COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 117 (2014) 489–501

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

Human fall detection on embedded platform using depth maps and wireless accelerometer

Bogdan Kwolek[a,*], *Michal Kepski*[b]

[a] *AGH University of Science and Technology, 30 Mickiewicza Av., 30-059 Kraków, Poland*
[b] *University of Rzeszow, 16c Rejtana Av., 35-959 Rzeszów, Poland*

ARTICLE INFO

Article history:
Received 21 May 2014
Received in revised form
7 September 2014
Accepted 23 September 2014

Keywords:
Fall detection
Depth image analysis
Assistive technology
Sensor technology for smart homes

ABSTRACT

Since falls are a major public health problem in an aging society, there is considerable demand for low-cost fall detection systems. One of the main reasons for non-acceptance of the currently available solutions by seniors is that the fall detectors using only inertial sensors generate too much false alarms. This means that some daily activities are erroneously signaled as fall, which in turn leads to frustration of the users. In this paper we present how to design and implement a low-cost system for reliable fall detection with very low false alarm ratio. The detection of the fall is done on the basis of accelerometric data and depth maps. A tri-axial accelerometer is used to indicate the potential fall as well as to indicate whether the person is in motion. If the measured acceleration is higher than an assumed threshold value, the algorithm extracts the person, calculates the features and then executes the SVM-based classifier to authenticate the fall alarm. It is a 365/7/24 embedded system permitting unobtrusive fall detection as well as preserving privacy of the user.

1. Introduction

Assistive technology or adaptive technology is an umbrella term that encompasses assistive and adaptive devices for people with special needs [1,2]. Special needs and daily living assistance are often associated with seniors, disabled, overweight and obese, etc. Assistive technology for aging-at-home has become a hot research topic since it has big social and commercial value. One important aim of assistive technology is to allow elderly people to stay as long as possible in their home without changing their living style.

Wearable sensor-based systems for health monitoring are an emerging trend and in the near future they are expected to make possible proactive personal health monitoring along with better medical treatment. Inertial measurement units (IMUs) are low-cost and low power consumption devices with many potential applications. Current miniature inertial sensors can be integrated into clothes or shoes [3]. Inertial tracking technologies are becoming widely accepted for the assessment of human movement in health monitoring applications [4]. Wearable sensors offer several advantages over other sensors in terms of cost, weight, size, power consumption, ease of use and, most importantly, portability. Therefore, in the last decade, many different methods based on inertial sensors were developed to detect human falls. Falls are a major cause of injury for older people and a significant obstacle in independent living of the seniors. They are one of the top causes of injury-related hospital admissions in people aged 65 years and over. The statistical results demonstrate that at least one-third of people aged 65 years and over fall one or more times a year [5]. An injured elderly may be laying on

[*] *Corresponding author*. Tel.: +48 17 8651592.
E-mail address: bkw@agh.edu.pl (B. Kwolek).

http://dx.doi.org/10.1016/j.cmpb.2014.09.005
0169-2607/© 2014 Elsevier Ireland Ltd. All rights reserved.

the ground for several hours or even days after a fall incident has occurred. Therefore, significant attention has been devoted to developing an efficient wearable system for human fall detection [6–9].

1.1. IMU based approaches to fall detection

The most common method for wearable-device based fall detection consists in the use of a tri-axial accelerometer and a threshold-based algorithm for triggering an alarm. Such algorithms raise the alarm when the acceleration is larger than a threshold value [10]. A variety of accelerometer-based methods and tools have been proposed for fall detection [11]. Typically, such algorithms require relatively high sampling rate. However, most of them discriminate poorly between activities of daily living (ADLs) and falls, and none of which is universally accepted by elderly. One of the main reasons for non-acceptance of the currently available solutions by seniors is that the fall detectors using only accelerometers generate too much false alarms. This means that some daily activities are erroneously signaled as fall, which in turn leads to frustration of the users.

The main reason of high false ratio of accelerometer-based systems is the lack of adaptability together with insufficient capabilities of context understanding. In order to reduce the number of false alarms, many attempts were undertaken to combine both accelerometer and gyroscope [6,12]. However, several ADLs like quick sitting have similar kinematic motion patterns with real falls and in consequence such methods might trigger many false alarms. As a result, it is not easy to distinguish real falls from fall-like activities using only accelerometers and gyroscopes. Another drawback of the approaches based on wearable sensors, from the user's perspective, is the need to wear and carry various uncomfortable devices during normal daily life activities. In particular, the elderly may forget to wear such devices. Moreover, in [13] it is pointed out that the common fall detectors, which are usually attached to a belt around the hip, are inadequate to be worn during the sleep and this results in the lack of ability of such detectors to monitor the critical phase of getting up from the bed.

In general, the solutions mentioned above are somehow intrusive for people as they require wearing continuously at least one device or smart sensor. On the other hand, these systems, comprising various kinds of small sensors, transmission modules and processing capabilities, promise to change the personal care, by supplying low-cost wearable unobtrusive solutions for continuous all-day and any-place health and activity status monitoring. An example of such solutions with a great potential are smart watches and smartphone-based technologies. For instance, in iFall application [14], data from the accelerometer is evaluated using several threshold-based algorithms and position data to determine the person's fall. If a fall is inferred, a notification is raised requiring the user's response. If the user does not respond, the system sends alerts message via SMS.

Despite several shortcomings of the currently available wearable devices, the discussed technology has a great potential, particularly, in the context of growing capabilities of signal processors and embedded systems. Moreover, owing to progress in this technology, data collection is no longer constrained to laboratory environments. In fact, it is the only technology that was successfully used in large scale collection of people motion data.

1.2. Camera based approaches to fall detection

Video-cameras have largely been used for detecting falls on the basis of single CCD camera [15,16], multiple cameras [17], specialized omni-directional ones [18] and stereo-pair cameras [19]. Video based solutions offer several advantages over others including the capability of detection of various activities. The further benefit is low intrusiveness and the possibility of remote verification of fall events. However, the currently available solutions require time for installation, camera calibration and they are not cheap. As a rule, CCD-camera based systems require a PC computer or a notebook for image processing. While these techniques might work well in controlled environments, in order to be practically applied they must be adapted to non-controlled environments in which neither the lighting nor the subject tracking is fully controlled. Typically, the existing video-based devices for fall detection cannot work in nightlight or low light conditions. Additionally, the lack of depth information can lead to lots of false alarms. What is more, their poor adherence to real-life applications is particularly related to privacy preserving. Nevertheless, these solutions are becoming more accessible, thanks to the emergence of low-cost cameras, the wireless transmission devices, and the possibility of embedding the algorithms. The major problem is acceptance of this technology by the seniors as it requires the placement of video cameras in private living quarters, and especially in the bedroom and the bathroom.

The existing video-based devices for fall detecting cannot work in nightlight or low light conditions. In addition, in most of such solutions the privacy is not preserved adequately. On the other hand, video cameras offer several advantages in fall detection over wearable devices-based technology, among others the ability to detect and recognize various daily activities. Additional advantage is low intrusiveness and the possibility of remote verification of fall events. However, the lack of depth information may lead to many false alarms. The existing technology permits reaching quite high performance of fall detection. However, as mentioned above it does not meet the requirements of the users with special needs.

Recently, Kinect sensor has been proposed to achieve fall detection [20–22]. The Kinect is a revolutionary motion-sensing technology that allows tracking a person in real-time without having to carry sensors. It is the world's first low-cost device that combines an RGB camera and a depth sensor. Thus, if only depth images are used it preserves the person's privacy. Unlike 2D cameras, it allows tracking the body movements in 3D. Since the depth inference is done using an active light source, the depth maps are independent of external light conditions. Owing to using the infrared light, the Kinect sensor is capable of extracting the depth maps in dark rooms. In the context of reliable fall detection systems, which should work 24 h a day and 7 days a week it is very important capability, as we already demonstrated in [21].

1.3. Overview of the method

In order to achieve reliable and unobtrusive fall detection, our system employs both the Kinect sensor and a wearable motion-sensing device. When both devices are used our system can reliably distinguish between falls and activities of daily living. In such a configuration of the system the number of false alarms is diminished. The smaller number of false alarms is achieved owing to visual validation of the fall alert generated on the basis of motion data only. The authentication of the alert is done on the basis of depth data and analysis of the features extracted on depth maps. Owing to the determined in advance parameters describing the floor the system analyses not only the shape of the extracted person but also the distance between the person's center of gravity and the floor. In situations in which the use of the wearable sensor might not be comfortable, for instance during changing clothes, bathing, washing oneself, etc., the system can detect falls using depth data only. In the areas of the room being outside of the Kinect field of view the system can operate using data from motion-sensing device consisting of an accelerometer and a gyroscope only. Thanks to automatic extraction of the floor no calibration of the system is needed and Kinect can be placed according to the user preferences at the height of about 0.8–1.2 m. Owing to using of depth maps only our system preserves privacy of people undergoing monitoring as well as it can work at nighttime. The price of the system along with working costs are low thanks to the use of low-cost Kinect sensor and low-cost PandaBoard ES, which is a low-power, single-board computer development platform. The algorithms were developed with respect to both computational demands as well as real-time processing requirements.

The rest of the paper is organized as follows. Section 2 gives an overview of the main ingredients of the system, together with the main motivations for choosing the embedded platform. Section 3 is devoted to short overview of the algorithm. A threshold-based detection of the person fall is described in Section 4. In Section 5 we give details about extraction of the features representing the person in depth images. The classifier responsible for detecting human falls is presented in Section 6. The experimental results are discussed in Section 7. Section 8 provides some concluding remarks.

2. The embedded system for human fall detection

This section is devoted to presentation of the main ingredients of the embedded system for human fall detection. At the beginning, the architecture of the embedded system for fall detection is outlined. Next, the PandaBoard is drafted briefly. Following that, the wearable device is presented in detail. Then, the Kinect sensor and its usefulness for fall detection are discussed shortly. Finally, data processing, feature extraction along with classification modules are discussed briefly in the context of the limited computational resources of the utilized embedded platform.

2.1. Main ingredients of the embedded system

Our fall detection system uses both data from Kinect and motion data from a wearable smart device containing accelerometer and gyroscope sensors. On the basis of data from the inertial sensor the algorithm extracts motion features, which are then used to decide if a fall took place. In the case of the fall the features representing the person in the depth images are dispatched to a classifier, see Fig. 1.

2.2. Embedded platform

The computer used to execute depth image analysis and signal processing is the PandaBoard ES, which is a mobile development platform, enabling software developers access to an open OMAP4460 processor-based development platform. It features a dual-core 1 GHz ARM Cortex-A9 MPcore processor with Symmetric Multiprocessing (SMP), a 304 MHz PowerVR SGX540 integrated 3D graphics accelerator, a programmable C64x DSP, and 1 GB of DDR2 SDRAM. The board 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, Chrome and Linux Ubuntu. The booting of the operating system is from SD memory card. Linux is well suited operating system for real-time embedded platforms since it provides various flexible inter-process communication methods, among others message queues. Another advantage of using Linux in an embedded device is rich availability of tools and therefore it has been chosen for managing the hardware and software of the selected embedded platform.

The data acquired by x-IMU inertial device with 256 Hz are transmitted wirelessly via Bluetooth to the processing device, whereas the Kinect sensor was connected to the device via USB, see Fig. 1. The fall detection system runs under Linux operating system. The application consists of five main concurrent processes that communicate via message queues, see Fig. 1. Message queues are appropriate choice for well

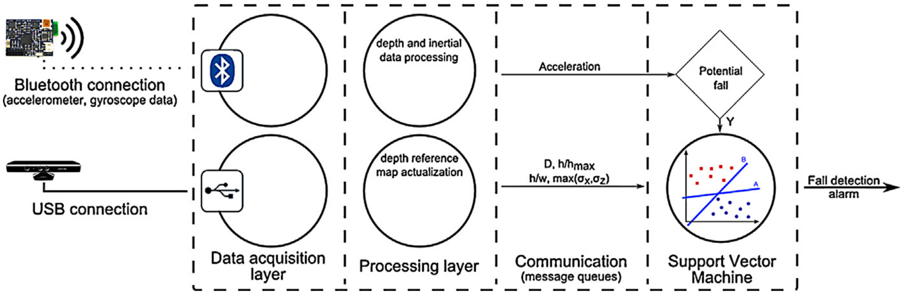

Fig. 1 – The architecture of the embedded system for fall detection.

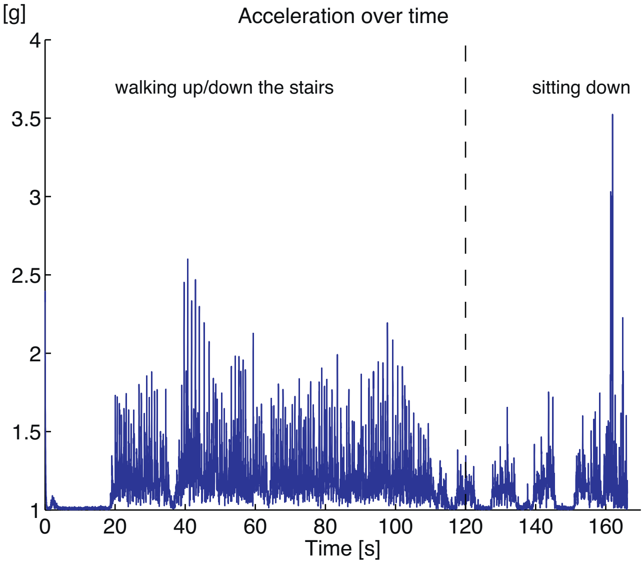

Fig. 2 – Acceleration over time for typical ADLs performed by an elderly.

structured data and therefore they were selected as a communication mechanism between the concurrent processes.

They provide asynchronous communication that is managed by Linux kernel. The first process is responsible for acquiring data from the wearable device, the second one acquires depth data from the Kinect, third process continuously updates the depth reference image, 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 extraction of the person on the basis of the depth reference maps has been chosen since the segmentation can be done with relatively low computational costs. The dual-core processor allows parallel execution of processes responsible for the data acquisition and processing.

2.3. *The inertial device*

The person movement is sensed by an x-IMU [23], which is a versatile motion sensing platform. Its host of on-board sensors, algorithms, configurable auxiliary port and real-time communication via USB, Bluetooth or UART make it powerful smart motion sensing sensor. The on-board SD card, USB-based battery charger, real-time clock and motion trigger wake up also allow on-board storage of data for later analysis. The x-IMU consists of triple axis 16-bit gyroscope and triple axis 12-bit accelerometer. The first sensor measures acceleration, the rate of change in velocity across time, whereas the gyroscope delivers us the rate of change of the angular position over time (angular velocity) with a unit of [deg/s]. The acceleration is measured in units of [g].

The measured acceleration components were median filtered with a window length of three samples to suppress the sensor noise. The accelerometric data were utilized to calculate the acceleration's vector length. Fig. 2 shows a sample plot of acceleration vector length vs. time for a person walking up and down the stairs, and after that sitting down. The plot depicts motion data of a person older than 65 years of age. As illustrated on the discussed plot, for typical daily activities of an elderly the acceleration assumes quite considerable values. As we can observe, during a rapid sitting down the acceleration value equal to 3.5 has been exceeded. Such a value is assumed very often as a decision threshold in simple threshold-based algorithms for fall detection [8,10]. Therefore, in order to reduce the number of false alarms, in addition to the measurements from the inertial sensor we employ the Kinect sensor whenever it is only possible. The depicted plots were obtained for the IMU device that was worn near the pelvis region. It is worth noting that the attachment of the wearable sensor near the pelvis region or lower back is recommended in the literature [11] because such a body part represents the major component of body mass and undergoes movement in most activities.

2.4. *Kinect sensor*

The Kinect sensor simultaneously captures depth and color images at a frame rate of about 30 fps. The device consists of an infrared laser-based IR emitter, an infrared camera and an RGB camera. The depth sensor consists of an infrared laser emitter combined with a monochrome CMOS sensor, which captures 3D data streams under any ambient light conditions. The CMOS sensor and the IR projector form a stereo pair with a baseline of approximately 75 mm. The sensor has an angular field of view of fifty-seven degrees horizontally and forty-three degrees vertically. The minimum range for the Kinect is about 0.6 m and the maximum range is somewhere between 4 and 5 m. The device projects a speckle pattern onto the scene and infers the depth from the deformation of that pattern. In order to determine the depth it combines such a structured light technique with two classic computer vision techniques, namely depth from focus and depth from stereo. Pixels in the provided depth images indicate the calibrated depth in the scene. The depth resolution is about 1 cm at 2 m distance. The depth map is supplied in VGA resolution (640 × 480 pixels) on 11 bits (2048 levels of sensitivity). Fig. 3 depicts sample color images and the corresponding depth maps, which were shot by the Kinect in various lighting conditions, ranging from day lighting to late evening one. As we can observe, owing to the Kinect's ability to extract the depth images in unlit rooms the system is able to detect falls in the late evening or even in the nighttime.

2.5. *The system for fall detection*

The system detects falls using Support Vector Machine (SVM), which has been trained off-line, see Fig. 4. The system acquires depth images using OpenNI (Open Natural Interaction) library. OpenNI framework supplies an application programming interface (API) as well as it provides the interface for physical devices and for middleware components. The acceleration components were median filtered with a window length of three samples. The size of the window has been determined experimentally with regard to noise supression as well as computing power of the utilized platform. A nearest neighbor-based interpolation was executed on the depth

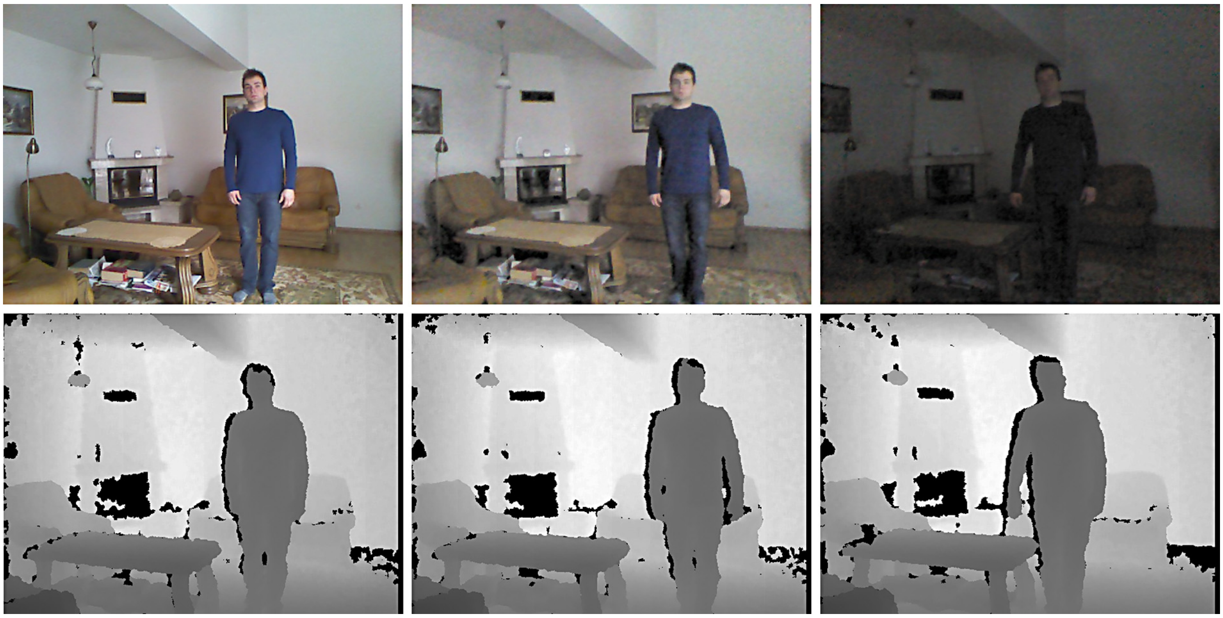

Fig. 3 – Color images (top row) and the corresponding depth images (bottom row) shot by Kinect in various lighting conditions, ranging from day lighting to late evening lighting. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

maps in order to fill the holes in the depth map and to get the map with meaningful values for all pixels. The median filter with a 5 × 5 window on the depth array has been executed to make the data smooth. Afterwards, the features are extracted both on motion data and the depth maps, see Feature Extraction block on Fig. 4. The depth features were then forwarded to the SVM classifier responsible for distinguishing between ADLs and falls.

3. Overview of the algorithm

At the beginning, motion data from IMU along with depth data from the Kinect sensor are acquired. The data is then median filtered to suppress the noise. After such a preprocessing the depth maps are stored in circular buffer see Fig. 5. The storage of the data in a circular buffer is needed for the extraction of the depth reference image, which in turn allows us extraction of the person. In the next step the algorithm verifies if the person is motion. This operation is carried out on the basis of the accelerometric data and thus it is realized with low computational cost. When the person is at rest, the algorithm acquires new data. In particular, no update of the depth reference map takes place if no movement of the person has been detected in one second period. If a movement of the person takes place, the algorithm extracts the foreground. The foreground is determined through subtraction of the current depth map from the depth reference image.

Given the extracted foreground, the algorithm determines the connected components. In the case of the scene change, for example, if a new object appears in the scene, the algorithm updates the depth reference image. We assume that the scene change takes place, when two or more blobs of sufficient area appear in the foreground image. Subsequently, given the accelerometric data, the algorithm examines whether a fall took place. In the case of possible fall the algorithm extracts the person along with his/her features in the depth map. The extraction of the foreground is done through differencing the current depth map from the depth reference map. Next, the algorithm removes from the binary image all connected components (objects) that consist of fewer pixels than assumed number of pixels. After that, the person is

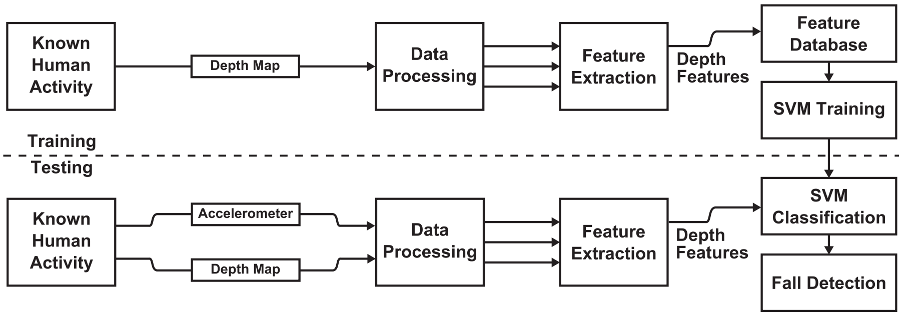

Fig. 4 – An overview of fall detection process.

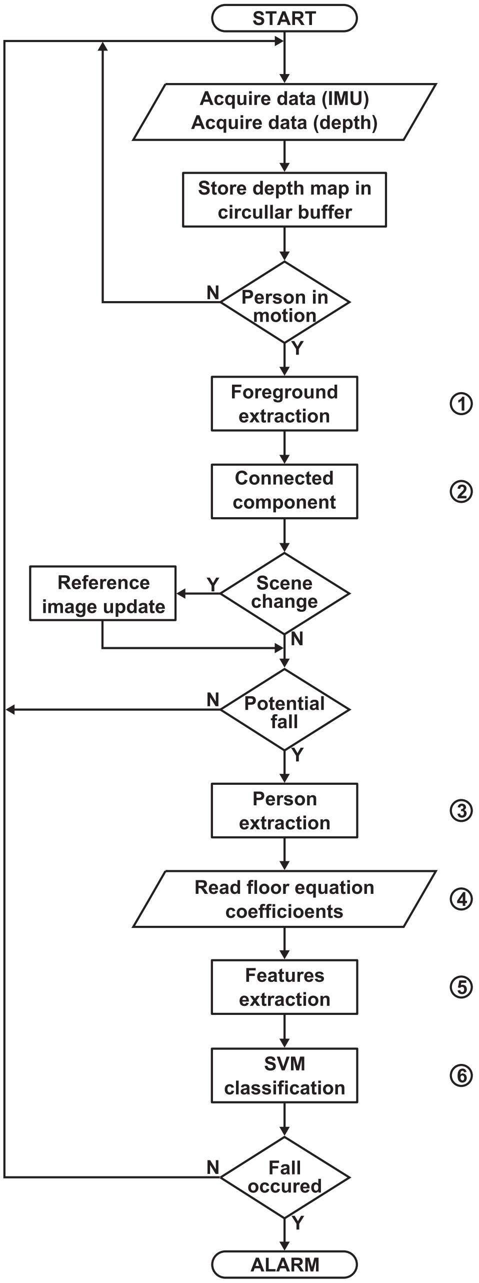

Fig. 5 – Flow chart of the algorithm for fall detection.

segmented through extracting the largest connected component in the thresholded difference map. Finally, the classifier is executed to acknowledge the occurrence of the fall.

4. Threshold-based fall detection

On the basis of the data acquired by the IMU device the algorithm indicates a potential fall. In the flow chart of the algorithm, see Fig. 5, a block Potential fall represents the recognition of the fall using data from the inertial device. Fig. 6 represents sample plots of the acceleration and angular velocities for falling along with daily activities like going down the stairs, picking up an object, and sitting down — standing up.

The x-IMU inertial device consists of triple-axis 12-bit accelerometer and triple-axis 16-bit gyroscope. The sampled acceleration components were used to calculate the total sum vector $SV_{Total}(t)$ as follows:

$$SV_{Total}(t) = \sqrt{A_x^2(t) + A_y^2(t) + A_z^2(t)} \qquad (1)$$

where $A_x(t)$, $A_y(t)$, $A_z(t)$ is the acceleration in the $x-$, $y-$, and $z-$ axes at time t, respectively. The SV_{Total} contains both the dynamic and static acceleration components, and thus it is equal to 1 g for standing, see plots of acceleration change curves in the upper row of Fig. 6. As we can observe on the discussed plots, during the process of falling the acceleration attained the value of 6 g, whereas during walking downstairs and upstairs it attained the value of 2.7 g. It is worth noting that the data were acquired by x-IMU, which was worn by a middle aged person (60+). The plots shown in the bottom row illustrate the corresponding change of angular velocities. As we can see, the change of the angular velocities during the process of falling is the most significant in comparison to non-fall activities. However, in practice, it is not easy to construct a reliable fall detector with almost null false alarms ratio using the inertial data only. Thus, our system employs a simple threshold-based detection of falls, which are then verified on the basis of analysis of the depth images. If the value of SV_{Total} is greater than 3 g then the system starts the extraction of the person and then executes the classifier responsible for the final decision about the fall, see also Fig. 5.

5. Extraction of the features representing person in depth images

In this section we demonstrate how the features representing the person undergoing monitoring are extracted. At the beginning we discuss the algorithm for person delineation in the depth images. Then, we explain how to automatically estimate the parameters of the equation describing the floor. Finally, we discuss the features representing the lying person, given the extracted equation of the floor.

5.1. *Extraction of the object of interest in depth maps*

In order to make the system applicable in a wide range of scenarios we elaborated a fast method for updating the depth reference image. The person was detected on the basis of a scene reference image, which was extracted in advance and then updated on-line. In the depth reference image each pixel assumes the median value of several pixels values from the past images, see Fig. 7. In the set-up stage we collect a number of the depth images, and for each pixel we assemble a list of the pixel values from the former images, which is then sorted in order to extract the median. Given the sorted lists of pixels the depth reference image can be updated quickly by removing the oldest pixels and updating the sorted lists with the pixels from the current depth image and then extracting the median value. We found that for typical human motions, satisfactory results can be obtained using 13 depth images. For the Kinect acquiring the images at 30 Hz we take every fifteenth image.

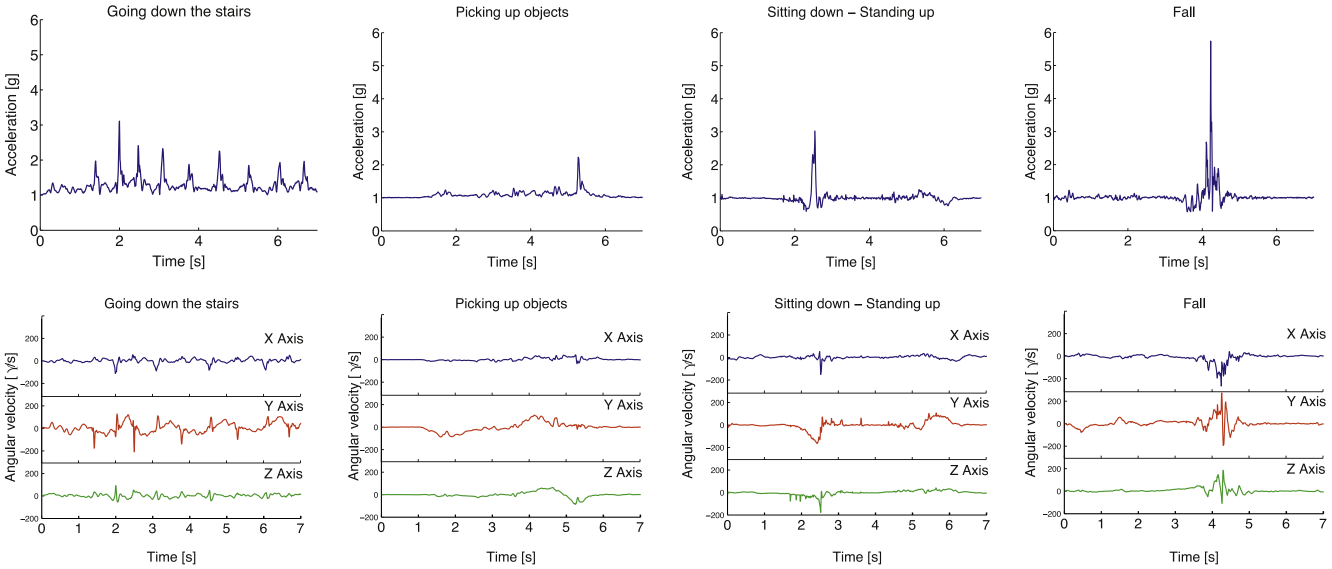

Fig. 6 – Acceleration (top row) and angular velocity (bottom row) over time for walking downstairs and upstairs, picking up an object, sitting down – standing up and falling.

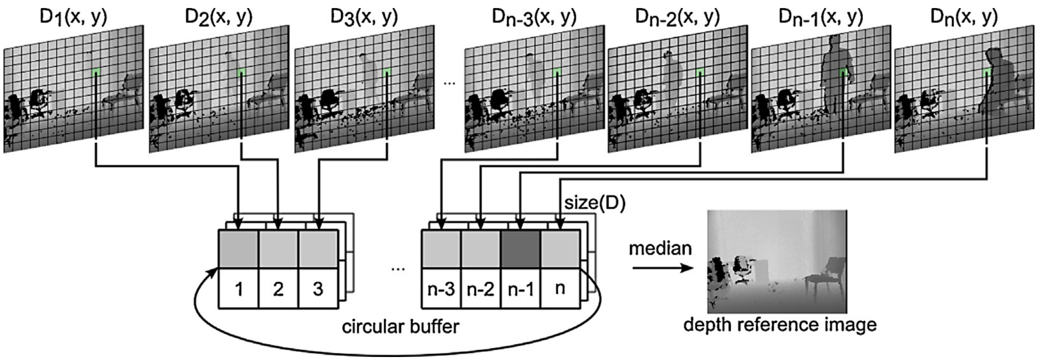

Fig. 7 – Extraction of depth reference image.

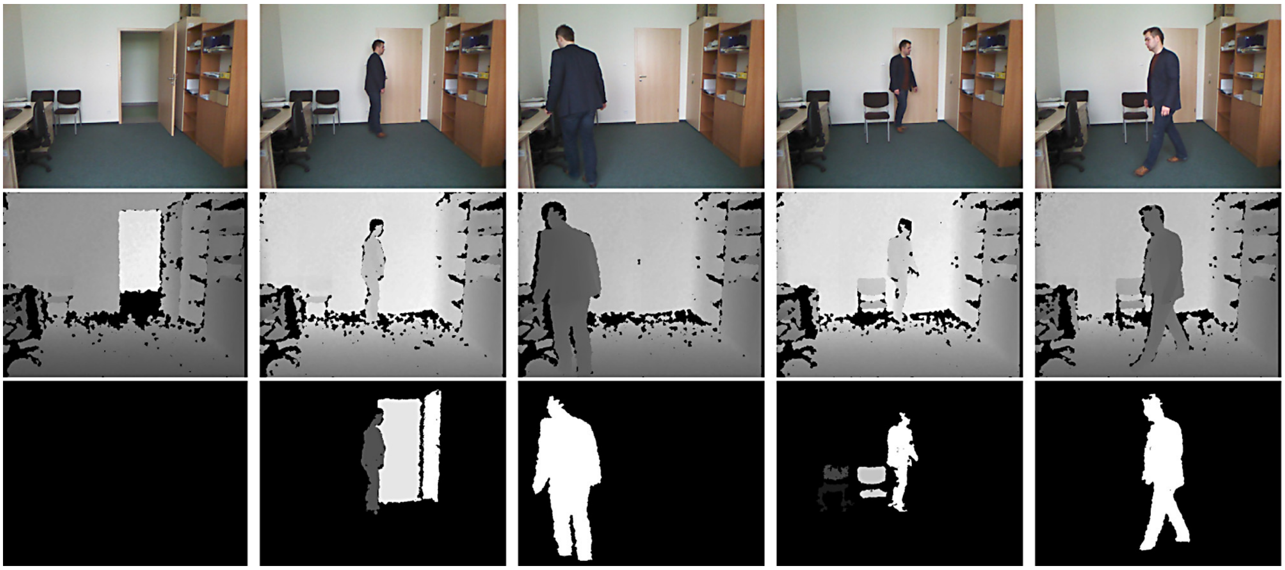

Fig. 8 – Delineation of person using depth reference image. RGB images (upper row), depth (middle row) and binary images depicting the delineated person (bottom row).

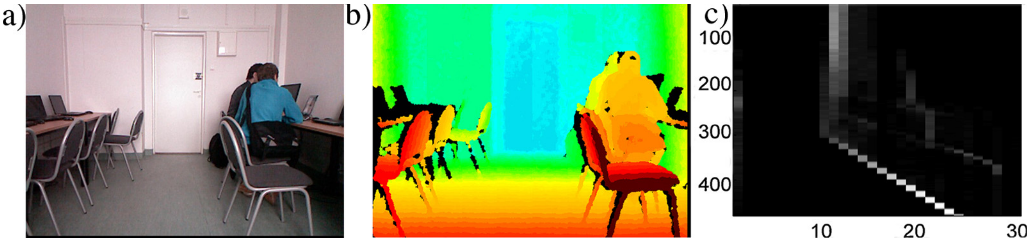

Fig. 9 – V-disparity map extracted on depth images from Kinect sensor: RGB image (a), corresponding depth image (b), v-disparity map (c).

The images shown in the 3rd row of Fig. 8 are the binary images with the foreground objects, which were obtained using the discussed technique. In the middle row there are the raw depth images, whereas in the upper one there are the corresponding RGB images. The RGB images are not processed by our system and they are only depicted for illustrative purposes. In the image #410 the person closed the door, which then appears on the binary image being a difference map between the current depth image and the depth reference image. As we can see, in frame 610, owing to adaptation of the depth reference image, the door disappears on the binary image and the person undergoing monitoring is properly separated from the background. Having on regard that the images are acquired with 25 frames per second as well as the number of frames that were needed to update of the depth reference image, the time required for removing the moved or moving objects in the scene is about six seconds. In the binary image corresponding to the frame 810 we can see a chair, which has been previously moved, and which disappears in the binary image corresponding to frame 1010. Once again, the update of the depth reference image has been achieved in about six seconds. As we can observe, the updated depth reference image allows us to extract the person's silhouette in the depth images. In order to eliminate small objects the depth connected components were extracted. Afterwards, small artifacts were removed. Otherwise, the depth images can be cleaned using morphological erosion.

In the detection mode the foreground objects are extracted through differencing the current image from such a reference depth map. Subsequently, the foreground object is determined through extracting the largest connected component in the thresholded difference map.

The images shown in the middle row of Fig. 8 are the raw depth images. As we already mentioned, the nearest neighbor-based interpolation is executed on the depth maps in order to fill the holes in the maps and to get the maps with meaningful values for all pixels. Thanks to such an interpolation the delineated persons contain a smaller amount of artefacts.

5.2. *V-disparity based ground plane extraction*

In [24] a method based on v-disparity maps between two stereo images has been proposed to achieve reliable obstacle detection. Given a depth map provided by the Kinect sensor, the disparity d can be determined in the following manner:

$$d = \frac{b \cdot f}{z} \quad (2)$$

where z is the depth (in meters), b stands for the horizontal baseline between the cameras (in meters), whereas f stands for the (common) focal length of the cameras (in pixels). The IR camera and the IR projector form a stereo pair with a baseline of approximately $b = 7.5$ cm, whereas the focal length f is equal to 580 pixels.

Let H be a function of the disparities d such that $H(d) = I_d$. The I_d is the v-disparity image and H accumulates the pixels with the same disparity from a given line of the disparity image. Thus, in the v-disparity image each point in the line i represents the number of points with the same disparity occurring in the i-th line of the disparity image. Fig. 9c illustrates the v-disparity image that corresponds to the depth image acquired by the Kinect sensor and depicted on Fig. 9b.

The line corresponding to the floor pixels in the v-disparity map was extracted using the Hough transform (HT). The Hough transform finds lines by a voting that is carried out in a parameter space, from which line candidates are obtained as local maxima in a so-called accumulator space. Assuming that the Kinect is placed at height about 1 m from the floor, the line representing the floor should begin in the disparities ranging from 15 to 25 depending on the tilt angle of the Kinect sensor. As we can observe on Fig. 9c the line corresponding to the floor begins at disparity equal to twenty four.

The line corresponding to the floor was extracted using HT operating o v-disparity values and a predefined range of the parameters. Fig. 10 depicts the accumulator of the HT, that has been extracted on the v-disparity image shown on Fig. 9c. The accumulator was incremented by the v-disparity values. It is worth noting that ordinary HT operating on thresholded v-disparity images often gives incorrect results. For visualization

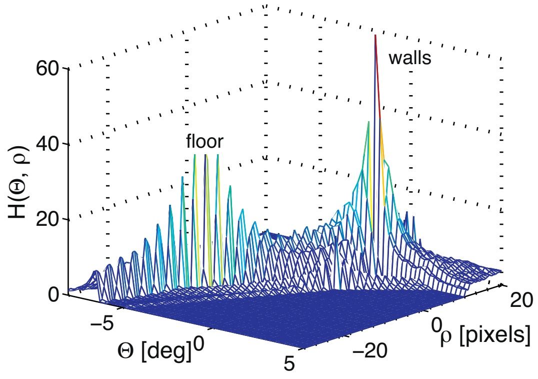

Fig. 10 – Accumulator of the Hough transform operating on v-disparity values from the image shown on Fig. 6c.

purposes the accumulator values were divided by 1000. As we can see on Fig. 10, the highest peak of the accumulator is for a line with Θ approximately equal to zero degrees. This means that it corresponds to a vertical line, i.e. line corresponding to the room walls, see Fig. 9c. In order to simplify the extraction of the peak corresponding to the floor, only the bottom half of the v-disparity maps is subjected to processing by HT, see also Fig. 9c. Thanks to such an approach as well as executing the HT on a predefined range of Θ and ρ, the line corresponding to floor can be estimated reliably.

Given the extracted line in such a way, the pixels belonging to the floor areas were determined. Due to the measurement inaccuracies, pixels falling into some disparity extent d_t were also considered as belonging to the ground. Assuming that d_y is a disparity in the line y, which represents the pixels belonging to the ground plane, we take into account the disparities from the range $d \in (d_y - d_t, d_y + d_t)$ as a representation of the ground plane. Given the line extracted by the Hough transform, the points on the v-disparity image with the corresponding depth pixels were selected, and then transformed to the point cloud.

After the transformation of the pixels representing the floor to the 3D points cloud, the plane described by the equation $ax + by + cx + d = 0$ was recovered. The parameters a, b, c and d were estimated using the RANdom SAmple Consensus (RANSAC) algorithm. RANSAC is an iterative algorithm for estimating the parameters of a mathematical model from a set of observed data, which contains outliers [25]. The distance to the ground plane from the 3D centroid of points cloud corresponding to the segmented person was determined on the basis of the following equation:

$$D = \frac{|aX_c + bY_c + cZ_c + d|}{\sqrt{a^2 + b^2 + c^2}} \tag{3}$$

where X_c, Y_c, Z_c stand for the coordinates of the person's centroid. The parameters should be re-estimated subsequent to each change of the Kinect location or orientation. A relevant method for estimating 3D camera extrinsic parameters has been proposed in [26]. It operates on three sets of points, which are known to be orthogonal. These sets can either be identified using a user interface or by a semi-automatic plane fitting method.

5.3. *Depth features for person detection*

The following features were extracted in a collection of the depth images in order to acknowledge the fall hypothesis, which is signaled by the threshold-based procedure:

- h/w – a ratio of width to height of the person's bounding box
- $h/h_{\max}$ – a ratio expressing the height of the person's surrounding box in the current frame to the physical height of the person
- 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.

Given the delineated person in the depth image along with the automatically extracted parameters of the equation describing the floor, the aforementioned features are easy to calculate.

6. The classifier for fall detection

At the beginning of this section we discuss the dataset that was recorded in order to extract the features for training as well as evaluating of the classifier. After that, we overview the SVM-based classifier.

6.1. *The training dataset*

A dataset consisting of images with normal activities like walking, sitting down, crouching down and lying has been composed in order to train the classifier responsible for examination whether a person is lying on the floor and to evaluate its detection performance. In total 612 images were selected from UR Fall Detection Dataset (URFD)[1] and another image sequences, which were recorded in typical rooms, like office, classroom, etc. The selected image set consists of 402 images with typical ADLs, whereas 210 images depict a person lying on the floor. The aforementioned depth images were utilized to extract the features discussed in Section 5.3. The whole UR Fall Detection dataset consists of 30 image sequences with 30 falls. Two types of falls were performed by five persons, namely from standing position and from sitting on the chair. All RGB and depth images are synchronized with motion data, which were acquired by the x-IMU inertial device.

Fig. 11 depicts the scatter plot, in which a collection of scatter plots is organized in a two-dimensional matrix simultaneously to provide correlation information among the attributes. As we can observe, the overlaps in the attribute space are not too significant. Thus, a linear SVM was utilized for classifying lying poses and typical ADLs. Although the non-linear SVM has usually better effectiveness in classification of non-linear data than its linear counterpart, it has much higher computational demands for prediction.

6.2. *Support vector machines (SVM)*

The basic idea of the SVM classification is to find a separating hyperplane that corresponds to the largest possible margin between the points of different classes [27]. The optimal hyperplane for an SVM means the one with the largest margin between the two classes, so that the distance to the nearest data point of both classes is maximized. Such a largest margin means the maximal width of the tile parallel to the hyperplane that contains no interior data points and thus incorporating robustness into decision making process. Given a set of datapoints $\mathcal{D}$: $\mathcal{D} = \left\{(x_i, y_i) | x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\right\}_{i=1}^{n}$ where each example x_i is a point in p-dimensional space and y_i is the corresponding class label, we search for vector $\omega \in \mathbb{R}^p$ and bias $b \in \mathbb{R}$, forming the hyperplane H: $\omega^T x + b = 0$ that seperates both classes so that: $y_i(\omega^T x_i + b) \geq 1$. The optimization problem that needs to be solved is: $\min_{\omega,b} \frac{1}{2}\omega^T\omega$ subject to: $y_i(\omega^T x_i + b) \geq 1$. The problem consists in optimizing a quadratic function subject to linear constraints, and can be solved with

[1] http://fenix.univ.rzeszow.pl/mkepski/ds/uf.html.

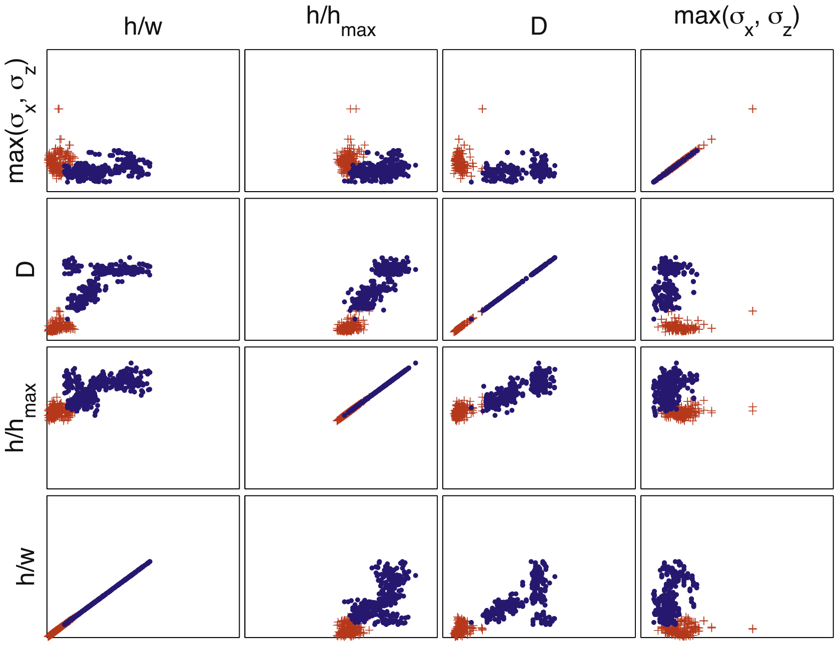

Fig. 11 – Multivariate classification scatter plot for features utilized for training of the fall classifier.

an off-the-shelf Quadratic programming (QP) solver. The linear SVM can perform prediction with p summations and multiplications, and the classification time is independent of the number of support vectors. We executed LIBSVM software [28] on a PC computer to train the fall detector.

7. Experimental results

We evaluated the SVM-based classifier and compared it with a k-NN classifier (5 neighbors). The classifiers were evaluated in 10-fold cross-validation. To examine the classification performances we calculated the sensitivity, specificity, precision and classification accuracy. The sensitivity is the number of true positive (TP) responses divided by the number of actual positive cases (number of true positives plus number of false negatives). It is the probability of fall, given that a fall occurred, and thus it is the classifier's ability to identify a condition correctly.

The specificity is the number of true negative (TN) decisions divided by the number of actual negative cases (number of true negatives plus number of false positives). It is the probability of non-fall, given that a non-fall ADL took place, and thus it shows how good a classifier is at avoiding false alarms. The accuracy is the number of correct decisions divided by the total number of cases, i.e. the sum of true positives plus sum of true negatives divided by total instances in population. That is, the accuracy is the proportion of true results (both true positives and true negatives) in the population. The precision or positive predictive value (PPV) is equal to true positives divided by sum of true positives and false positives. Thus, it shows how many of the positively classified falls were relevant. In Table 1 are shown results that were obtained in 10-fold cross-validation by the classifier responsible for the lying pose detection and the aforementioned dataset. As we can see, both specificity and precision are equal to 100%, i.e. the ability of the classifier to avoid false alarms and its exactness assume perfect values.

Table 2 shows results of experimental evaluation of the system for fall detection, which were obtained on depth image sequences from URFD dataset. They were obtained on thirty image/acceleration sequences with falls and thirty image/acceleration sequences with typical ADLs like sitting down, crouching down, picking-up an object from the floor and lying on the sofa. The number of images in the sequences with falls is equal to 3000, whereas the number of images with sequences with ADLs is equal to 9000. All images have

Table 1 – Performance of lying pose classification.

	Fall	No Fall	
SVM			
Fall	208	0	Accuracy = 99.67%
No fall	2	402	Precision = 100%
	Sens. = 99.05%	Spec. = 100%	
k-NN			
Fall	208	0	Accuracy = 99.67%
No fall	2	402	Precision = 100%
	Sens. = 99.05%	Spec. = 100%	

Table 2 – Performance of fall detection.

	SVM – depth only	SVM – depth + acc.	Threshold UFT [10]	Threshold LFT [10]
Accuracy	90.00%	98.33%	95.00%	86.67%
Precision	83.30%	96.77%	90.91%	82.35%
Sensitivity	100.00%	100.00%	100.00%	93.33%
Specificity	80.00%	96.67%	90.00%	80.00%

corresponding motion data. In the case of incorrect response of the system the remaining part of the sequence has been omitted. This means that the detection scores were determined on the basis of the number of the correctly/incorrectly classified sequences. As we can observe, the Threshold UFT method [10] achieves good results. The results obtained by SVM-classifier operating on only depth features are slightly worse than results of Threshold UFT method. The reason is that the update of the depth reference image was realized without the support of the motion information. This means that a simplified system has been built using only blocks, which are indicated in Fig. 5 as numerals in circles. In particular, in such a configuration of the system all images are processed in order to extract the depth reference image. The algorithm using both motion data from accelerometer and depth maps for verification of IMU-based alarms achieves the best performance. Moreover, owing to the use of the IMU device the computational efforts associated with the detection of the person in the depth maps are much smaller.

Five volunteers with age over 26 years attended in evaluation of our developed algorithm and the embedded system for fall detection. Intentional falls were performed in an office by five persons toward a carpet with thickness of about 2 cm. The x-IMU device was worn near the pelvis. 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 a settee. All intentional falls have been detected correctly. In particular, quick sitting down, which is not easily distinguishable ADL from an intentional fall when only an accelerometer or even both an accelerometer and a gyroscope are used, has been classified as an ADL.

It is well known that the precision of Kinect measurements decreases in strong sunlight. In order to investigate the influence of sunlight on the performance of fall detection we analyzed the person extraction in depth maps acquired in strong sunlight, see sample images on Fig. 12. As noted in [29], the in-home depth measurements on a person being in sunlight, i.e. in sunlight that passes a closed window can be made with limited complications. Some body parts of a person in sunlight may not return measurements, see images (d and e) in Fig. 12. As we can see, the measurements in image (f) are better in comparison to the measurements shown in image (e) due to smaller sunlight intensity. In order to assess the influence of such strong sunlight on the performance of fall detection we calculated the person depth features in a collection of such depths maps with missing measurements. As expected, the change of the features in comparison to the features extracted on manually segmented persons is not significant, i.e. within several percent. Such change of the values of the features does not degrade the high performance of fall detection given the algorithms used here.

The system that was evaluated in such a way has been implemented on PandaBoard-ES platform. In particular, we trained off-line the SVM classifier, and then used the parameters that were obtained in such a way in a fall predictor, executed on the PandaBoard. Prior the implementation of the system on the PandaBoard we compared processor performances using Dhrystone 2 and Double-Precision Whetstone benchmarks. Our experimental results show that the Dhrystone 2 score on Intel i7-3610QM 2.30 GHz and PandaBoard ES OMAP4460 is equal to 37423845 and 4214871 [lps], respectively, whereas Double-Precision Whetstone is equal to 4373 and 836 [MWIPS], respectively. This means that PandaBoard offers considerable computational power. Finally, the whole system was

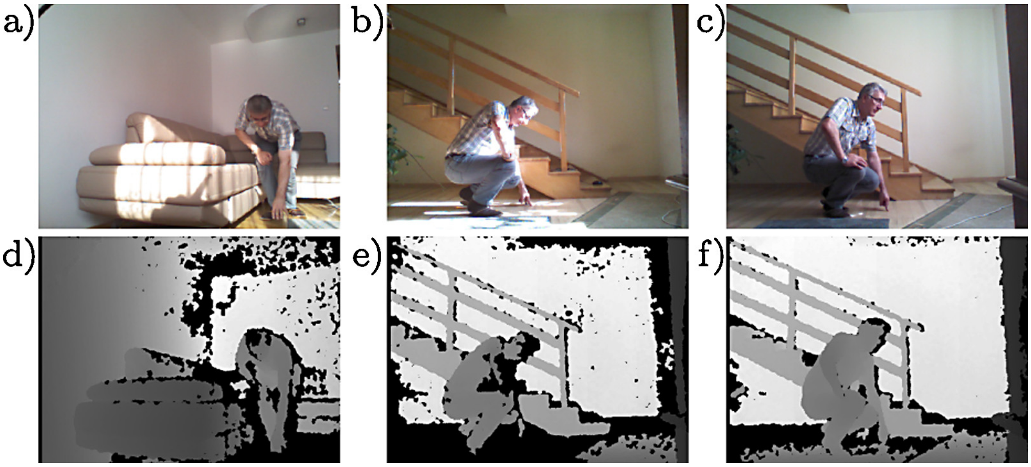

Fig. 12 – Color images (top row) and the corresponding depth images (bottom row) acquired by Kinect in sunlight. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

implemented and evaluated on PandaBoard. The code profiler reported about 50% CPU usage by the module responsible for update of the depth reference map.

8. Conclusions

In this paper we demonstrated an embedded system for reliable fall detection with very low false alarm. The detection of the fall is done on the basis of accelerometric data and depth maps. A tri-axial accelerometer is used to indicate the potential fall as well as to indicate if the person is in motion. If the measured acceleration is higher than an assumed threshold value, the algorithm extracts the person, calculates the features and then executes the SVM-based classifier to authenticate the fall alarm. We demonstrate that person surrounding features together with the distance between the person center of gravity and floor lead to reliable fall detection. The parameters of the floor equation are determined automatically. The extraction of the person is only executed if the accelerometer indicates that he/she is in motion. The person is extracted through the differencing the current depth map from the on-line updated depth reference map. The system permits unobtrusive fall detection as well as preserves privacy of the user. However, a limitation of the Kinect sensor is that sunlight interferes with the pattern-projecting laser, so the proposed fall detection system is most suitable for indoor use.

Conflict of interests

The authors have no conflict of interests.

Acknowledgement

This work has been supported by the National Science Centre (NCN) within the project N N516 483240.

REFERENCES

[1] M. Chan, D. Esteve, C. Escriba, E. Campo, A review of smart homes – present state and future challenges, Comput. Methods Programs Biomed. 91 (1) (2008) 55–81.

[2] A.M. Cook, The future of assistive technologies: a time of promise and apprehension, in: Proc. of the 12th Int. ACM SIGACCESS Conf. on Comp. and Accessibility, ACM, New York, USA, 2010, pp. 1–2.

[3] F. Hoflinger, J. Muller, R. Zhang, L.M. Reindl, W. Burgard, A wireless micro inertial measurement unit (imu), IEEE Trans. Instrum. Meas. 62 (9) (2013) 2583–2595.

[4] F. Buesching, U. Kulau, M. Gietzelt, L. Wolf, Comparison and validation of capacitive accelerometers for health care applications, Comput. Methods Programs Biomed. 106 (2) (2012) 79–88.

[5] S. Heinrich, K. Rapp, U. Rissmann, C. Becker, H.-H. König, Cost of falls in old age: a systematic review, Osteoporos. Int. 21 (2010) 891–902.

[6] N. Noury, A. Fleury, P. Rumeau, A.K. Bourke, G.O. Laighin, V. Rialle, J.E. Lundy, Fall detection – principles and methods, in: IEEE Int. Conf. on Eng. in Medicine and Biology Society, 2007, pp. 1663–1666.

[7] X. Yu, in: 10th Int. Conf. on E-health Networking, Applications and Services, Approaches and principles of fall detection for elderly and patient. (2008) 42–47.

[8] R. Igual, C. Medrano, I. Plaza, Challenges, issues and trends in fall detection systems, Biomed. Eng. Online 12 (1) (2013) 1–24.

[9] M. Mubashir, L. Shao, L. Seed, A survey on fall detection: principles and approaches, Neurocomputing 100 (2013) 144–152.

[10] A.K. Bourke, J.V. O'Brien, G.M. Lyons, Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm, Gait & Posture 26 (2) (2007) 194–199.

[11] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, T. Jamsa, Comparison of low-complexity fall detection algorithms for body attached accelerometers, Gait & Posture 28 (2) (2008) 285–291.

[12] A.K. Bourke, G.M. Lyons, A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor, Med. Eng. Phys. 30 (1) (2008) 84–90.

[13] T. Degen, H. Jaeckel, M. Rufer, S. Wyss, SPEEDY: a fall detector in a wrist watch, in: Proc. of the 7th IEEE Int. Symp. on Wearable Comp., Washington, DC, USA, IEEE Computer Society, 2003, p. 184.

[14] F. Sposaro, G. Tyson, ifall: an Android application for fall monitoring and response, in: IEEE Int. Conf. on Engineering in Medicine and Biology Society, 2009, pp. 6119–6122.

[15] D. Anderson, J.M. Keller, M. Skubic, X. Chen, Z. He, Recognizing falls from silhouettes, in: Annual Int. Conf. of the Engineering in Medicine and Biology Society, 2006, pp. 6388–6391.

[16] C. Rougher, J. Meunier, A. St-Arnaud, J. Rousseau, Monocular 3D head tracking to detect falls of elderly people, in: Annual Int. Conf. of the IEEE Eng. in Medicine and Biology Society, 2006, pp. 6384–6387.

[17] R. Cucchiara, A. Prati, R. Vezzani, A multi-camera vision system for fall detection and alarm generation, Expert Syst. 24 (5) (2007) 334–345.

[18] S.-G. Miaou, P.-H. Sung, C.-Y. Huang, A customized human fall detection system using omni-camera images and personal information, in: Distributed Diagnosis and Home Healthcare, 2006, pp. 39–42.

[19] B. Jansen, R. Deklerck, Context aware inactivity recognition for visual fall detection, in: Proc. IEEE Pervasive Health Conference and Workshops, 2006, pp. 1–4.

[20] M. Kepski, B. Kwolek, I. Austvoll, Fuzzy inference-based reliable fall detection using Kinect and accelerometer The 11th Int. Conf. on Artificial Intelligence and Soft Computing. LNCS, vol. 7267, Springer, 2012, pp. 266–273.

[21] M. Kepski, B. Kwolek, Fall detection on embedded platform using Kinect and wireless accelerometer 13th Int. Conf. on Computers Helping People with Special Needs. LNCS, vol. 7383, Springer, 2012, pp. 407–414.

[22] G. Mastorakis, D. Makris, Fall detection system using Kinect's infrared sensor, J. Real-Time Image Process. (2012) 1–12.

[23] 3D Orientation Sensor IMU. http://www.test.org/doe/

[24] R. Labayrade, D. Aubert, J.-P. Tarel, Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation, in: Intelligent Vehicle Symposium, 2002, IEEE, vol 2, 2002, pp. 646–6512.

[25] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.

[26] R. Deklerck, B. Jansen, X.L. Yao, J. Cornelis, Automated estimation of 3d camera extrinsic parameters for the

monitoring of physical activity of elderly patients, in: XII Mediterranean Conference on Medical and Biological Engineering and Computing, IFMBE Proceedings, vol. 29, 2010, pp. 699–702.

[27] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.

[28] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 1–27.

[29] E.E. Stone, M. Skubic, Unobtrusive, continuous, in-home gait measurement using the microsoft kinect, IEEE Trans. Biomed. Eng. 60 (10) (2013) 2925–2932.