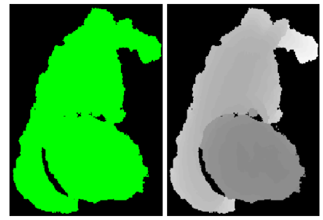

Fig. 2. Person extraction in overhead depth maps, the extracted blob (left) and the corresponding depth map (right).

III. EMBEDDED SYSTEM FOR FALL DETECTION

At the beginning of this Section we discuss two modes of work of the system. Afterwards, we outline the steering of the pan-tilt head. Finally, we shortly overview the PandaBoard.

A. Modes of Operation of the System

The system detects falls on the basis of motion data from a body-worn accelerometer and features, which are extracted on the basis of depth map sequences. There are two modes of work of the system. In the first one the system utilizes acceleration data to signal a potential fall event. Such a fall hypothesis is then validated on the basis of features extracted from depth maps. The final decision about the fall is taken on the basis of features describing both lying pose and features reflecting body movements in map sequences. In order to reduce the computational costs the person is not detected frame-by-frame but instead a circular buffer is utilized to hold a collection of the preceding depth maps. In case of the potential fall, the stored frames are utilized to detect a person and then to calculate both static and dynamic features. Thanks to such an approach the fall can be detected reliably with low computational cost. In the second operation mode the system detects the person in each frame to extract his/her centroid, which is required by the controller of the active head to keep the target in the central part of the current depth map. The decision about the fall can be undertaken on the basis the depth map only or using both accelerometric data and depth maps.

The accelerometric data are acquired by x-IMU device and then transmitted wirelessly to the PandaBoard, which executes a selected fall detection algorithm. The Kinect Xbox sensor is connected to the board via USB. The microcontroller of the active head is connected with the PandaBoard through I2C bus.

B. Pan-Tilt Head

The homemade active head consist of a microcontroller (MCU) and two servomechanisms to rotate the camera in two axes, see Fig. 3. The microcontroller board is based on the 8-bit ATmega328 chip with 16 MHz clock and 2 KB RAM. It is equipped with 6 analog inputs, 14 digital I/O pins, where six of them can be used to perform pulse width modulation (PWM). The utilized MCU has a number of facilities for communication with other devices: UART TTL serial, I2C or SPI. To obtain smooth camera rotations, two PID controllers (one for each degree-of-freedom) are employed. After the actuator outputs are calculated, the motor servos are controlled using PWM.

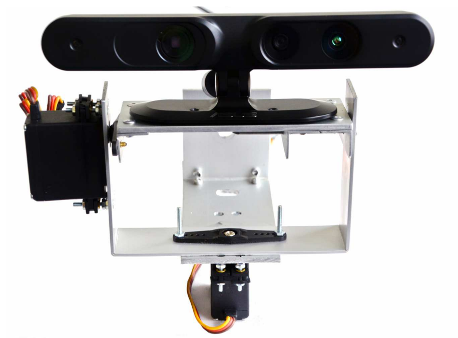

Fig. 3. Depth sensor (Asus Xtion PRO) and our pan-tilt unit.

C. PandaBoard

PandaBoard is a low cost, mobile software development platform based on the Texas Instruments OMAP4430 system on a chip (SoC). It is driven by the dual-core ARM Cortex-A9 OMAP4430, with each core running at 1 GHz, a 304 MHz PowerVR SGX540 integrated 3D graphics accelerator, a programmable C64x DSP, and 1 GB of DDR2 SDRAM. Our experimental evaluation of the processing performance shows that the Dhrystone 2 score is equal to 4214871 [lps], the Double-Precision Whetstone is equal to 836 [MWIPS], whereas the number of iterations/sec in CoreMark benchmark is equal to 2858. The board also contains wired 10/100 Ethernet along with wireless Ethernet and Bluetooth connectivity. The PandaBoard ES can support various Linux-based operating systems such as Android and Linux Ubuntu. A block diagram of the board is shown on Fig. 4.

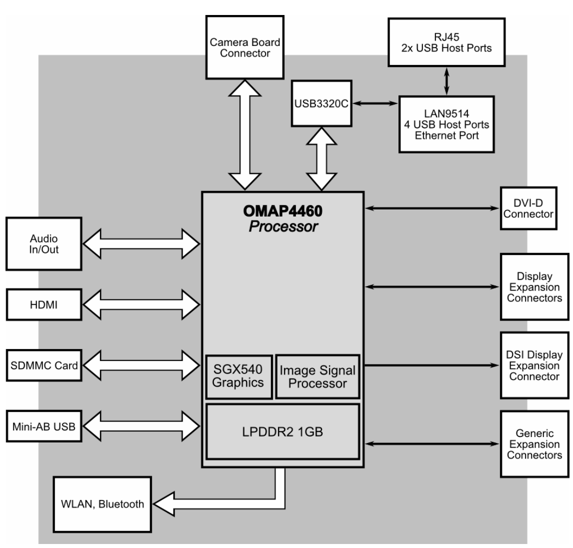

Fig. 4. Block diagram of the PandaBoard ES.

IV. REAL-TIME DATA ACQUISITION AND PROCESSING

The human fall detection system runs under Linux operating system. The fall detection application executes five main concurrent processes that communicate via message queues, see Fig. 5. The message queues provide asynchronous communication between processes. The messages placed onto the queue are stored until the receiver retrieves them. This means that the sender and the recipient of the message do not need to interact with the queue at the same time. The first process is accountable for acquiring motion data from the wearable device, the second one acquires depth maps from the depth sensor, third process continuously updates the reference depth map, fourth one is responsible for data processing and feature extraction, whereas the fifth process is accountable for data classification and triggering the fall alarm. The dual-core processor of the utilized PandaBoard allows parallel execution of acquisition and processing processes.

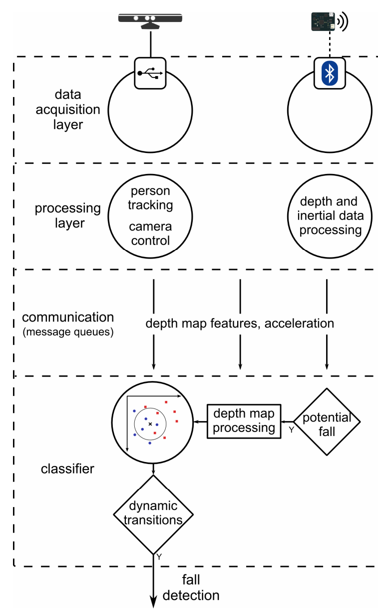

Fig. 5. Data acquisition, processing and communication between the main processes.

The following features are extracted from the frontal depth maps to recognize the lying pose:

- H/W - a ratio of height to width of the person's bounding box in the depth maps
- H/H_{max} - a proportion expressing the height of the person's surrounding box in the current frame to the physical height of the person, projected onto the depth map
- D - the distance of the person's centroid to the floor
- $\max(\sigma_x, \sigma_z)$ - standard deviation from the centroid for the abscissa and the applicate, respectively.

In addition to the above features the algorithm calculates also the ratio $H(t)/H(t-\Delta T)$, where t denotes the time in which the impact took place, and ΔT is equal to 600 ms. Owing to the use of the body-worn accelerometer to sense the motion of the person undergoing monitoring, time moment of the impact, i.e. time t, can be determined precisely and with low computational cost. The discussed features were utilized by a classifier responsible for fall detection on the basis of the frontal depth maps.

The detection of the fall in the overhead depth maps is done on the basis of the following features:

- $H/H_{\max}$ - a ratio of head-floor distance to the height of the person
- A - a ratio expressing the person's area in the image to the area at assumed distance to the camera
- l/w - a ratio of major length to major width of a blob representing the person on the depth image.

The ratio $H(t)/H(t-\Delta T)$, where $H(t)$ denotes the distance between the head and the floor is calculated as well to express the speed of the person movement in the depth maps [11].

Figure 5. depicts the UML diagram of data processing for the Kinect mounted at the ceiling. The diagram for the system configured for processing the frontal depth maps does not have the block responsible for camera control.

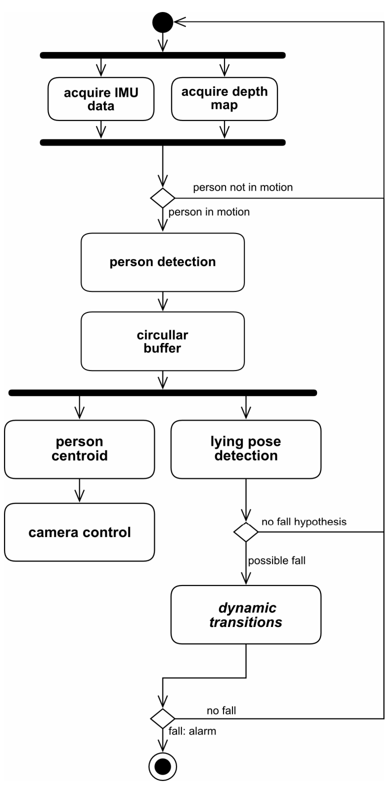

Fig 5. Data processing (UML diagram)

V. Fall Detection Dataset

The UR Fall Detection (URFD) dataset consists of depth map sequences acquired by Kinect sensors with the corresponding motion data, which were acquired by a body-worn accelerometer. The sensing unit was worn near the spine on the lower back. The motion data contains the acceleration over time in the x, y, and z axes together with the precalculated $SV_{Total}(t)$. They were calculated in the following manner:

$$SV_{Total}(t) = \sqrt{A_x(t) + A_y(t) + A_z(t)} \tag{1}$$

where $A_x(t)$, $A_y(t)$, $A_z(t)$ stand for the acceleration in reference to the local x, y, and z axes at time t, respectively. The frontal depth maps with the corresponding RGB images were acquired by a static Kinect that was placed at the height of 1 m from the floor, whereas the top view RGB-D maps were acquired by a second Kinect, which has been mounted at a ceiling at the height of 3 m. Figure 6. depicts sample RGB and depth images from the discussed dataset. In the top row are RGB and depth images acquired by the frontal Kinect, whereas in the second row are RGB and depth images acquired the overhead sensor. The plot depicts the SV_{Total} values vs. time, i.e. frame number.

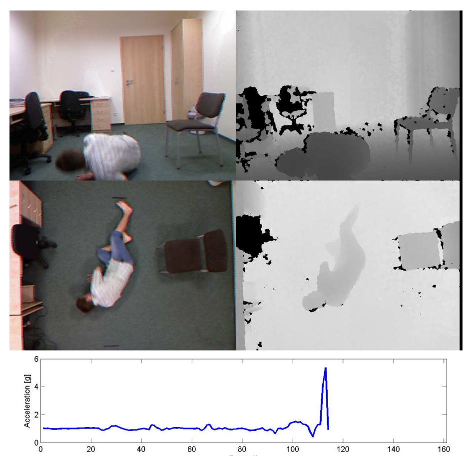

Fig 6. Sample images from the UR Fall Detection dataset with corresponding plot of the acceleration vs. time.

The dataset consists of thirty image sequences with falls, thirty image sequences with typical ADLs like crouching down, picking-up an object from the floor, sitting down, and ten sequences with fall-like activities as fast lying on the floor and lying on the bed/couch. Two kinds of falls were performed by five persons: from standing position and from sitting on the chair. All RGB and depth images are synchronized with the motion data. They were recorded at 30 Hz frame rate. The dataset is available for download via the following link: http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html.

VI. EVALUATION OF THE SYSTEM

The fall detection system has been evaluated on the URFD dataset. Table 1. shows the performance of the system that has been achieved on frontal URFD data sequences. As we can notice, slightly better results were obtained by the k-NN classifier (with three neighbors) in comparison to the linear SVM. The SVM classifier has been trained on a PC using LIBSVM software [12].

Table 1. Performance of fall detection on frontal URFD data sequences [%].

	k-NN	SVM
Accuracy	95.71	94.28
Precision	90.90	88.24
Sensitivity	100.00	100.00
Specificity	92.50	90.00

Table 2. presents the performance of fall detection on overhead data sequences from the UR Fall Detection dataset. The discussed results were obtained by a linear SVM. As we can notice, the results are better in comparison to results obtained on the frontal sequences.

Table 2. Performance of fall detection on overhead URFD data sequences [%].

	Accuracy	Precision	Sensitivity	Specificity
SVM	99.45	98.21	100.0	99.22

Table 3. presents times needed for update of the depth reference images and person extraction using region growing. The discussed processing times were obtained on PandaBoard ES and a personal computer equipped with Intel i7-3610QM 2.3 GHz 8 GB RAM. Having on regard that the depth reference image is updated every 15th frame acquired by the depth sensor, the whole depth map to be updated can be divided into blocks and each of them can be accommodated in time shorter than 15~ms. The subtraction of the current depth image from the depth reference image can be realized in about 7~ms. The region growing time is average time that was obtained on the sequence available on: http://fenix.univ.rzeszow.pl/~mkepski/demo/act.mp4. The board was powered by Linaro 12.11 operating system, whereas the code C++ code was compiled using GCC~4.6.3.

Table 3. Processing times [ms].

	PandaBoard ES	Intel i7
Depth reference image update	182.86	24.61
Region growing	16.70	3.80

Ten volunteers with age over 26 years attended in an evaluation of the developed algorithm and the embedded system for fall detection in real-time. Intentional falls were performed in an office by six persons towards a carpet with thickness of about 2 cm. Each individual performed three types of falls, namely forward, backward and lateral at least three times. Each individual performed also ADLs like walking, sitting, crouching down, leaning down/picking up objects from the floor as well as lying on the floor. The acceleration threshold has been set to 2.6~g to filter the fall events from the ADls. All intentional falls have been detected appropriately.

VII. CONCLUSIONS

Most of the image-based systems require time for installation, camera calibration and they are not cheap since a considerable computational power is needed to execute in real-time the time consuming algorithms. Moreover, the false alarm of systems known from the literature is unacceptable for practical applications. In this work we have presented a low-cost embedded system for fall detection. The system has been evaluated on publicly available dataset. The presented system permits reliable and unobtrusive fall detection as well as preserves privacy of the user. We reduced the number of the false alarms through combining the features extracted from motion data and depth maps.

ACKNOWLEDGMENT

This work was supported by University of Rzeszow as well as by NCN under a research grant 2014/15/B/ST6/02808.

REFERENCES

[1] S. Heinrich, K. Rapp, U. Rissmann, C. Becker, and H.-H. König, “Cost of falls in old age: a systematic review,” *Osteoporos. Int.*, vol. 21, no. 6, pp. 891–902, Jun. 2010.

[2] M. E. Tinetti, “Predictors and Prognosis of Inability to Get Up After Falls Among Elderly Persons,” *JAMA J. Am. Med. Assoc.*, vol. 269, no. 1, p. 65, Jan. 1993.

[3] R. Igual, C. Medrano, and I. Plaza, “Challenges, issues and trends in fall detection systems,” *Biomed. Eng. Online*, vol. 12, p. 66, 2013.

[4] J. Willems, G. Debard, B. Bonroy, B. Vanrumste, T. Goedeme, “How to detect human fall in video? An overview,” *Int. Conf. on Positioning and Context-Awareness*, Antwerp, Belgium, May 2009.

[5] A. K. Bourke, J. V O'Brien, and G. M. Lyons, “Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm,” *Gait & Posture*, vol. 26, no. 2, pp. 194–9, 2007.

[6] F. Bagalà, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, and J. Klenk, “Evaluation of accelerometer-based fall detection algorithms on real-world falls,” *PLoS One*, vol. 7, no. 5, p. e37062, Jan. 2012.

[7] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier, “Fall detection from depth map video sequences,” ser. LNCS. Springer Berlin Heidelberg, vol. 6719, pp. 121-128, 2011.

[8] M. Kepski and B. Kwolek, “Fall detection on embedded platform using Kinect and wireless accelerometer,” *in Proc. of the 13th Int. Conf. on Computers Helping People with Special Needs*, LNCS, Springer, pp. II:407-414, 2012.

[9] D. Webster and O. Celik, “Systematic review of Kinect applications in elderly care and stroke rehabilitation,” *J. of NeuroEngineering and Rehabilitation*, vol. 11, 2014.

[10] E. E. Stone and M. Skubic, “Fall Detection in Homes of Older Adults Using the Microsoft Kinect,” *IEEE J. Biomed. Heath. Informatics*, Mar. 2014.

[11] M. Kepski and B. Kwolek, “Detecting human falls with 3-axis accelerometer and depth sensor,” in *Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 770–773.

[12] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for Support Vector Machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1, 2011.