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

Michal Kepski[1] and Bogdan Kwolek[2,*]

[1]University of Rzeszow, Pigonia 1, 35-310 Rzeszow, Poland
[2]AGH University of Science and Technology, 30 Mickiewicza Av., 30-059 Krakow, Poland
[*]http://home.agh.edu.pl/~bkw/contact.html

Abstract: In this paper we present efficient and effective algorithms for fall detection on the basis of sequences of depth maps and data from a wireless inertial sensor worn by a monitored person. A set of descriptors is discussed to permit distinguishing between accidental falls and activities of daily living. Experimental validation is carried out on freely available dataset consisting of synchronized depth and accelerometric data. Extensive experiments are conducted in the scenario with a static camera facing the scene and an active camera observing the same scene from above. Several experiments consisting in person detection, tracking and fall detection in real-time are carried out to show efficiency and reliability of the proposed solutions. The experimental results show that the developed algorithms for fall detection have high sensitivity and specificity.

1. Introduction

With the rapidly growing aging population on a global scale, the need of improving elderly well-being is getting crucial. Smart home technologies can be utilized as means to improve both the quality of care and wellbeing of dependent people. Its form called assistive domotics focuses on making it possible for seniors and people with disabilities to remain at home, safe and comfortable. Smart home technologies are becoming a viable option for older adults who would prefer to stay in the comfort of their homes in place of move to a retirement home or a healthcare facility [1].

The aim of user-centered ubiquitous computing is to develop solutions yielding personal assistance, which at the same time sense variations in human environment and dynamically respond to user needs. It is self-evident that such technology has strong potential to cope with major societal challenges posed by aging society [2]. Such an increased level of intelligence has a potential to provide improved quality of care in addition to helping elderly people access the knowledge required to offer better decisions when interacting with smart environments [3]. One of the crucial factors that at present pose serious bottlenecks to augment people's lives with ubiquitous computing in a broader scale is reduced number of affordable energy-saving devices for human activity monitoring and/or energy-efficient units.

Falls are leading cause of morbidity from injury and mortality in the elderly. They are the major reason of injury-related hospitalization in persons aged 65 years and over and account for significant fraction of all hospital admissions in this age-group [4]. Even falls that do not lead to physical injuries can result in the so called post-fall syndrome [5], which typically manifests itself in loss of confidence, loss of muscle and control, problems with balance, and walking disorders leading to loss of mobility and independence. Fear of falling has been identified as one of the key symptoms of this syndrome. The cause for this is that seniors are afraid to lie after the fall on the floor in solitude and without help for a long time [6]. It has also been shown that getting up quickly after the fall can reduce the risk of death even by 80% and the necessity of hospitalization by 26% [7]. Thus, falls should be detected as early as possible. For this reason, assistive technologies have strong potential not only to assist in daily activities, but they also have capabilities to reduce risks of fall events.

With the goal to permit prolonged independent living in a secure and homely environment, reliable fall detection is a significant task in the area of Ambient Assisted Living (AAL) [8]. Medical alert systems were introduced in the 1980s as uncomplicated push-button devices worn around the neck. Afterwards, they were extended about accelerometer-based algorithms to automatically raise an appropriate alert that a fall has occurred. One of the biggest limitations of such automatic fall detection systems is occurrence of false alarm alerts. When such a device is used, the false alarm could be triggered by an everyday activity such as quickly dropping into a seated position in a chair or even by bending down.

The ambient device-based systems are capable of detecting falls in a non-intrusive way by exploiting audio, vibration, pressure and visual information, to name a few of the most frequently used sources of information in this domain [1]. There are several types of sensors used in this field, including measuring the vibration of the floor to detect falls [9], detecting falls by using pressure mats [10] or impulse-radar sensors [11]. A fall detection system relying on one of the mentioned above sensors typically has a high false alarm rate. None of the sensors mentioned above, if used separately in a fall detection system, is able to meet the requirements of end-users regarding the level of false alarms. Despite the enormous effort of research and the number of IMU-based devices on the market, there is still no system that has sufficient reliability and is accepted by end-users.

To overcome the limitations of these devices, a wide range of vision-based monitoring systems with fall detection functionality have been proposed over recent years [12][13][14]. Vision-based systems usually use image sequences to analyze motion features of human body, and distinguish features of fall events from non-fall activities in order to infer about occurrence of fall. They provide valuable information for assistive monitoring but raise privacy concerns. Besides, such systems fail to work in darkness or when the elderly is outside of the observed area. Their major advantage is that the user does not need to wear any specialized apparatus. However, most of these solutions are energy demanding and expensive. Moreover, their deployment is cumbersome, and only a few of them can meet the demands of the end-users in the detection of the fall in real homes or health care facilities.

Event-driven systems, in which the system activities are triggered in response to events, usually representing a significant change of the state of controlled or monitored physical variables, exhibit certain advantages over other approaches, particularly, in resource-constrained applications. Last years have witness an upsurge in the research interest to harness the advantages of event-based paradigm applied to a wide spectrum of engineering disciplines including signal processing and control. Application areas of such systems include energy-efficient control, energy-efficient signal processing, rehabilitation [15], event-driven visual attention [16], or frame-free event-driven vision systems [17], to mention a few.

In this work, we present an event-driven system for fall event detection using measurements from a body-worn accelerometer and depth sensor(s). In response to significant motion variation indicated by an accelerometer, the system fetches depth maps from a circular buffer and then processes them to validate the fall event. This way, the most time consuming depth image processing is executed only in case of a significant change of the person's motion, i.e. high likelihood of fall occurrence. We discuss and compare the efficiency of fall detection on depth maps provided by a ceiling mounted camera as well as a wall-mounted (facing the user) camera. With the purpose of extending the viewing area the overhead camera was mounted on a pan-tilt motorized head. The aim of the controller of the pan-tilt unit was to keep the moving person in the center of the current depth map. We discuss the person delineation as well as feature extraction in both camera settings. We show experimentally that the results achieved in both camera settings are promising. We discuss the advantages and limitations of the considered camera setups. We show experimentally that a two-camera system achieves perfect classification performance on data from freely available URFD dataset.

2. Background and Related Work

Recent fall alert devices are usually able to recognize when a person wearing the device has fallen by using accelerometers and optionally gyroscopes, and through detecting changes in the body's orientation and speed. An obvious limitation of such devices is that the senior may be not capable to press the emergency button after the fall due to loss of consciousness or just because of over-excitation. Thus, the applicability of such devices is limited to niche markets as nursing homes. Moreover, today's devices are not widely accepted by primary end-users [7], particularly those who are not impaired. The reason for this is that current systems are not able to guarantee good sensitivity (nearly 100%) with enough specificity to limit the number of false alarms [18][19]. An alarm is false when it is triggered unnecessarily or for cause other than fall event. In practice, this means that certain fall-like activities activate alarms, which in turn lead to irritation of the end-users. The reason for this is that the acceleration ranges are overlapping for falls and activities of daily living (ADLs).

Besides solutions outlined above, more complex systems are now utilized to improve the fall detection accuracy [20][8]. Such fall detection systems can be divided into two major categories, that is, based on wearable sensors and context-aware systems [4]. Micro-electro-mechanical systems (MEMS) are extensively used in a wide range of applications. MEMS accelerometers are one of the most common types of MEMS sensors, due to their simplicity, ease of fabrication, low price and good usability [21]. In comparison to vision sensors, wearable inertial sensors are lighter, smaller, easier to use, and most importantly, they consume less energy and are far cheaper. They allow collection of data outside of laboratory environments and are perceived as one of the best sensors for AAL [1]. Hence, many different algorithms have been proposed to explore, support or improve fall detection using only accelerometer(s) [22] or an inertial measurement unit(s) [23][24].

Usually, approaches relying on body-worn accelerometer utilize a threshold-based algorithm to examine if a person's movement is higher than some preset threshold [25][22]. However, as shown in [19], such systems are too sensitive and thus generate substantial number of false alarms. A similar conclusion has been drawn by an international group of researchers [18], which evaluated the effectiveness of threshold-based algorithms to identify falls on data from real falls. The dataset of 29 real–world falls contains accelerations of persons' movement, each for a period of two days. In the evaluation, thirteen different algorithms were examined with respect to their capability of identifying of real falls. Regrettably, none of the examined algorithms gave satisfactory results in terms of both sensitivity (capability of recognizing falls that in reality took place) and specificity (ability to properly recognize a movement as a non-fall).

Fall detection methods relying on body-worn accelerometers can be ineffective in detection of slow falls [26], such as collapsing after a heart attack, which usually do not feature significant accelerations. Moreover, as noticed in [27][18], some fall phases detected in experimentally simulated falls are usually not detectable in acceleration signals from heterogeneous realworld falls. Thus, having on regard reduced availability of motion data with real-world falls, the usefulness of machine learning-based methods might be limited in practice. In general, although it is not easy to distinguish between falls and fall-like activities, the inertial sensors are thought to be very useful sensors in fall detection [4]. They are now massively utilized in the mobile smart devices, including smartwatches. Smartwatches are lightweight and waterproof, so they can be kept on when taking a bath. Overall, the inertial sensors can greatly support fall detectors built on other sensors, including vision and depth sensors [28][29][20].

Among the possible types of context-aware detectors, vision systems offer a promising way of recognizing of human actions [30] as well as detecting human falls [31]. Variety of vision-based fall detection algorithms have been developed in recent years [13][14]. One of their advantages is that the monitored person does not need to wear any special apparatus. On the other hand, this form of fall-monitoring is both most intrusive and most expensive. Despite many approaches to preserve privacy, people in the observed spaces still have feel of being-watched. In consequence, the usual CMOS/CCD cameras are very often unacceptable, especially in the bedrooms or bathrooms. Moreover, while near-infrared light sources made it possible to record video in low-light conditions and during the night, the quality of the videos might be insufficient to achieve automatic fall detection with high sensitivity and specificity. Nevertheless, thanks to technology progress in the area of smart camera [32] and smart home, the CMOS/CCD camera-based systems offer many monitoring capabilities.

The cameras providing in real-time the depth maps can considerably enhance the detection and tracking performance making possible to reliably extract the head trajectory, which has been proven to be very useful in fall detection [33]. An entire view of the scene can be very advantageous in fall detection. In [34], it has been demonstrated how the omnidirectional cameras can be utilized to achieve coupled fall detection and tracking. In general, the omnidirectional cameras have proven to be the very useful if big visual field coverage is desired. Thermal video cameras, which detect the amount of thermal radiation emitted/reflected from objects in the scene can also provide very informative information for reliable fall detection [35].

Just a few years ago, the Kinect's sensor has been proposed for detection of humans' falls [36][28][37]. As shown experimentally [36][28], the depth maps delivered by the Kinect sensor are enough to extract the person from the background. What is more, owing to estimation of dense depth maps on the basis of speckle pattern of infrared laser light, the detection of the person can be done anytime. Despite several approaches to Kinect-based fall detection [37], the existing algorithms do not provide both high sensitivity and specificity. By integrating the acceleration data with video or depth maps [38][28], the recognition of activities [29] as well as emergency situations can be noticeably improved.

Our work differs from research in the area of fall detection (e.g. [37]) in that we do not use only inertial device or depth maps standalone but we use both an inertial unit and depth sensor. The rationale for such an approach is that the current accelerometer-based algorithms being to sensitive, generate too much false alarms. Assuming that such algorithms typically produce a few alarms a day [19], our approach is able to reduce the false alarm ratio to almost null. Having on regard that accelerometers are frequently available in smartwatches, as well as considering further progress in this area, the obtrusiveness and the discomfort when wearing such a device will be limited. With a view to high computation cost of vision-based algorithm for fall detection we apply event-driven approach to data processing and system design. This allows us not only to reduce computational overload but also allows us to determine precisely the time at which the impact took place. Since in the relevant literature [37][14] there is no detailed comparison of approaches for fall detection on the basis of ceiling-mounted and wall-mounted cameras, we discuss algorithms for both approaches as well as present experimental results on freely available fall detection dataset. Since in the second approach an active camera is recommended in order to extend the observation area, we develop an effective algorithm for person's head detection and tracking. We show experimentally that the results achieved by this algorithm on maps acquired by the active camera are promising.

3. Architecture and Main Components of the System

While embedded vision is comparatively a novel term, as a technology, it is highly established in a number of domains, and the most successful applications are in the area of factory automation. Smart cameras are another example of successful applications of the embedded vision technology [32]. In addition, one can specify a number of successful applications of this technology in surveillance and transport. Healthcare is one of the main application areas for the embedded vision [3]. The embedded vision technology has significant potential to change the health monitoring in home, for instance through mobile phone applications for monitoring the user's state of health and reporting it to a medical center. One of successful examples of embedded vision systems is the Microsoft Kinect game controller [39], which has been designed to perform real-time tracking of the movement of the users. Although Kinect was initially devised only as a motion sensing device for computer games, a strong interest of the computer vision community led to developing several new applications, including applications for activity recognition [30] or rehabilitation [37].

In our approach, a body-worn accelerometer is utilized to indicate a potential fall event and a depth camera is employed to authenticate fall alert. The proposed event-driven sensor data processing method fetches from a circular buffer a sequence of depth maps, which were acquired prior to the fall and then processes it to authenticate the fall alert, instead of processing data frame-by-frame, see Fig. 1. In general, if the person acceleration is higher than a predetermined threshold the algorithm executes a lying pose detector as well as optionally employs a dynamic feature to finally confirm the fall. In consequence, more computationally demanding authentication of the fall is not processed frame-by-frame. Such data streams processing has been designed specifically to operate with the least amount of energy consumed while achieving reliable fall detection in real-time [40].

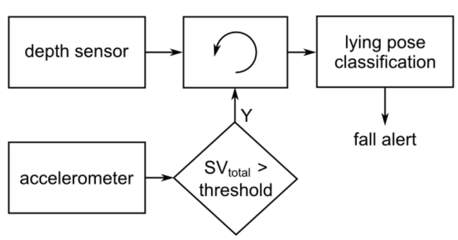

Fig. 1: Event-driven architecture for fall detection.

The presented system can operate in two main modes. In the first mode the fall authentication is achieved using depth maps acquired by a static depth sensor facing the scene, whereas in the second one the verification of the fall is achieved using depth maps provided by an active ceiling-mounted camera. The main difference between the modes of the system lies in the person detection algorithm. In the first mode with a static depth sensor, the person is extracted by differencing the current depth map from an accommodated depth map of the background. In the second mode with the active camera, the person is delineated using depth region growing followed by a person's head detector. If a person moves the system delineates the person in each frame to extract his/her centroid, which is in turn required by a controller of the active camera to keep the target in the center of the current depth map.

The fall detection algorithm is executed on a PC or on PandaBoard depending on the configuration. It runs under Linux operating system. The accelerometric data are acquired by x-IMU device and then transmitted by Bluetooth to the receiver device of the processing unit. The Xbox Kinect sensor is connected to the processing board via USB. The connection between the microcontroller of the active camera and the board is realized by I2C bus.

4. Person Detection in Depth Maps

In this Section we discuss algorithms for person extraction in depth map sequences. The next Subsection is devoted to explaining how person is delineated in depth maps acquired by a Kinect facing the scene, whereas the subsequent Subsection details the method for person detection in depth maps acquired by a Kinect and mounted on the ceiling, i.e. providing the top view of the scene.

4.1. Person Detection in Frontal Depth Maps

The key technology behind Kinect is a variant of structured light in which a pseudo-random speckle pattern is projected onto the scene by a laser-based IR emitter and then observed by an IR camera. The shift of such a speckle pattern in space is measured and after that mapped to depth through triangulation. However, the depth maps acquired in such a way often contain much noise. Thus, typical detectors when trained from the widely applied image feature descriptors, which demonstrated to be successful on visible images, cannot achieve promising results. As demonstrated in [36][28], depth information is sufficient to extract human by the use of depth background maps collected by a fixed Kinect. As noticed in [28], person extraction on the basis of depth background maps can be done at low computational cost.

In our event-driven approach the algorithm extracts the person at low computational cost and then processes the foreground image to prove whether a more costly update of depth background is needed. Moreover, the accommodation of depth background maps is done only in map areas in which the scene took change. The scene changes are detected with low computational cost through extracting coherent depth maps on the foreground map and then examining if the size of the component with person increased considerably, for instance due to opening a door, or the number of the foreground components is larger than one, i.e. if there is an non-person object of sufficient size in the foreground.

In the person extraction algorithm we can distinguish a part that is executed every frame and a part, which is evoked when there is a scene change, see lines 1-7,22 and 8-20 in Algorithm 1, respectively. Let us assume that there is given a background model $B(x, y)$ and a buffer Q consisting of Q_{size} last depth frames. In each frame the algorithm takes a new frame $D_t(x, y)$ and then extracts the foreground $F_t(x, y)$ through determining the absolute value of difference between the current depth map and the depth background map, see lines 2-3. Then, the algorithm determines the connected components in the binarized foreground images, calculates their number as well as their areas. Afterwards, it determines the blob belonging to person, see line 5, and stores it on the person image $P_t(x, y)$. Finally, it examines if the number of blobs is larger than one or there is a significant change of the person area on consecutive person images. If both conditions are false the algorithm returns the previous background model, see line 22-23.

Algorithm 1 Person extraction using depth reference map

1: Given a background model $B(x, y)$, and buffer
   $Q = \{D_{t-1}(x, y), D_{t-2}(x, y), \ldots, D_{t-Q_{size}}(x, y)\}$
2: Acquire new depth map $D_t(x, y)$ ▷ Fig. 2c
3: Determine foreground

$$F_t(x,y) = \begin{cases} D_t(x,y), & \text{if } |D_t(x,y) - B(x,y)| \geq B_{th} \\ 0, & \text{otherwise } |D_t(x,y) - B(x,y)| < B_{th} \end{cases}$$

4: Determine Blobs on $F_t(x, y)$ using connected comp. ▷ Fig. 2d
5: Assign to person image $P_t(x, y)$ the Blob with the greatest similarity to $P_{t-1}(x, y)$
6: Determine number of Blobs N_b
7: **If** $\frac{area(P_t(x,y))}{area(P_{t-1}(x,y)} > T_a$ or $N_b > 1$
8:     Determine

$$F_{t-3}(x,y) = \begin{cases} D_{t-3}(x,y), & \text{if } |D_{t-3}(x,y) - B(x,y)| \geq B_{th} \\ 0, & \text{otherwise } |D_{t-3}(x,y) - B(x,y)| < B_{th} \end{cases}$$

9:     Determine ROI, allocate stack S, determine seed on the basis of F_{t-3}, allocate logical table L and initialize it with false
10:     $P_t(x, y)$ = RegionGrowing($D_t(x, y)$, seed)
11:     $D'_t(ROI) = D_t(ROI) - P_t(ROI)$ ▷ Fig. 2f
12:     Push $D'_t(ROI)$ on stack S
13:     $L = L$ or logical($D'_t(ROI)$) ▷ Fig. 2g
14:     **If** for each (x, y), $L(x, y) \neq$ false
15:         $B'(x, y)$ = Median(S)
16:         return $B'(x, y)$
17:     **Else**
18:         acquire new depth map
19:         determine seed
20:         go to line #9
21: **Else**
22:     $B'(x, y) = B(x, y)$
23: return $B'(x, y)$ ▷ Fig. 2h

A modification of the scene layout requires an update of the depth background map. Figure 2 illustrates a dynamic scene, where a person closes the door. In such a scenario the depth model should accommodate to scene changes. In our approach the depth background map is a temporal median over a set of the depth maps. The depth background map is updated only in a region of interest ROI, which is represented by a rectangular sub-image. It contains the foreground objects, see Fig. 2d. After determining the foreground image $F_{t-3}(x, y)$ the algorithm uses it to determine a seed region, which is used by a region growing procedure, see 10th line in Algorithm 1. Then, in region constrained by ROI, the person blob extracted by the region growing is removed from the current depth map, see 11th line in Algorithm 1 and Fig. 2f. The image $D'(x, y)$ extracted in such a way is then pushed on a stack S, which holds a set of depth maps required for determining the temporal median. The temporal median is calculated if all pixels in the logical table L are true. This means that all pixels in the ROI region changed the value from false to true, i.e. that at every (x, y) location, at least one background pixel D' has been stored in every location (x, y) in S, which in turn is employed in the median filtering. After updating the depth background map, see 15th line in Algorithm 1 as well as Fig. 2h, the algorithm returns the background model. By comparing depth background map before the scene change, see Fig. 2b, and depth background map, which has been extracted after the scene change, see Fig. 2h, we can notice that the model accommodated to change of the scene. In particular, it contains only objects belonging to the room. As we can notice, the depth background model takes into account the closed door. The thresholds T_a and B_{th} were determined experimentally. The buffer Q contains three last consecutive depth maps $D_{t-1}, D_{t-2}, D_{t-3}$, i.e. Q_{size} is set to 3. In the current implementation it contains also depth maps for the calculation of dynamical features as well as maps for re-initialization of the background model. The region growing function is discussed below.

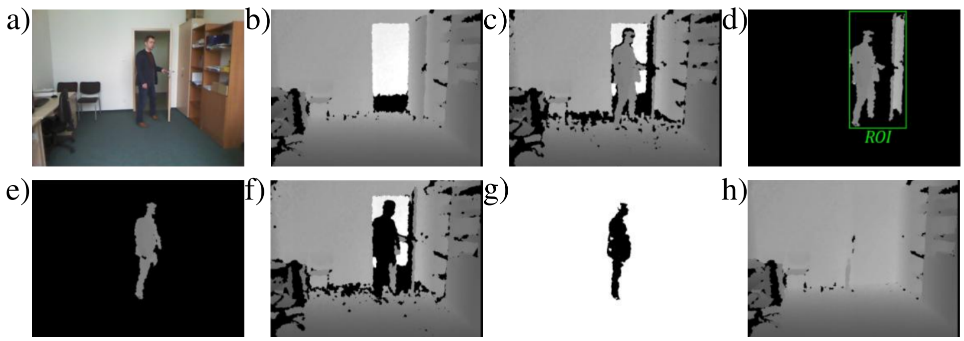

Fig. 2: Person extraction in a dynamic scene. RGB input image a), initial depth background model b), input depth map c), foreground blobs d), segmented person e), input depth map after removing person f), logical table L g), updated depth background model of the scene h)

4.2. Person Detection in Depth Maps from Ceiling-mounted Active Camera

The Kinect sensor has an angular field of view of 43° vertically and 57° horizontally. The observation area of an overhead Kinect mounted at the altitude of 2.6 m from the floor is about 5.5 m^2. In order to increase the field of observation a home-made pan-tilt head has been utilized to rotate the Kinect sensor. Thanks to the use of such a pan-tilt motorized head the observation area covered by the device is far larger and in effect the Kinect can cover a typical room, say 15-20 m^2 [41]. When a person moves, a PI controller rotates the camera in order to keep him/her in the central part of the image. The person is detected in real-time on the basis of a depth region growing. The person's position is represented by the centroid of the delineated blob. In order to decrease the number of pixels that can be potentially included into the person blob, in advance, the algorithm detects the floor using RANSAC algorithm [42].

The seeded region growing [43] was originally designed for intensity, i.e. gray-value images. The method takes a set of seeds as starting points along with the image. The regions are then grown from these seed pixels to pixel neighbors depending on a region membership criterion, which determines whether the adjacent point should join a region or not. The magnitude of the difference δ between pixel's intensity value and the region's mean is used as a decision criterion. The order in which the pixels are processed is determined by a global priority queue, which orders all candidate pixels by their fitness scores. This way the pixel with the smallest measured difference is assigned to the respective region. If the unlabeled pixel meets two or more boundary pixels from adjacent regions, it is joined into a region that has the smallest similarity distance and then it is marked as border region. The above process continues until all pixels are assigned to a region.

A disadvantage of the original region growing is that it does not update the previous entries in the sequentially sorted list (SSL) to reflect new differences from a region whose mean has been updated. In an improved seeded region growing [44] the border pixels, which have the same minimum δ value are processed in parallel. The improved algorithm employs an ascending priority queue (PQ) and several LIFO queues, where each LIFO queue contains pixels with the same δ value. When a new pixel is added to the PQ, it is inserted into a LIFO queue that corresponds to the pixel's δ value. This means that instead of removing individual pixels from the PQ, the entire LIFO queue corresponding to the smallest δ value is removed.

In our approach, the person is extracted in the sequence of depth maps using a modified region growing, which starts from a single seed region. The pseudo-code of the algorithm is given in Algorithm 2, whereas the symbols utilized in the pseudo-code are explained below. An NHQ queue contains the pointers to pixels neighboring with the depth region, whereas a QM holds the NHQ queues indexed according to δ. The queue with the smallest δ is denoted by FQ.

Table 1 Notation in the Algorithm 2

RG_M	The mean value of the depth region
δ_{th}	Threshold for δ value
NHQ	Neighbours holding queue
QM	Map of queues holding δ
FQ	Queue with smallest δ
Labels	
NO_L	Not visited pixel
IN_NQ	Pixel is inserted in NHQ queue
IN_Q	Pixel is in QM queue
LA	Pixel is assigned to the region

At the beginning, the seed pixels are assigned the LA label, the mean depth value of the pixels belonging to seed is determined and the neighboring pixels are inserted in the neighbors holding queue (NHQ). Then, the algorithm iterates until NHQ and QM are nonempty. At the beginning of the iterative process the algorithm iterates until the NHQ queue is not empty, see line #3. In the discussed loop the algorithm deletes the pixel from the NHQ queue and then calculates its δ. If δ is smaller or equal a threshold δ_{th} the algorithm inserts the pixel in the QM's queue indexed by the δ value and assigns him the IN_Q label, otherwise it takes next pixel from NHQ queue. After terminating the loop, the algorithm examines if the QM queue is not empty, see 10th line in pseudo-code. If yes, it deletes from QM the queue FQ with the smallest δ and then iterates until the FQ is nonempty. At each step in the loop, the algorithm deletes the pixel from the FQ queue, see 13th line, assigns him the LA label, and it assigns the neighboring pixels with NO_L labels to NHQ queue as well as changes their label from NO_L to IN_NQ label. After terminating the loop, the algorithm actualizes the average value of depth. It continues until NHQ and QM are nonempty.

Figure 3 demonstrates the extracted person blob by the discussed algorithm together with the images illustrating the region growing stages. Due to the nature of the distribution of depth values on the person's head, i.e. gradually decreasing depth values, the algorithm at first extracts the head. This means, that if the seed region is located in the head's area, the algorithm will extract the head first, see Fig. 3b, then the arms, see Fig. 3c, and then the remaining body. As we can notice, this

Algorithm 2 Depth region growing

Assign LA to seed pixels, calculate RG_M, insert the neighboring pixels in NHQ and assign them IN_NQ
While NHQ $\neq \emptyset$ and QM $\neq \emptyset$
    While NHQ $\neq \emptyset$
        Delete pixel from NHQ
        Calculate its δ
        If $\delta > \delta_{th}$
            continue
        Insert it in QM queue with index δ
        Assign him IN_Q label
    If QM $\neq \emptyset$
        Delete FQ from QM
        While FQ $\neq \emptyset$
            Delete pixel from FQ
            Assign him LA label
            Assign its neighbors with NO_L to NHQ
            and assign them IN_NQ label
    Actualize RG_M about the pixel values from FQ

creates a possibility for analysis of shapes, which arise during region growing to authenticate that the extracted regions belong to the person's area. In particular, the oval shape of the head can be approximated by an ellipse, see Fig. 3b, whereas the head-arm part can be approximated by an ellipse or T-shape like figure, depending on the relative position between person and the camera, see Fig. 3c.

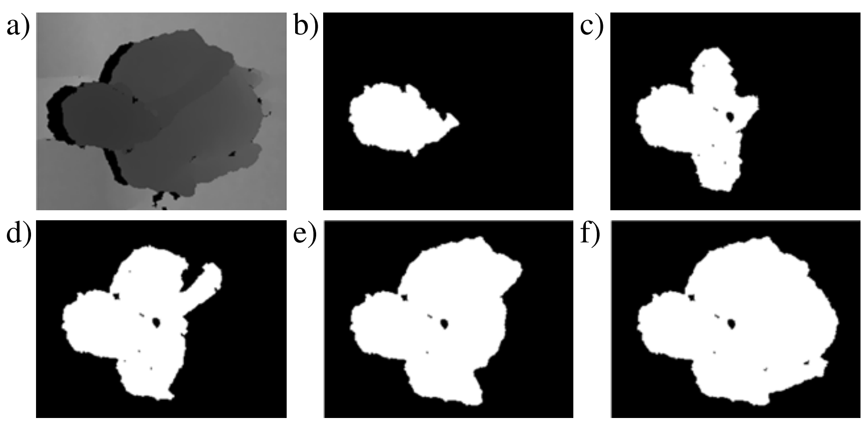

Fig. 3: Person extraction on depth maps acquired by a ceiling-mounted camera using region growing.

4.3. Finding person in depth maps

Regular depth region growing suffers from the effect of region chaining (overspill), which takes place when two separate regions are grown into single region while they are really split. To ameliorate delineation of the subject in such circumstances as well as to improve person following by the active camera, we can configure the system to execute a person's head finder. The detector can also be used for automatic initialization of person tracking. The head detection is realized by a linear SVM, which operates on Histogram of Oriented Depths (HOD) features [45]. The HOD features locally describe the orientation of depth changes. In this work, they are determined in sub-windows of fixed size [46]. The scaling ratio is determined on the basis of the distance between the camera and the head's part, which is closest to the camera. The sub-windows of fixed size undergo a subdivision into cells. The descriptors are extracted for each cell and finally the oriented depth gradients are assembled into 1D histograms.

5. Feature extraction

First part of this Section is devoted to explain how we indicate fall events on the basis of motion data. Subsequently, we present recognition of lying pose in depth maps, which are acquired by the frontal as well as the ceiling-mounted depth sensor. Afterwards, we discuss features that describe dynamic transitions of the body. Finally, we explain how person falls are detected.

5.1. Fall indicating using body-worn accelerometer

A lot of various techniques were proposed to achieve reliable fall detection using IMUs [25]. Usually, the accelerometer-based techniques indicate the alarm if the acceleration reaches a certain threshold value. An algorithm proposed in [47] relies on change in body orientation. It raises alarm if the square root of sum of the squares of acceleration components exceeds a preset threshold value.

In the discussed system, a fall impact is signaled if the Total Sum Vector SV_{total} is larger than 2.5 g. The SV_{total} value is determined as follows:

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

where $A_x(t)$, $A_y(t)$, $A_z(t)$ stand for the acceleration in the $x-$, $y-$, and $z-$axes at time t, respectively. The SV_{total} value includes both the dynamic and static acceleration components. It equals to $9.81 m/s^2$ when the accelerometer has no acceleration, and zero when it is in free fall. Having on regard that the potential fall is indicated if the SV_{total} is larger than experimentally determined threshold, the fall event is signaled only on the basis of body impact, i.e. we do not consider free-fall and post-fall phases. Such phases were considered in approaches aiming at detecting the falls on the basis of acceleration data only [25, 18]. In order to measure the movements of the whole body the device was located around the pelvis, which is close to the center of the body mass. Thus, similarly to [18], the device was placed near the spine on the lower back. As pointed out in [48], since the acceleration signal measured from wrist varies considerably, the signal of ADL samples strongly overlaps with that of the fall events. An analysis [25] of acceleration signals from 240 falls demonstrated that a fall with the smallest trunk magnitude produced a value of 3.5 g. The mentioned above value provided 100% fall-detection accuracy. In [48], the SV_{total} value from waist was set 2.0 g, i.e. to smaller value in comparison to the value used in [25]. As noted, this difference might be partly explained by the median filtering, which changes the absolute peak value of the impact signal.

5.2. Fall detection dataset

The classifiers responsible for fall detection were trained on depth map sequences from URFD dataset[1]. The URFD dataset consists of depth map sequences acquired by two Kinect sensors with the corresponding motion data, which were collected by a body-worn accelerometer. The motion data consisting of the acceleration over time in the x, y, and z axes were acquired by a x-IMU device with sampling rate of 256 Hz. The frontal depth maps with the corresponding RGB images were acquired by a fixed Kinect that was placed at 1 m altitude from the floor, whereas the top view RGB-D maps were acquired by a second static Kinect, which has been mounted at a ceiling at the height of 3 m from the floor. All depth maps are synchronized with the motion data. The dataset contains thirty image/acceleration sequences with 30 falls, which were simulated by five persons, including one 50+ performer. They simulated falls from standing and from sitting on the chair. A part of dataset with frontal images contains also forty image/acceleration sequences that contain typical ADLs like sitting down, picking-up an object from the floor, crouching down, as well as ten data sequences with fall-like activities, consisting in quick lying on the floor and lying on the bed/couch. The sequences with falls consists of 3K images with corresponding motion exemplars, whereas the total number of images in ADLs sequences is equal to 10K.

5.3. Recognition of lying pose

On the basis of features representing the extracted person in the depth maps we trained classifiers responsible for distinguishing falls from ADLs. The lying pose has been distinguished from ADLs using classifiers trained on features representing the extracted person in the depth maps. For each of the considered camera setting a separate lying pose classifier has been prepared. For training and evaluating the classifiers we selected 2425 and 525 depth maps from URFD dataset. Such image set was then employed to build k-NN classifiers and to train linear SVM classifiers, whose main task was to check whether the person is lying on the floor. Below we discuss the features that were utilized in both camera settings.

5.3.1. Recognition of lying pose in depth maps from facing camera: The lying pose in frontal depth images has been recognized using both depth features and features expressed as a point cloud. The conversion from depth data expressed in the 2D array to data expressed as the point cloud has been done using factory calibrated settings. The following features were extracted from the frontal depth maps to recognize the lying pose:

- H/W - ratio of height to width of the person's bounding box
- T/T_{max} - ratio of height of the person's surrounding box 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.
- P_{40} - ratio of the number of points belonging to the person, contained in a surrounding cuboid of height 40 cm from the floor, with respect to total number of points belonging to the person.

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

5.3.2. Lying pose recognition in depth maps from ceiling-mounted camera: The detection of the lying pose in the maps from ceiling-mounted active camera has been realized on the basis of the following features:

- H/H_{max} - ratio expressing the head-floor distance to the person's height
- $area$ - ratio of the person's area in the depth map with respect to the area of the top-view blob of the person in the standing pose
- l/w - ratio expressing the major length to major width of person's blob in the depth image.

The major length and width (eigenvalues) have been calculated as follows [49]:

$$\begin{aligned} l &= 0.707\sqrt{(a+c) + \sqrt{b^2 + (a-c)^2}} \\ w &= 0.707\sqrt{(a+c) - \sqrt{b^2 + (a-c)^2}} \end{aligned} \tag{2}$$

$$\begin{aligned} a = \frac{M_{20}}{M_{00}} - x_c^2,\ \ b = 2(\frac{M_{11}}{M_{00}} - x_c y_c),\ \ c = \frac{M_{02}}{M_{00}} - y_c^2, \\ M_{00} = \sum_x \sum_y F(x,y),\ \ M_{11} = \sum_x \sum_y xyF(x,y), \\ M_{20} = \sum_x \sum_y x^2 F(x,y),\ \ M_{02} = \sum_x \sum_y y^2 F(x,y) \end{aligned}$$

where $F(x, y)$ indicates a pixel on binary image representing the extracted person, whereas x, y stand for image coordinates.

5.4. Dynamic transitions

During human fall the head-floor distance changes rapidly due body transition from vertical orientation to horizontal one. The distance between the person's centroid and the floor also varies considerably and quickly over the accidental fall. The ratio of the areas occupied by the person in the depth maps from a ceiling-mounted camera also changes meaningfully over the accidental fall. Therefore, by analyzing the above mentioned cues we can settle whether the body transition is intentional or not.

In the setting with ceiling-mounted sensor, aside from the features discussed in previous subsection, we employed also a dynamical feature incorporating information about the speed of the falling person's body towards the floor. The speed of the falling body was modeled through the distance between the farthest person points and the floor. The discussed feature was determined in the following manner:

$$h(t) = \frac{H(t)}{H(t-\Delta T)} \tag{3}$$

where value of $H(t)$ is determined at the time of the impact, and $H(t-\Delta T)$ is calculated ΔT prior the impact. The value of ΔT has been chosen experimentally and it was set to 600 ms. It is worth noting that typical lead times for falls range from 650 - 800 ms. The experimental results showed that in the scenario with the ceiling-mounted depth sensor such a feature quite reliably describes the fall dynamics. In the depth maps from a ceiling-mounted sensor the peak value of $H(t)/H(t-\Delta T)$ for the fall assumes values smaller than one. The accelerometer used as an indicator of the potential simplifies determining this ratio since the impact time t can be determined readily at low computational cost.

5.5. Fall detection

In the setting with the frontal camera the decision about the fall is taken on the basis of SV_{total} value (1) as well as lying pose detector. If the SV_{total} exceeds the value equal to 2.5 g the lying pose detector is executed to authenticate the fall. This means that the accelerometer filters most of the fall-like activities and as a result the lying pose detector is executed only in the case of high likelihood of fall event. In the scenario with the overhead camera, aside from the mentioned above cascade of two classifiers, the dynamical transitions are considered too. If both the accelerometer-based classifier signals high likelihood of fall event and the lying pose classifier indicates that the person is lying on the floor, then the threshold-based classifier is executed to examine if the value of dynamic transition (3) of the head assumes a value smaller than 0.6. In consequence, the final decision about the fall in the setting with the ceiling-mounted camera is taken on the basis of a chain consisting of three classifiers. The discussed chain includes a classifier that on the basis of acceleration data signals potential fall, as well as classifiers of lying pose and dynamic transition, which decisions are taken on the basis of depth maps. The final classifier of the fall considers different modalities and has lower false positive rate and high precision.

6. Real-Time Data Acquisition and Processing. Tuning and Parameters

Having on regard that fall detection system should be inexpensive as well as work anytime and consume low power, we designed an event-driven processing framework in which the body-worn accelerometer is utilized to signal high likelihood of the fall occurrence and depth maps are not analyzed frame-by-frame, but instead they are stored in a circular frame buffer. In case of high likelihood of fall event the previously stored depth maps are fetched from the circular buffer and then processed to extract the features. In the setting with the frontal and fixed camera the frame-by-frame calculations comprise person extraction through taking absolute value of difference between current depth map and depth background map, determining the connected components as well as calculating the areas and number of the connected components, see lines 1-6 in Algorithm 1. These operations do not require a significant computational power. In the setting with the active camera the person is extracted in every frame using the region growing, which is optionally followed by the person-finder. The extraction of the features is more time consuming due to extraction of the floor and determining the point clouds. However, owing to event-driven data processing such computations are realized only in case of high likelihood of fall. The real-time data processing can be realized on a PC running Linux. The algorithms were also executed on PandaBoard in order to demonstrate their application potential and possibility of running on low-cost computing boards.

The fall detection application uses an asynchronous message-driven communication model to propagate information throughout four application layers. It runs five main concurrent processes communicating via message queues, which are one of the interprocess communication mechanisms available under Linux and yield asynchronous communication among processes, see Fig. 4. In such a communication model a process usually referred to as the sender writes the generated messages to a queue, while one or more other receiver processes retrieve them from the queue. Once a message has been read, it is deleted by the kernel from the queue. This means that the sender and the recipient of the message do not have to cooperate with the queue at the same time. Even if several receivers are listening on a channel, each message can be retrieved by single process only. As we can see on Fig. 4, the first process is amenable for acquiring motion data from the wearable device, the second process acquires the maps from the depth sensor, third one extracts the person, fourth process is accountable for processing of data and feature extraction, whereas the fifth process is accountable for data classification and triggering the fall alarm. In case of use of the PandaBoard, the dual-core processor allows parallel execution of processing and acquisition processes. The IMU signals were collected with frequency of 256 Hz and 12-bit resolution.

The algorithms for person detection use parameters that were determined experimentally. The algorithm for person extraction using depth reference map requires the thresholds T_a and B_{th}, which were determined experimentally. The extraction performance does not drop significantly with the change of parameters mentioned above. The region growing is resistant to changes of experimentally determined δ_{th} values.

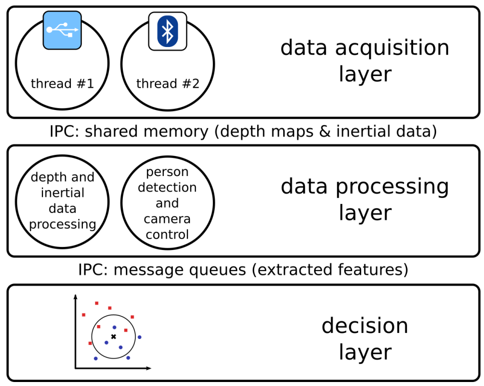

Fig. 4: Data acquisition, processing and communication between main processes.

7. Experimental results

In this Section we present the experimental results that were obtained on URFD dataset. In the subsequent subsections we discuss evaluation results that were obtained on publicly available URFD dataset using the presented fall detector along with performance of person detector and tracker.

7.1. Performance measures

The performance of the fall detector was evaluated with respect to sensitivity, specificity, accuracy and precision. Sensitivity and specificity were calculated as follows:

$$sensitivity = \frac{TP}{TP + FN} \times 100 \tag{4}$$

$$specificity = \frac{TN}{TN + FP} \times 100 \tag{5}$$

where TP stands for True Positives (number of detected falls), FN denotes False Negatives (number of undetected falls), FP indicates False Positives (number of ADL examples giving false fall alarms), whereas TN specifies True Negatives (number of ADL examples not giving fall alarms). A perfect fall detector should be described as 100% sensitive (e.g., all falls are identified as falls) and 100% specific (e.g., all ADLs are not identified as falls).

The accuracy is defined as the proportion of true responses (both True Positives and True Negatives) among the total number of cases examined. It measures how well the system predicts both categories. The precision is defined as the proportion of the true positives against all the positive results (both True Positives and False Positives). They were calculated as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (6)$$

$$precision = \frac{TP}{TP + FP} \times 100 \quad (7)$$

Accuracy expresses how close the indicated values are to the corresponding true values. When a method is precise, the amount of random variation is small.

7.2. Evaluation of the fall detector

The system can be configured to perform fall detection on the basis of depth data only or both accelerometric and depth data. In the second option, thanks to indicating that person's movement is above some preset threshold, the detection performance is far better. It is superior due to two stage decision process, where at first stage the fall-like activities are filtered out on the basis of acceleration data from body-worn device, and depth map analysis is conducted on a subset of frames, which likely contain fall event. Thanks to the use of accelerometer as an indicator of the potential fall the impact time can be determined easily and precisely enough, and thus the dynamic features (3) can be determined without considerable computational overheads.

7.2.1. Threshold Selection: Keeping in mind that the accelerometer is used only to signal potential falls, the acceleration threshold was set to 2.5 to indicate all fall and fall-like activities as non-ADLs. The experimental results described below were obtained using the X-IMU accelerometer that was worn near the spine on the lower back, and which was attached to body using an elastic belt around the waist.

7.2.2. Evaluation of the fall detector for facing camera: The lying pose detector on depth maps from the facing camera was evaluated on 2425 images from URFD dataset. The samples are selected so that the representative set included images that represent all the possible poses of lying person. We decided to base the system on the SVM and k-NN detectors since they demonstrated high fall detection performance. They were built on features discussed in subsection 5.3.1. The experimental results that were obtained by a linear SVM and a k-NN with five neighbors are shown in Tab. 2. Both classifiers gave identical results. The k-nn with three neighbors achieves slightly worse results. As we can notice the presented results are promising both in terms of sensitivity and specificity.

7.2.3. Evaluation of the fall detector for overhead camera: The algorithm for lying pose recognition in depth maps from the ceiling-mounted sensor has been evaluated on 875 representative images from URFD dataset. From the above mentioned dataset a subset of 60% images was chosen for the training, whereas the remaining images were used only in testing. Since the discussed image sequences were acquired by a static depth sensor, the person has been extracted by differencing the depth maps from the depth background map. The discrimination between falls and ADLs has been performed by a linear SVM and a k-NN with 5 neighbors. The discussed classifiers operated on features discussed in Section 5.3.2. The classification performances, which were obtained by the classifiers mentioned above are shown in Tab. 3. As we can observe, the results

Table 2: Performance of lying pose recognition on frontal depth maps of URFD dataset.

			True		
			Fall	No Fall	
Estimated	SVM, k-NN	Fall	898	6	Accuracy=99.55%
		No fall	5	1516	Precision=99.34%
			Sens.= 99.45%	Spec.= 99.61%	

Table 3: Performance of lying pose detection on overhead depth maps from URFD dataset [50].

			True		
			Fall	No Fall	
Estimated	SVM	Fall	244	9	Accuracy=97.52%
		No fall	4	268	Precision=96.44%
			Sens.= 98.39%	Spec.= 96.75%	
	k-NN	Fall	244	10	Accuracy=97.33%
		No fall	4	267	Precision=96.06%
			Sens.= 98.39%	Spec.= 96.39%	

obtained by lying pose detectors operating on features from ceiling-mounted sensor are promising in terms of both accuracy and sensitivity. We evaluated also k-nn classifier with three neighbors, which gave slightly worse results. The C parameter of the SVM classifier has been set to default value, i.e. to one. Experiments consisting in an exhaustive grid search over the parameter space to find the best setting demonstrated that the presented results do not change significantly for the C values differing from the default value.

Subsequently, we conducted evaluations in terms of the usefulness of the dynamic feature for distinguishing between fall and fall-like actions. They were evaluated in context of improving the distinguishability between the accidental falling and the intentional lying on the floor. Figure 5 depicts the receiver operating characteristic (ROC) curve of the dynamic feature. It illustrates the classification performance of a binary classifier for different values of the discrimination threshold. The best accuracy was achieved for ΔT equal to 500 ms and $\mathsf{Threshold}$ equal to 0.525.

Afterwards, we asked two volunteers to act as evaluators of the dynamic feature. It turned out that a simple cascade classifier consisting of the lying pose detector and the dynamic transition detector performs very well in practice as it has almost null false alarm ratio. In particular, the cascade gives promising results if a moment in which there was the impact is determined precisely. In consequence, the fall detector consisting of accelerometer-based fall indicator, depth map-based lying pose detector and dynamic transition detector achieves the best results for the overhead camera. It is worth noting that such a classifier detected properly all falls in the depth map sequences from URFD dataset. However, the discussed classifier was not able to completely eliminate the false alarms. The inability to eliminate false alarms by fall detectors working on images from a single camera motivated us to elaborate a fall detector using images both from facing and overhead cameras.

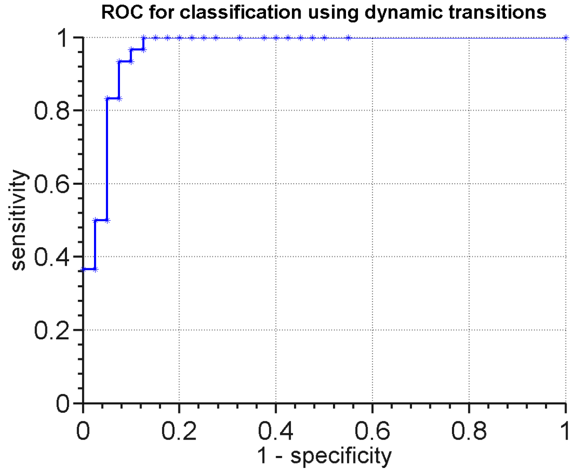

Fig. 5: ROC for dynamic feature.

7.2.4. Evaluation of the fall detector using data from facing and overhead cameras: Using the features extracted from both facing and overhead cameras we evaluated a fall detector consisting in accelerometer-based fall indicator and k-NN classifier with 5 neighbors for lying pose detection. The classifier operates on features discussed in Subsections 5.3.1 and 5.3.2. It has been trained on features extracted from the maps, which were used in Subsection 7.2.3, plus corresponding depth images from the facing sensor. The results are presented in Tab. 4. As we can observe, the results obtained in the discussed camera setting are superior in comparison to results presented previously.

7.2.5. Comparison of performance of fall detection: Table 5 summarizes the classification performances, which were obtained using the discussed camera settings. The camera setup with frontal camera gives slightly better results in comparison to setup with the top camera. The best results were obtained using data from both cameras.

7.3. Evaluation of person detector and tracker

If the system is configured to work with a static ceiling-mounted camera, the person can be extracted at low computational cost by differencing the current depth map from continuously updated depth reference map of the scene. On the PandaBoard this operation takes about 10 ms. However, as we already mentioned, the ceiling-mounted and fixed Kinect has quite limited observation area.

Table 4: Performance of lying pose recognition on frontal and overhead depth maps of URFD dataset [%].

			True		
			Fall	No Fall	
Estimated	k-NN	Fall	903	0	Accuracy=100.00%
		No fall	0	1192	Precision=100.0%
			Sens.= 100.00%	Spec.= 100.00%	

Table 5 Performance of lying pose detection [%].

	front. cam.	top cam.	front. + top cam.
Accuracy	99.55	97.52	100.0
Precision	99.34	96.44	100.0
Sens. of fall det.	99.45	98.39	100.0
Spec. of fall det.	99.61	96.75	100.0

By the use of a ceiling-mounted motorized head to rotate the Kinect, the observation area can be expanded noticeably. In such a setup with pan-tilt sensor, more sophisticated and time consuming algorithms are required to extract the subject and to follow it by the active camera.

We began with evaluation of the depth region growing in the depth maps from the URFD dataset. It is worth noting that in all sequences the person was extracted correctly. This means that the presented person tracker achieved perfect performance on all fall events registered in URFD. Next, our region growing was examined on five depth map sequences that were acquired by the pan-tilt camera. The subject was moving freely around the room in an area that could not be observed by the fixed camera. In the discussed experiments, it was required not only to extract the person in real-time, but additionally to keep he/she in the central part of the depth maps acquired by the active camera. Before starting the delineation of person, the camera was stationary for a while to initially extract the subject through differencing the current depth map from the depth reference map of the scene. It is worth mentioning that in all frames acquired by the active camera, including depth maps containing intentional falls, every main body part has been extracted properly. On Fig. 6 there are presented illustrative results, which were achieved on depth images colleted by the active camera. On the PandaBoard the time needed for region growing-based person delineation depends on the blob size and ranges from 17 ms to about 25 ms. A video illustrating the person tracking by pan-tilt depth camera is available under the following link: `http://fenix.univ.rzeszow.pl/~mkepski/demo/act.mp4`.

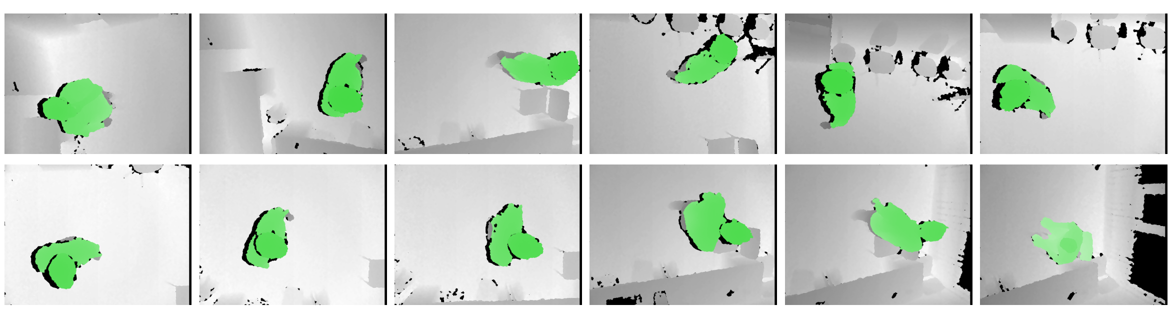

Fig. 6: Region growing – based person detection and tracking on depth maps acquired by a ceiling-mounted active camera.

The person detector, which is discussed in subsection 4.3 has been evaluated on 254 positive examples and 638 negative examples of which 60% were used for training and the remaining 40% for testing. The depth maps with the delineated person were rotated to a canonical pose on the basis of axis of the person's blob. They were also scaled according to the distance of his/her head to the camera. Table 6 contains experimental results that were obtained using the HOD-SVM detector, see also [51][46]. As we can observe, the detector achieves better performance when the rotation of silhouettes to the canonical pose takes place. On the other hand, the improvement in the performance is not considerable, and this in turn indicates that the algorithm is fairly resistant to variations in head poses. This is due to extracting of HOD features on gradients forming elliptical like structures on the person's head seen on depth maps from an overhead camera. The discussed results were achieved using the HOD with the cell size equal to 8×8. The time needed for person detection on the PandaBoard is equal to 41 ms.

Table 6 Performance of person detection using HOD-SVM on depth maps from ceiling-mounted active camera [%].

	accuracy	precision	sensitiv.	specificity
rotat.	99.45	98.21	100.0	99.22
no rotat.	98.91	98.18	98.18	99.22

8. Discussion

Both camera settings have advantages and disadvantages. As we already mentioned, ceiling-mounted depth sensors are rarely utilized in fall detection research [37][8]. One of the reasons for this is limited monitoring area of depth sensors, including Kinect. Our experimental results demonstrate that owing to mounting the depth sensor on a motorized pan-tilt unit the observation field can be extended considerably. As a result, typical senior rooms of size up to 25m^2 can be monitored by the single depth sensor. Moreover, we found that person delineation in such a setting with an active camera can be done quite reliably and fast. Our modified depth-region growing for person extraction demonstrated value in several experiments. The person detection times on the low-cost PandaBoard are close to times needed for real-time processing. The computational power of current PCs is sufficient to execute in real-time our algorithms for person detection and fall recognition. One of the most significant obstacles to the introduction of fall detection systems and their acceptance by seniors is the barrier of costs of devices and their everyday use [8]. In this context it is worth noting that our system for fall detection can be built relatively inexpensively, using a low-cost depth camera, a wireless accelerometer and low-cost processing boards like PandaBoard.

One of the advantages of the setting with the active ceiling-mounted camera is that the number of situations in which occlusions impede person extraction is much smaller in comparison to setup with a facing camera. In this context it is worth noting that there is almost no significant work that deals with fall detection in case of visual occlusions [13][8]. Another observation is that for such a camera setting a similar performance of lying pose detection can be obtained with smaller number yet more discriminative features in comparison to depth map features that are needed for reliable fall detection using a facing camera. Moreover, in setup with the ceiling-mounted camera the dynamical features, which demonstrated high discrimination power, can be computed quite easy, particularly if a body-worn accelerometer is used. We showed experimentally that two-camera system achieves perfect detection performance on data from freely available URFD dataset.

As we already mentioned, our work differs from relevant work since we focus on ceiling-mounted pan-tilt depth sensor, and last but not least, in that we are using a body-worn accelerometer to indicate the context of the event. In this way, an expert knowledge about the specificity of the fall detection problem has been realized in the form of event-driven architecture and a cascade of classifiers. In consequence, a decision about the fall is not taken by a single classifier, trained using a machine learning technique, but it is taken on the basis of carefully designed and evaluated classifiers, which attempt to mimic human experts. Our conclusions are in line with the research findings presented in [27][18] in that the motion patterns of real-falls might differ from simulated falls, particularly if falling person is trying to save himself from falling in order minimize the effects of the fall. The event-based approach to fall detection does not introduce unobtrusiveness due to possible use of suit-integrated accelerometers, which employ the human body motions to continuously recharge the battery [52].

9. Conclusions

In this work, efficient algorithms for fall detection were developed, implemented and tested using depth map sequences and wireless inertial sensor worn by a monitored person. A set of descriptors for depth maps has been proposed to permit classification of person poses as well as his/her actions. The experimental validation was carried out on prepared and then shared data repository consisting of synchronized depth and accelerometric data. Extensive experiments and tests were conducted in the scenario with a static camera facing the scene and an active camera observing the scene from above. The algorithms were designed with regard to low computational demands and possibility of their run on ARM platforms. Several experiments consisting in person detection, tracking and fall detection in real-time were carried out to show efficiency and reliability of the proposed solutions. Both camera settings were compared in terms of person detection and fall recognition. The experimental results showed that the developed algorithms for fall detection have both high sensitivity and specificity. In future work, we will investigate scenarios with two persons as well as scenarios with occlusions. The region growing-based person detection should be extended to deal with activities like sleeping on a coach.

10. Acknowledgment

This work was supported by Polish National Science Center (NCN) under a research grant 2014/15/B/ST6/02808.

11. References

[1] P. Rashidi and A. Mihailidis, "A survey on ambient-assisted living tools for older adults," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 579–590, May 2013.

[2] M. Friedewald and O. Raabe, "Ubiquitous computing: An overview of technology impacts," *Telemat. Inf.*, vol. 28, no. 2, pp. 55–65, May 2011.

[3] G. Acampora, D. Cook, P. Rashidi, and A. Vasilakos, "A survey on ambient intelligence in healthcare," *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2470–2494, Dec 2013.

[4] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *BioMedical Engineering OnLine*, vol. 12, 2013.

[5] J. Murphy and B. Isaacs, "The post-fall syndrome: a study of 36 elderly patients," *Gerontology*, vol. 28, pp. 265–270, 1982.

[6] 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, 1993.

[7] N. Noury, P. Rumeau, A. Bourke, G. Laighin, and J. Lundy, "A proposal for the classification and evaluation of fall detectors," *IRBM*, vol. 29, no. 6, pp. 340 – 349, 2008.

[8] J. Hamm, A. G. Money, A. Atwal, and I. Paraskevopoulos, "Fall prevention intervention technologies: A conceptual framework and survey of the state of the art," *J. of Biomedical Informatics*, vol. 59, pp. 319 – 345, 2016.

[9] D. Litvak, Y. Zigel, and I. Gannot, "Fall detection of elderly through floor vibrations and sound," in *Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, 2008, pp. 4632–4635.

[10] Z. Zhang, U. Kapoor, M. Narayanan, N. Lovell, and S. Redmond, "Design of an unobtrusive wireless sensor network for nighttime falls detection," in *Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, Aug 2011, pp. 5275–5278.

[11] R. Z. Morawski, Y. Yashchyshyn, M. Pirek, F. F. Jacobsen, K. Ovsthus, and W. Winiecki, "Monitoring of human movements by means of impulse-radar sensors," *Telecommunication Rewiew & Telecommuniacation News*, vol. 6, pp. 598–602, 2015.

[12] M. Yu, S. M. Naqvi, A. Rhuma, and J. Chambers, "One class boundary method classifiers for application in a video-based fall detection system," *IET Computer Vision*, vol. 6, no. 2, pp. 90–100, 2012.

[13] Z. Zhang, C. Conly, and V. Athitsos, "A survey on vision-based fall detection," in *Proc. of the 8th ACM Int. Conf. on PErvasive Technologies Related to Assistive Environments*. ACM, 2015, pp. 1–7.

[14] S. Sathyanarayana, K. R. Satzoda, S. Sathyanarayana, and S. Thambipillai, "Vision-based patient monitoring: a comprehensive review of algorithms and technologies," *J. of Ambient Intelligence and Humanized Computing*, pp. 1–27, 2015.

[15] Y. Wang and J. Winters, "An event-driven dynamic recurrent neuro-fuzzy system for adaptive prognosis in rehabilitation," in *Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, 2003, pp. II:1256–1259.

[16] F. Rea, G. Metta, and C. Bartolozzi, "Event-driven visual attention for the humanoid robot icub," *Frontiers in Neuroscience*, vol. 7:234, 2013.

[17] J. Perez-Carrasco, B. Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, S. Chen, and B. Linares-Barranco, "Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 35, no. 11, pp. 2706–2719, 2013.

[18] F. Bagala, 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(5), e37062, 2012.

[19] M. Kangas, R. Korpelainen, I. Vikman, L. Nyberg, and T. Jamsa, "Sensitivity and false alarm rate of a fall sensor in long-term fall detection in the elderly," *Gerontology*, vol. 61, no. 1, pp. 61 – 68, 2015.

[20] C. Chen, R. Jafari, and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, pp. 1–21, 2015.

[21] F. Hoflinger, J. Muller, R. Zhang, L. Reindl, and W. Burgard, "A wireless micro inertial measurement unit (IMU)," *IEEE Trans. on Instrumentation and Measurement*, vol. 62, no. 9, pp. 2583–2595, Sept 2013.

[22] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jamsa, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait & Posture*, vol. 28, no. 2, pp. 285 – 291, 2008.

[23] Q. Li, J. Stankovic, M. Hanson, A. Barth, J. Lach, and G. Zhou, "Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information," in *Sixth Int. Workshop on Wearable and Implantable Body Sensor Networks*, June 2009, p. 138–143.

[24] J. Jacob, T. Nguyen, D. Lie, S. Zupancic, J. Bishara, A. Dentino, and R. Banister, "A fall detection study on the sensors placement location and a rule-based multi-thresholds algorithm using both accelerometer and gyroscopes," in *IEEE Int. Conf. on Fuzzy Syst.*, 2011, pp. 666–671.

[25] A. Bourke, J. O'Brien, and G. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait & Posture*, vol. 26, no. 2, pp. 194–199, 2007.

[26] M. Lustrek, H. Gjoreski, S. Kozina, B. Cvetkovic, V. Mirchevska, and M. Gams, "Detecting Falls with Location Sensors and Accelerometers," in *AAAI Conf. on Artificial Intell.* AAAI Press, 2011, pp. 1662–1667.

[27] M. Kangas, I. Vikman, L. Nyberg, R. Korpelainen, J. Lindblom, and T. Jamsa, "Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects," *Gait & Posture*, vol. 35, no. 3, pp. 500 – 505, 2012.

[28] M. Kepski and B. Kwolek, "Fall detection on embedded platform using Kinect and wireless accelerometer," in *Int. Conf. on Computers Helping People with Special Needs*. Springer-Verlag, 2012, pp. II:407–414.

[29] C. Chen, R. Jafari, and N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors," *IEEE Trans. on Human-Machine Systems*, vol. 45, no. 1, pp. 51–61, Feb 2015.

[30] L. Chen, H. Wei, and J. Ferryman, "A survey of human motion analysis using depth imagery," *Pattern Recogn. Lett.*, vol. 34, no. 15, pp. 1995–2006, Nov. 2013.

[31] M. Yu, A. Rhuma, S. Naqvi, L. Wang, and J. Chambers, "A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment," *IEEE Trans. on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1274–1286, Nov 2012.

[32] C. Wu and H. Aghajan, "Real-time human pose estimation: A case study in algorithm design for smart camera networks," *Proc. of the IEEE*, vol. 96, no. 10, pp. 1715–1732, Oct 2008.

[33] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "3D head tracking for fall detection using a single calibrated camera," *Image Vision Comput.*, vol. 31, no. 3, pp. 246–254, 2013.

[34] L. A. B. E. Demiroz, A. A. Salah, "Coupling fall detection and tracking in omnidirectional cameras," in *LNCS*, vol. 8749, 2014, pp. 73–85.

[35] M. V. Sokolova, J. Serrano-Cuerda, J. C. Castillo, and A. Fernández-Caballero, "A fuzzy model for human fall detection in infrared video," *J. of Intelligent & Fuzzy Systems*, vol. 24, no. 2, pp. 215–228, 2013.

[36] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier, "Fall detection from depth map video sequences," in *LNCS*, vol. 6719. Springer Berlin Heidelberg, 2011, pp. 121–128.

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

[38] K. Cuppens, C.-W. Chen, K. Wong, A. Van de Vel, L. Lagae, B. Ceulemans, T. Tuytelaars, S. Van Huffel, B. Vanrumste, and H. Aghajan, "Integrating video and accelerometer signals for nocturnal epileptic seizure detection," in *Int. Conf. on Multimodal Interaction*, 2012, pp. 161–164.

[39] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, pp. 116–124, 2013.

[40] B. Kwolek and M. Kepski, "Improving fall detection by the use of depth sensor and accelerometer," *Neurocomputing*, vol. 168, pp. 637 – 645, 2015.

[41] M. Kepski and B. Kwolek, "Fall detection using ceiling-mounted 3d depth camera," in *Int. Conf. on Computer Vision Theory and Appl. (VISAPP)*, pp. 2:640–647.

[42] R. Raguram, J.-M. Frahm, and M. Pollefeys, "A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus," in *Proc. of the 10th European Conf. on Computer Vision: Part II*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 500–513.

[43] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, Jun. 1994.

[44] A. Mehnert and P. Jackway, "An improved seeded region growing algorithm," *Pattern Recogn. Lett.*, vol. 18, no. 10, pp. 1065–1071, 1997.

[45] L. Spinello and K. Arras, "People detection in RGB-D data," in *IEEE/RSJ Int. Conf. on IROS*, Sept 2011, pp. 3838–3843.

[46] M. Kepski and B. Kwolek, "Fall detection using body-worn accelerometer and depth maps acquired by active camera," in *Proc. 11th Int. Conf. on Hybrid Artificial Intelligent Systems*. Springer, 2016, pp. 414–426.

[47] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable sensors for reliable fall detection," in *IEEE Int. Conf. on Engineering in Medicine and Biology Society*, 2005, pp. 3551–3554.

[48] M. Kangas, A. Konttila, I. Winblad, and T. Jamsa, "Determination of simple thresholds for accelerometry-based parameters for fall detection," in *29th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, 2007, pp. 1367–1370.

[49] B. Horn, *Robot Vision*. Cambridge, MA: The MIT Press, 1986.

[50] B. Kwolek and M. Kepski, “Fall detection using Kinect sensor and fall energy image,” in *Hybrid Artificial Intelligent Systems*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 8073, pp. 294–303.

[51] M. Kepski and B. Kwolek, “Embedded system for fall detection using body-worn accelerometer and depth sensor,” in *IEEE 8th Int. Conf. on Intell. Data Acquisition and Advanced Computing Systems*, vol. 2, 2015, pp. 755–759.

[52] S. Jung, S. Hong, J. Kim, S. Lee, T. Hyeon, M. Lee, and D.-H. Kim, “Wearable fall detector using integrated sensors and energy devices,” *Scientific Reports 5, Article number: 17081*, 2015. [Online]. Available: http://www.nature.com/articles/srep17081