Applied Soft Computing 40 (2016) 305–318

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Fuzzy inference-based fall detection using kinect and body-worn accelerometer

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

[a] AGH University of Science and Technology, 30 Mickiewicza Avenue, 30-059 Krakow, Poland
[b] University of Rzeszow, Pigonia 1, 35-310 Rzeszow, Poland

ARTICLE INFO

Article history:
Received 4 August 2015
Received in revised form 30 October 2015
Accepted 15 November 2015
Available online 8 December 2015

Keywords:
Human activity analysis
Fuzzy logic
Fall detection

ABSTRACT

In this paper, we present a new approach for reliable fall detection. The fuzzy system consists of two input Mamdani engines and a triggering alert Sugeno engine. The output of the first Mamdani engine is a fuzzy set, which assigns grades of membership to the possible values of dynamic transitions, whereas the output of the second one is another fuzzy set assigning membership grades to possible body poses. Since Mamdani engines perform fuzzy reasoning on disjoint subsets of the linguistic variables, the total number of the fuzzy rules needed for input–output mapping is far smaller. The person pose is determined on the basis of depth maps, whereas the pose transitions are inferred using both depth maps and the accelerations acquired by a body worn inertial sensor. In case of potential fall a threshold-based algorithm launches the fuzzy system to authenticate the fall event. Using the accelerometric data we determine the moment of the impact, which in turn helps us to calculate the pose transitions. To the best of our knowledge, this is a new application of fuzzy logic in a novel approach to modeling and reliable low cost detecting of falls.

1. Introduction

Human behavior understanding is becoming one of the most active and extensive research topics in artificial intelligence and cognitive sciences. Automatic activity recognition is a process the objective of which is to interpret the behavior of the observed entities in order to generate a description of the recognized events or to raise an alarm. The capture of data associated to these entities can be achieved by sensors such as cameras that collect images of a specific scene, or inertial sensors that measure physical quantities of the moving object regardless of illumination or scene clutter. One of the biggest challenges in decision making about the alarm on the basis of such sensor readings is to cope with uncertainty, complexity, unpredictability and ambiguity [1,2].

Traditional machine learning techniques pose some limitations in modeling human behavior due to the lack of any reference to the inherent uncertainty that human decision-making has. On the other hand, fuzzy logic poses the ability to imitate the human way of thinking to effectively utilize modes of reasoning that are rough rather than exact. With fuzzy logic we can indicate mapping rules in terms of linguistically understandable variables rather than numbers. Processing the words gives us the opportunity to express imprecision, uncertainty, partial truth and tolerance [3]. In consequence, fuzzy logic-based inference engines are capable of achieving robustness and close resemblance with human-like decision making in ambiguous situations.

Recognition and monitoring of activities of daily living (ADLs) is important ingredient of human behavior understanding [4–6]. Several approaches have been proposed to distinguish between activities of daily living and falls [7–10]. Falls are a major health risk and a significant obstacle to independent living of the seniors and therefore significant work has been devoted to ensure robustness of assistive devices [11]. However, regardless of a lot of efforts undertaken to obtain reliable and unobtrusive fall detection, current technology does not meet the seniors' needs. The main cause for not accepting of currently available technology by elderly is that the existing devices generate too much false alarms. In consequence, some ADLs are mistakenly indicated as falls, which in turn leads to substantial frustration of the users of such devices.

The most common method for fall detection consists in using a body-worn tri-axial accelerometer and proving whether acceleration's amplitude crosses a fixed threshold [12]. Typically, such algorithms distinguish poorly between activities of daily living and falls, and none of which is commonly accepted by elderly. The main reason of poor separability and high false ratio of devices using only accelerators is lack of adaptability together with

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

http://dx.doi.org/10.1016/j.asoc.2015.11.031
1568-4946/© 2015 Elsevier B.V. All rights reserved.

insufficient capabilities of context understanding. In order to improve the recognition performance the use of both accelerometer and gyroscope has been investigated by several research groups [13,14]. However, it is not easy to achieve low false alarm ratio since several ADLs like quickly lying down on a bed share similar kinematic motion patterns together with real falls. It is worth noting that despite of several shortcomings of the wearable devices for fall detection, the discussed technology has a great potential to provide support for seniors. It is also the only technology that was successfully used in large scale collection of motion patterns for research in the field of fall detection. Nowadays wearable devices like smart watches, which are frequently equipped with miniature inertial sensors, are unobtrusive and can deliver motion data during dressing up, bath as well as standing up the bed, i.e. during critical phases, in which considerable number of accidental falls and injuries take place.

Several methods have been developed so far to detect falls using various kinds of video cameras [15,2,16]. In general, video-based fall detection systems show some potential and reliability in detecting falls in public places. However, in home environments the RGB cameras are less useful since they do not preserve privacy. Moreover, video camera-based algorithms cannot extract the object of interest all time of the day, especially in dark rooms. As indicated in [2], such algorithms only work in normal illumination conditions, whereas the fall risk of adults is much larger in low lighting conditions. For that reason, in [15,2] in order to recognize different activities in various environments, both controlled as well as unstructured, an infrared illumination was utilized to enable the web cameras to deliver images of sufficient quality in poor lighting conditions. It is also worth noting that the currently available prototype devices require time for installation, camera calibration and they are not cheap since a considerable computational power is needed to execute in real-time the time consuming algorithms. While these techniques might give good results in several scenarios, in order to be practically applied they must be adapted to non-controlled environments in which neither the lighting nor the subject undergoing tracking is under full supervision. Additionally, the lack of depth information can lead to lots of false alarms.

Recently, Microsoft introduced the Kinect sensor, which delivers dense depth maps under difficult lighting conditions. This motion sensing device features an RGB camera and depth sensor, which consists of an infrared projector combined with a monochrome CMOS sensor capturing 3D data under any ambient light conditions with not direct natural illumination. The depth information is then utilized to estimate a skeletal model of any humans in Kinect's view using a Random Forest classifier, which assigns each pixel as being a body part or background [17]. The RGB camera data is not used for this due to its high variability in poor lighting conditions. Pixels corresponding to each body part are then clustered and fitted to a skeletal model on a frame-by-frame basis. The Kinect has strong potential in human behavior recognition in a wide range of illuminations, which occur during a typical 24-hour day-night cycle. Depth information is very important cue since the entities may not have consistent color and texture but they must occupy an integrated region in the 3D space. In particular, owing to this property the person can be reliably extracted at low computational cost.

In the area of fall detection, the Kinect sensor has been introduced quite recently [18,14,10]. In [18], an overall fall detection success rate of 98.7% has been obtained using the distance of gravity center to floor level and velocity of a moving body. In [14], we presented a method for fusing the features extracted on the depth maps with data from inertial sensors. A Takagi-Sugeno (TS) fuzzy inference system has been used to trigger a fall alarm using the information about person's motion and the distance of gravity center to the floor. The discussed methods are resistant to changes of light conditions since they utilize the center of the gravity, which is extracted on depth maps only. As demonstrated in recent work, fusion of depth camera and body-worn inertial sensors can improve recognition of human activities [19] as well as detection of nocturnal epileptic seizure [20]. However, as indicated in [21], although there already exist several methods to detect human activities based either on wearable sensors or on cameras, there is little work that is devoted to combining the two modalities. In [22], a two-stage system is used for fall detection. The first stage is responsible for characterizing the vertical state of a segmented 3D object for individual frames, and events through a temporal segmentation of the vertical state time series of the tracked 3D objects. In the second stage the system employs an ensemble of decision trees and the features extracted from an on ground event to acknowledge belief that a fall preceded it. Overall, the depth information is very advantageous in real-time tracking of human faces [23], since the head trajectories resulting from the tracking are very useful in behavior recognition, particularly in fall detection [24].

In our previous work [14], the acceleration data from the accelerometer, the angular velocity data from the gyroscope, and the center of gravity data of a moving person that was determined on the basis of Kinect depth maps were used as inputs in a fuzzy inference module to fuse information and then to generate alarms when falls occurred. The work described in here builds on that previous research by rebuilding the fuzzy engine and extending it about new modules. The system uses an accelerometer and depth data. Instead of using a single inference module, we utilize two input fuzzy inference engines, which are responsible for inferring separately the static poses and dynamic transitions, and which outputs are used by the output fuzzy engine, enhancing the confidence of the inference. In consequence, multiple independent measurements and the corresponding features are authenticated by the measurements of the another sensor as well as different features. This way a cooperative arrangement emerges and confidence of the fall alert is enhanced. This is because static actions such as lying on the floor are different from dynamic transitions, and in particular different measurements are needed to describe them. As a result, considerably smaller number of rules is required to describe how the Fuzzy Inference System (FIS) should make a decision regarding classifying the input data.

To overcome difficulties related to the high dimensional input spaces, an idea of hierarchical fuzzy systems has been proposed in [25]. Based on discussed linear hierarchical structure, in which the number of the rules increases linearly with the number of the input variables, it has been proven in [26] that hierarchical fuzzy systems are universal approximators. However, such hierarchical fuzzy structures can become universal approximators only when there is sufficient number of free parameters. In [27] it has been proven that universal approximation property can be obtained by increasing the number of hierarchical levels. Our approach differs from the above approaches since in the input level we utilize fuzzy reasoning on disjoint subsets of the linguistic variables, which express different modalities of the observations. In contrast to these hierarchical models as well as hierarchical structure of rules [28], our output engine does not operate on raw input variable(s), but it only operates on fuzzy sets and the membership grades inferred at the preceding level of knowledge extraction.

The next contribution of this paper is employing the information about the time of impact to extract very informative temporal features. Thanks to relatively high sampling rate of the accelerometer the system precisely determines the time instants characterizing the fall, which are then used to calculate depth map-based temporal features. We demonstrate experimentally that such features are very informative as well as that fusion and manipulation of linguistic variables and rules is easy. Our modular system reduces cost and has flexibility, increased system reliability and good scalability. The subsequent contribution is our fuzzy architecture with reduced power consumption. Owing to combination of crisp and fuzzy relations, i.e. alerts produced by the accelerometer in case of rapid motion, there is no need to perform fuzzy inference frame-by-frame. Instead, we collect the depth maps in a circular buffer and process them if there is evidence of a fall. Reduced power consumption was a key feature to be incorporated into the system design. The resulting easy-to-install fall detection system is unobtrusive and preserves privacy, and operates all time of the day.

The contribution of this work is a fuzzy inference system consisting of fuzzy inference subsystems, which are responsible for drawing conclusions about the static poses and dynamic transitions. The accelerometer filters at low computational cost slow motions and gives time stamps at which depth-based features describing rapid motion should be calculated. The proposed linguistically understandable classifiers can be generalized to other applications especially when sensor fusion is involved. The proposed method is general and can be used to fuse heterogeneous sensors.

Surprisingly from our findings, there is very minimal or almost no research that had tackled the problem of activity recognition on the basis of different modalities using the fuzzy approach. Fuzzy logic has been used in several systems for human fall detection. In [29], a fuzzy logic-based posture recognition method identifies four static postures with an accuracy of 74.29% on 62 video sequences. The algorithms can detect emergency situations such as a fall within a health smart home. An approach [30] uses fuzzy logic to generate on ground, in between, and upright state confidences from the body orientation and the spine height features. The confidences are then thresholded to trigger a fall alarm. With one false alarm the fall detection accuracy of 98.6% was reported on a set of 40 falls and 32 non-falls collected in a laboratory setting. The method relies on skeletal joint data, which can be extracted by Microsoft Kinect SDK/OpenNI. However, as indicated in [22], the software for skeleton extraction has a limited range of the skeletal tracking, approximately 1.5–4 m from the motion sensor. Such range is insufficient to capture falls in many areas of typical senior rooms. Moreover, in many typical ADLs the Kinect has difficulties in tracking all joints [31]. Thus, recent systems for reliable fall detection [14,2,32] do not take into account the Kinect RGB images and only rely on depth maps to delineate the person(s).

The remaining part of this paper is organized as follows. In Section 2, we outline the architecture and main ingredients of the fuzzy system. Section 3 is devoted to presentation of data processing. In Section 4, we discuss descriptors that are used to distinguish between falls and daily activities. Section 5 presents the fuzzy system. Experimental results are discussed in Section 6. Section 7 provides concluding remarks.

2. Architecture and main ingredients of the system

Although fuzzy inference has already been used to provide reliable representation of person falls, our approach differs significantly from the most significant work in this area [33,2] in several aspects. First of all, the final decision is taken on the basis of reasoning from a fuzzy knowledge and two linguistic variables, which are described by fuzzy sets, provided by two Mamdani-type fuzzy engines. Each engine is responsible for extracting different kinds of knowledge for later reasoning by a Sugeno-type fuzzy inference engine, which provides a crisp decision on either fall on no-fall. The first fuzzy engine performs reasoning about human pose, whereas the second one is responsible for reasoning about motion of the person. Secondly, two different sensors providing necessary redundancy are fused using fuzzy rules. Thirdly, the reasoning is done not for every frame, but it is executed only in case of a possible fall, which is detected with low computational cost through thresholding of the accelerometric data, see Fig. 1. The data needed to perform the inference about the speed of the pose transition are stored in a circular buffer, which delivers them to authenticate a possible fall event if necessary. Thanks to use of the disjoint linguistic variables in the first stage we reduced the computational overheads. Such a processing architecture has been designed to consume least amount energy while achieving reliable fall detection in real-time.

3. Data processing

At the beginning of this section, we discuss how the accelerometric data are used to trigger the processing of the depth maps. Afterwards, we present processing of depth data.

3.1. Triggering the processing of depth images

On the basis of the data acquired by the IMU (Inertial Measurement Unit) device the algorithm indicates a potential fall. The decision is taken on the basis of the thresholded total sum vector SV_{total}, which is calculated from the sampled data in the following manner:

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

where $A_x(t), A_y(t), A_z(t)$ is the acceleration in reference to the local x-, y-, and z-axes at time t, respectively. It contains both the dynamic and static acceleration components, and thus it is equal to 1 g for standing.

Fig. 2 illustrates sample plots of acceleration change curves for falling along with daily activities like going down the stairs, picking up an object, and sitting down–standing up. As we can observe on the discussed plots, during the falling phase the acceleration attained the value of 6 g, whereas during walking downstairs it attained the value of 3 g. As we already mentioned, it is equal to 1 g for standing. The sensor signals were acquired at a frequency of 256 Hz and resolution of 12 bits. The data were acquired by x-IMU device [34], which was worn near the pelvis by a middle aged person. Such placement of the inertial device has been chosen since this body part represents the major component of body mass and undergoes movement in most activities.

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 through fuzzy inference on both depth and accelerometric data. The critical issue in threshold-based approach is the selection of a appropriate threshold since if the value is too high the system (having sensitivity < 100%) might miss some real falls but never triggers false alarms (with 100% specificity), while if the threshold value is too low the system will detect all actual falls (100% sensitivity) but, at the same time, it may trigger some false alarms (specificity < 100%). Thus, choosing the threshold for accelerometric data to be utilized in a fall detector is a compromise between sensitivity and specificity. In our approach, if the value of SV_{total} is greater than 3 g then the system begins the extraction of the person and then executes fuzzy inference engine responsible for the final decision about the fall. Since the smallest acceleration measured from a fall is about 3 g [35], the assumed threshold allows us to filter all falls for further depth-based authentication.

3.2. Processing of depth data

The depth maps acquired by the Kinect sensor are continuously stored in a circular buffer. In a basic mode of person extraction, the depth image sequence from the circular buffer is utilized to extract a depth reference image, which is in turn employed to delineate the

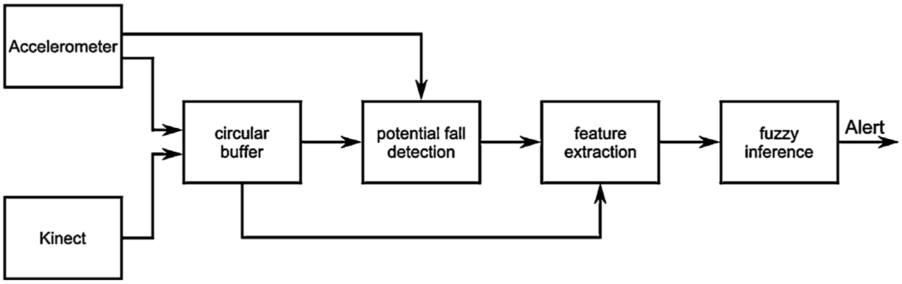

Fig. 1. Diagram of the fuzzy system for reliable fall detection.

person. The extraction of the person is achieved through differencing the current depth image from such a depth reference image. In the current implementation the depth reference image is updated on-line, which makes possible to perform fall detection in dynamic scenes. Each pixel in the depth reference image assumes a temporal median value of the past depth values from the circular buffer. In the initialization stage the system collects a number of the depth images, and for each pixel it assembles a list of the pixel values from the former images, which is then sorted in order to determine the temporal median. Given the sorted lists of pixels the depth reference image of the scene 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. For typical human motions and typical scene modifications, for instance, as a result of a movement of a chair, satisfactory results can be attained on the basis of fifteen depth maps. In order to avoid enclosure of the person into depth reference image, for example, if he/she is at standstill for a while, we take every fifteenth depth image acquired by the Kinect sensor. This means that for Kinect sensor acquiring the images at 30 Hz, the depth reference image is entirely refreshed in 7.5 s. The person is extracted with 30 fps through differencing the current depth image from the depth reference image updated in such a way.

In the basic mode of person extraction, in order to prevent disappearance of the person (on the binary image indicating the foreground objects) if he/she is not in motion for a while, i.e. to avoid assigning the person to the depth reference map, we perform updating of the depth reference image only when the person is in motion. The scene change can be inferred on the basis of differencing the consecutive depth maps, which is in fact the simplest method of motion detection.

Since the person is the most important subject in the fall detection, we consider also motion data from the accelerometer to sense the person's movement and scene changes. When the person is at rest, the algorithm acquires new data. If the person is not at rest during assumed in advance period of time, the algorithm extracts the foreground and then it determines the connected components to decide if a scene change took place. 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. If no substantial scene change

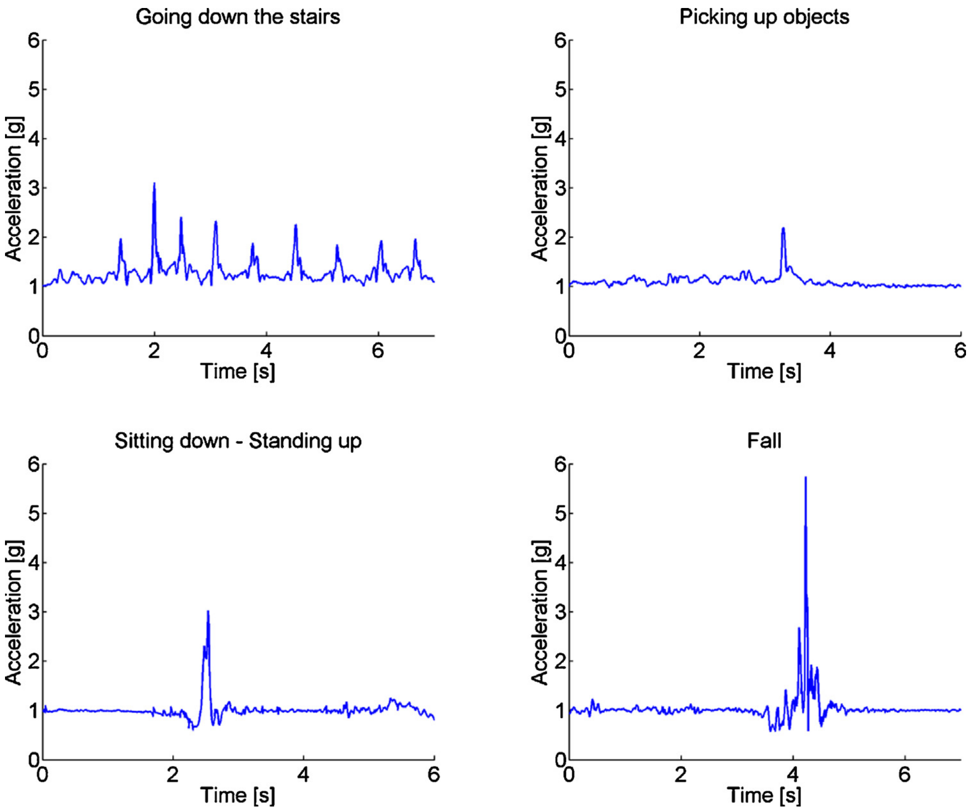

Fig. 2. Acceleration over time for walking downstairs, picking up an object, sitting down–standing up and falling.

is detected then there is no necessity to update the scene reference depth map, and in such a case the algorithm acquires a new depth map. As mentioned in the previous subsection, the person is detected on the basis of depth images updated in such a way only if SV_{total} exceeds the threshold. Prior to the person detection the system examines if the depth image has been updated on the basis of minimum ten consecutive images acquired before the fall, i.e. whether the depth image has been refreshed for the duration of five seconds preceding the beginning of fall. If not, the system takes the depth images from the circular buffer and updates the depth reference map. In order to cope with such circumstances we continuously store in the circular buffer 150 depth images. If the depth image has not been properly refreshed just before the fall the alarm is triggered with the delay.

It is worth noting that the procedure responsible for extraction of the depth reference map can be replaced by a block executing another algorithm, for instance relying on the well-known mixture of distributions [36], which were recently used to delineate the person in a system for fall detection [22]. Fig. 3 depicts sample depth maps and binary images with the extracted person. As already mentioned, in order to preserve the privacy of the user as well as to make the system ready to work any time, the RGB images corresponding to the depth maps are not acquired by our fall detection system.

As illustrated in Fig. 4, in certain circumstances the algorithm presented above can oversegment the person. When the algorithm detects such an oversegmentation it switches from the basic mode of person extraction to region growing based person extraction. Algorithm 1 presents the person extraction in the second mode. The input of the function Person Ex R G is current depth image D_{xy}, a depth reference image B_{xy} and a circular buffer Q_{xyz}. At the beginning, in the first call of the function, the algorithm determines the seed region for the region growing through differencing B_{xy} from the previous depth map, i.e. depth image acquired just before the person oversegmentation. A *roi* region surrounding the person and the segmented object is determined as well in order to restrict the processing area of the depth maps. In each call of the function, it extracts foreground F_{xy} and updates the *roi*. Afterwards, starting from the seed, it delineates the person blob and stores it in the image P_{xy}. The growing of the person region is executed until the blob area is smaller than a prespecified value or the depth/distance values are within a predefined range from the seed region. The values used in the stop condition are scaled regarding to the distance of the seed to the camera. Finally, given the delineated person, the discussed algorithm updates the *seed* region for the next call of the Region Growing.

Algorithm 1. Person extraction using region growing

```
Precondition: roi and seed are declared as static variables
1:      function $P_{xy}$ = PersonExRG($D_{xy}$, $B_{xy}$, $Q_{xyz}$)
2:          [roi, seed] = INIT($Q_{xyz}$, $B_{xy}$)              ▷ called only once
3:          $F_{xy} = |D_{xy} - B_{xy}|$
4:          roi = ROI($F_{xy}$, roi)
5:          $P_{xy}$ = RegionGrowing($D_{xy}$, roi, seed)
6:          seed = UpdateSeed($P_{xy}$)
7:      end function
```

As already mentioned, in the basic mode of person detection, the depth reference image is entirely refreshed in 7.5 s. When the person delineation is done in the second mode the depth image should be updated as fast as possible. This has been achieved through updating the depth reference image in *roi* region on the basis of person-free areas. Such person-free areas can be calculated straightforwardly through differencing the maps P_{xy} with the extracted person from the current depth image D_{xy}, see 3rd line in Algorithm 2. Such maps, where the areas belonging to person assume zero values, are then pushed on the stack, see call of Sadd function. The function LastPixel extracts the most recent non-person pixel in the *roi* area. After the switch from the region growing-based person detection to the basic mode, the depth reference image in the *roi* area is replaced by the B_{xy} map. A sample movie at http://fenix.univ.rzeszow.pl/ mkepski/demo/personseg.mp4 compares the person extraction in both modes.

Algorithm 2. Update of the depth reference map

```
Require: $S_{xy}$ is declared as static variable
1:    function [$B_{xy}$, $D'_{xy}$] = UDRM($D_{xy}$, $B_{xy}$, $Q_{xyz}$, $P_{xy}$, roi)
2:        $S_{xy}$ = INIT($Q_{xyz}$)                          ▷ called only once
3:        $D'_{xy} = D_{xy}(roi) - P_{xy}(roi)$
4:        $S_{xy}$ = Sadd($D'_{xy}$, roi)
5:        $B_{xy}$ = LastPixel($S_{xy}$, roi)
6:    end function
```

The switch from the basic mode of the person extraction to the mode based on the region growing is realized on the basis of the value of binary variable *regionGrMode*, see Algorithm 3. The simplest way to detect a connection of the person blob with another blob, i.e. to detect a situation in which the algorithm should switch from the basic mode to the second mode, is to compare the area of the current foreground with the area of the foreground in the previous map, see call of function Thresh. The region growing based mode should be finished if all pixels in the depth reference image B_{xy} are updated. This is achieved through summing the values of person-free areas D'_{xy} in a binary image L_{xy}, see 7th line in Algorithm 3. Having on regard that in D'_{xy} image the pixels belonging to person areas assume the value 0, the algorithm switches to the basic mode of person detection if all L_{xy} pixels in the *roi* area have depth values different from zero.

Algorithm 3. Setting conditional variable *regionGrMode*

```
Require: $L_{xy}$ and $F'_{xy}$ are declared as static variables
1:      function regionGrMode = RGM($D_{xy}$, $B_{xy}$)
2:          $F'_{xy}$ = INIT                                ▷ called only once
3:          $F_{xy} = |D_{xy} - B_{xy}|$
4:          regionGrMode = Thresh($F_{xy}$, $F'_{xy}$)
5:          $F'_{xy} = F_{xy}$
6:      end function
7:      function regionGrMode = updateRGM($D'_{xy}$, roi)
8:          $L_{xy}$ = INIT()                               ▷ called only once
9:          $L_{xy}$ = OR ($L_{xy}$, $D'_{xy}(roi)$)
10:         regionGrMode = IFstop($L_{xy}$)
11:     end function
```

The extraction of the person is executed only when the condition $SV_{\text{total}} > threshold$ is true. After the person extraction, the floor equation coefficients are uploaded to delineate the ground in the point cloud and then to extract features describing the activities. In particular, thanks to the floor extracted in advance, some point cloud features are extracted with regard to the floor. The depth and point cloud features are then employed to decide if a fall occurred. If the SV_{total} is smaller or equal to the assumed threshold then new data from the accelerometer is acquired.

4. Descriptors of the human activities

In this section, we explain the extraction of the ground plane that is used in calculating the fall descriptors. The descriptors are discussed in the second part of the section.

4.1. Ground plane extraction

After the transformation of the depth pixels to the 3D point cloud, the ground plane described by the equation $ax + by + cx + d = 0$ was recovered. Assuming that the optical axis of the Kinect camera is almost parallel to the floor, a subset of the points with the lowest altitude has been selected from the entire point cloud and

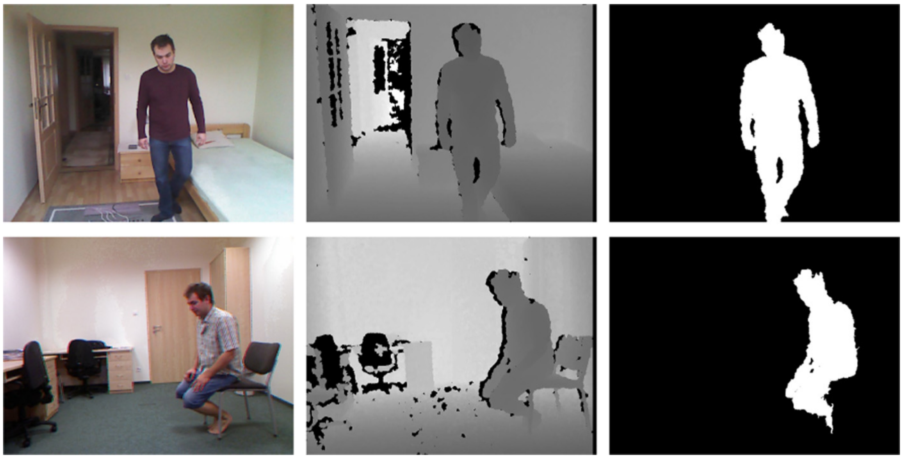

Fig. 3. Delineation of person using depth reference image. RGB images (left), depth (middle) and binary images depicting the delineated person (right).

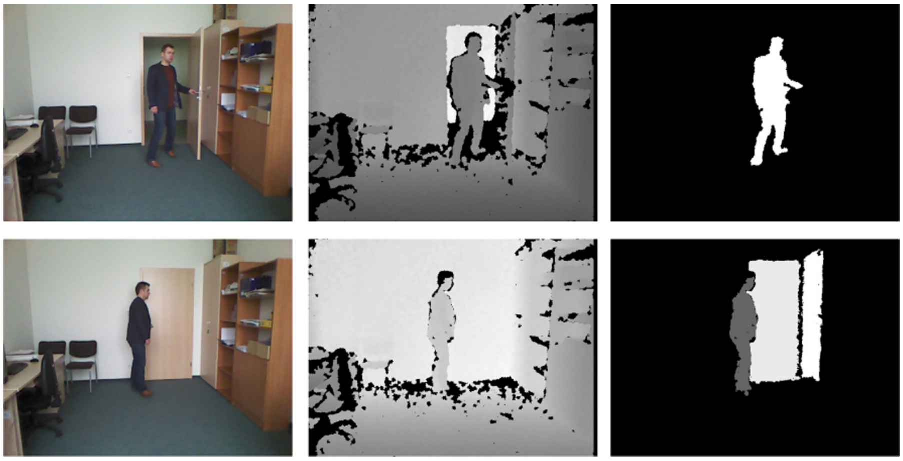

Fig. 4. Delineation of person on dynamic scene. RGB images (left), depth (middle) and binary images depicting the delineated person (right).

then utilized in the plane estimation. 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 [37]. The distance to the ground plane from the 3D centroid of point cloud corresponding to the segmented person has been determined on the basis of an expression for point–plane distance:

$$D(t) = \frac{|ax_c(t) + by_c(t) + cz_c(t) + d|}{\sqrt{a^2 + b^2 + c^2}} \tag{2}$$

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.

4.2. Depth features

The following features are extracted on the depth images in order to authenticate the fall hypotheses, and which are calculated if person's acceleration is above the preset threshold:

- H/W – a ratio of height to width of the person's bounding box in the depth maps.
- H/H_{max} – a proportion expressing the height of the person's surrounding box in the current frame to the physical height of the person, projected onto the depth image.
- D – the distance of the person's centroid to the floor.
- $max(\sigma_x, \sigma_z)$ – largest 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.

Fig. 5 depicts a person in depth images together with the H/W and H/H_{max} features. In order to determine the box enclosing the person the algorithm seeks for the largest blob in the binary image representing the foreground objects. As we can observe on the discussed depth maps with graphically marked features, the depth features assume quite different values during an example fall event and typical daily activities like walking and sitting on a chair.

The P_{40} descriptor is calculated on 3D point clouds. On the basis of the extracted person in the depth image a corresponding person's point cloud is determined in 3D space, see Fig. 6. Afterwards, a cuboid surrounding the person's point cloud is determined. Then, a sub-cuboid of 40 cm height and placed on the floor is extracted within such a cuboid. Finally, a ratio of the number of the points contained within the cuboid of 40 cm height to the number of the points being within the surrounding cuboid is calculated. The distance of each point to the floor is calculated on the basis of (2).

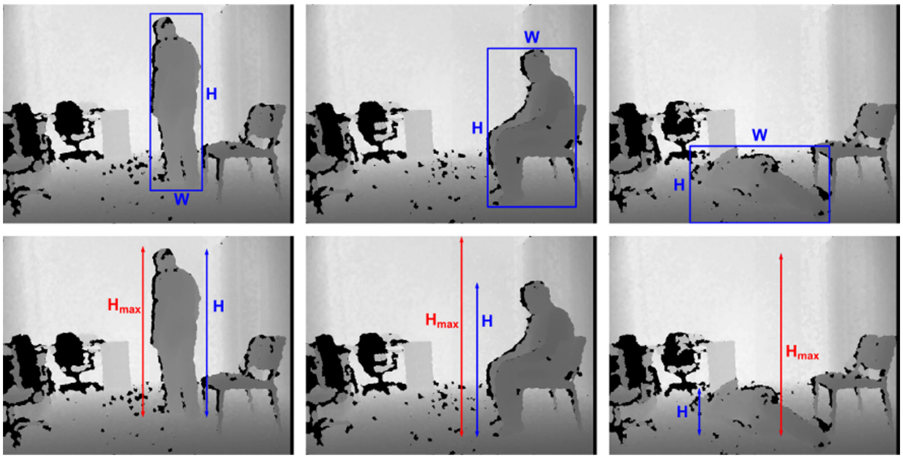

Fig. 5. Person on depth maps with the marked H/W and H/H_{max} features.

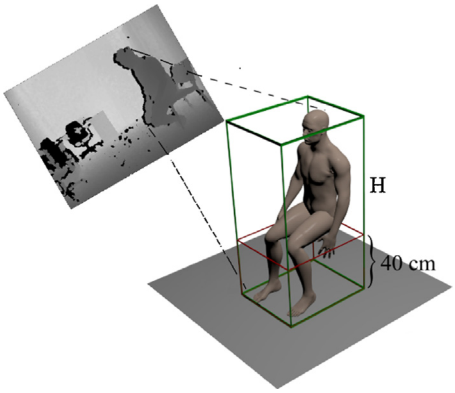

Fig. 6. Determining the P_{40} descriptor in the points cloud.

The 40 cm height of the cuboid has been chosen experimentally to include all 3D points belonging to a lying person on the floor.

5. Proposed fuzzy inference engine

At the beginning of this sections we outline Mamdani and Takagi-Sugeno fuzzy modeling. Then we justify reasons why we use fuzzy reasoning to discriminate between daily activities and falls. Afterwards, we discuss the proposed fuzzy engine.

5.1. Mamdani and Takagi-Sugeno fuzzy models

Fuzzy inference systems have become one of the most well-known applications of fuzzy logic. One of the reasons for significant interest on fuzzy inference systems is the ability to express the behavior of the system in an interpretable way for humans as well as to incorporate human expert knowledge and intuition with all its nuances, since domain experts are able to determine the main trends of the most influential variables in the system. This is because expert rules are based on evidence from data, a priori knowledge as well as intuition along with large experience and expertise. In consequence, typically they present a high level of generalization. Moreover, such a representation is highly interpretable. Assuming there are enough rules, a collection of fuzzy rules can accurately represent arbitrary input–output mappings [38]. In general, as the complexity of a system increases, the usefulness of fuzzy logic as a modeling tool increases.

A fuzzy inference system consists of three components: a rule-base, which contains a pool of fuzzy rules; a database of the membership functions used in the fuzzy rules; and a reasoning mechanism, which performs the inference. Two main types of fuzzy modeling schemes are the Mamdani and Takagi-Sugeno model [39]. In the Mamdani scheme each rule is represented by *if – then* conditional propositions [40]. A fuzzy system with two inputs x_1 and x_2 (antecedents) and one output y (consequent) is described by a collection of r conditional *if – then* propositions in the form:

$$\text{if}\, x_1 \,\text{is}\, A_1^k \,\text{and}\, x_2 \,\text{is}\, A_2^k \,\text{then}\, y^k \,\text{is}\, B^k, \;\text{for}\, k = 1, 2, \ldots, r \tag{3}$$

where A_1^k and A_2^k are fuzzy sets representing the kth antecedent pairs and B^k is fuzzy set representing the kth consequent. If we adopt max and min as our choice for the T-conorm and T-norm operators, respectively, and use max–min composition, the aggregated output for the r rules is given as follows:

$$\mu_{B_k}(y) = \max_k [\min[\mu_{A_1^k}(\text{in}(i)), \mu_{A_2^k}(\text{in}(j))]], \quad k = 1, 2, \ldots, r \tag{4}$$

For max-product (or correlation-product) implication technique, the aggregated output for a set of disjunctive rules can be determined in the following manner:

$$\mu_{B_k}(y) = \max_k [\mu_{A_1^k}(\text{in}(i)) \cdot \mu_{A_2^k}(\text{in}(j))], \quad k = 1, 2, \ldots, r \tag{5}$$

where the inferred output of each rule is a fuzzy set scaled down by its firing strength via algebraic product. Such a truncation or scaling is conducted for each rule, and then the truncated or scaled membership functions from each rule are aggregated. In conjunctive system of rules, the rules are connected by and connectives, whereas in case of disjunctive system the rules are connected by the or connectives. This kind of fuzzy system is also called a linguistic model because both the antecedents and the consequents are expressed as linguistic constraints. As a consequence, the knowledge base in Mamdani FIS is easy to understand and to maintain. The model structure is manually designed and the final model is neither trained nor computationally optimized, even though some heuristic tuning of the fuzzy membership functions is common in practice. Mamdani's model expresses the output using fuzzy terms based on the provided rules. Since this approach is not exclusively reliant on a dataset, a model that presents a high level of generalization can be obtained even when a small amount of experimental data is in disposal. All the existing fuzzy systems that are used as universal approximators are Mamdani fuzzy systems.

Takagi-Sugeno (TSK) fuzzy model consists of *if – then* rules that embody the fuzzy antecedents, and a mathematical function acting as the rule consequent part. A typical rule in a TSK model with two inputs x and y and output z, has the following form:

$$\text{if } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z = f(x, y) \tag{6}$$

where $z = f(x, y)$ is a crisp function in the consequent. Typically $f(x, y)$ is a polynomial function in the inputs x and y. In a TSK model each rule has a crisp output that is given by a function. As a result the overall output is determined via a weighted average defuzzification. It is a data driven approach in which the membership functions and rules are generated using an input–output data set. The Takagi-Sugeno model is typically constructed in two steps consisting of extracting the fuzzy rules and then optimizing the parameters of the linear regression models. The final output is a weighted average of a set of crisp values. The main difference between the two approaches lies in the consequent of fuzzy rules, since Mamdani fuzzy systems utilize fuzzy sets as rule consequent, whereas Takagi-Sugeno fuzzy systems employ linear functions of input variables as rule consequent. The first two stages of the fuzzy inference process, namely, fuzzification of the inputs and applying the fuzzy operator, are exactly the same. The main difference between them is that the Sugeno output membership functions are either linear or constant. Such a constant membership function gives us a zero-order Sugeno fuzzy model that can be viewed as a special case of the Mamdani FIS, in which each rule's consequent is specified by a fuzzy singleton.

5.2. Motivation

One of the main reason for choosing fuzzy inference was a desire to take the advantages of semantics to model fall events using noisy fall descriptors, given a limited dataset with simulated falls, which might differ from real-falls. As indicated in [41], in most cases, stringent requirements on the quality of training dataset are imposed in data-driven applications, including applications based of integration of ANN with fuzzy logic.

The goal of this work was to develop linguistically understandable classifier permitting reliable fall detection on noisy or vague data. As demonstrated in [42], fuzzy systems can give better results in comparison to classical approaches, particularly when data are imprecise. There is no doubt that a reliable system for fall detection should cope with such data, including occlusions. It is well known that observations of real-world fall events are inherently affected by uncertainty, vagueness, and imprecision. Very often in such real-world scenarios the observation errors do not follow a single probability distribution. In those cases, classical stochastic models of the error present a limited applicability. Moreover, our research findings reveal that state transitions in real-falls are highly unpredictable and they strongly depend not only on fall direction, type of substrate (wet/dry, parquet, carpet), whether in proximity of the individual is an object or not, but primary they depend on the way how a falling person is trying to save or to minimize the fall consequences. On the other hand, the model parameters of the classical classifiers or the automatically induced rules highly depend on the training set characteristics. Thus, in the case of insufficient training data or lack of data from real-world falls events, a considerable number of undesirable fall alerts can be generated. Moreover, the available semantics allows us to perform further linguistic summarization of observations, and for instance, recognize motionless long lie.

5.3. Fuzzy engine for discrimination between ADLs and fall events

As the number of fuzzy inputs and linguistic variables of each fuzzy set increases, the number of fuzzy rules grows exponentially. For n variables each of which can take m values, the number of rules is m^n. Thus, for the considered set of the fall descriptors, the number of fuzzy rules to be created by an expert is far too large.

Table 1
Fuzzy rules for fall event modeling.

Rule		Pose	Transition		Fall
1		Lying	Medium		Yes
2		Maybe	Medium		Yes
3		Lying	Fast		Yes
4		Maybe	Fast		Yes
5	if	Not-lying	Fast	then	No
6		Maybe	Slow		No
7		Not-lying	Slow		No
8		Not-lying	Medium		No
9		Lying	Slow		No

Having on regard that the lying pose and fall transitions concern static and dynamic actions, which need totally different observations to describe them, we developed a two-level fuzzy engine, where two fuzzy sets describing the lying pose and motion transitions are inferred first, whereas the final decision is generated through reasoning on such fuzzy sets, see Fig. 7. Thanks to such a structure optimization the number of fuzzy rules is far smaller. We demonstrated how to avoid introducing intermediate output variables with less or no physical meaning that are typical for hierarchical fuzzy systems. We proposed how to aggregate input variables and rules into two groups describing human posture and motion, which are close to human perception of the fall event. Apart from the reduction of fuzzy rules quantity, our approach offers improved capability of understanding and verification of such rules by a human expert. Due to the disjoint linguistic variables the computation complexity is reduced. Moreover, we obtained better flexibility and scalability of the system.

The Sugeno FIS has smaller computational demands in comparison to the Mamdani FIS because it does not involve the computationally expensive defuzzification process. Another rationale for choosing the Sugeno FIS is that it always generates continuous surfaces. The continuity of the output surface is essential in the fall detection system since the existence of discontinuities might result in substantially different outputs originating from similar inputs. For that reason, the fall alert is triggered by a Sugeno fuzzy inference system, see Fig. 7, which operates on confidence membership grades, provided by two Mamdani-type fuzzy inference systems. The output of the first Mamdani engine is a fuzzy set, which assigns grades of membership to the possible values of dynamic transitions, whereas the output of the second one is another fuzzy set assigning membership grades to possible body poses. As we can observe in Fig. 7, thanks to making use of Mamdani fuzzy inference systems as well as availability of the fuzzy sets and the membership grades, no fuzzification of the inputs is needed in the Sugeno inference system. Fig. 8 illustrates the lying pose confidence membership function and the transition confidence membership function. The parameters that define the fuzzy sets can be changed to focus more on the selected modality.

In Table 1, there are shown fuzzy rules for fall event modeling. As we can see, the lying pose confidence membership function, see also Fig. 8, is described by three fuzzy sets: lying, maybe, not-lying, whereas the transition confidence membership function is described by the following fuzzy sets: fast, medium, slow. Thus, the total number of fuzzy rules for fall modeling is equal to nine. The Sugeno FIS outcomes of fuzzy rules are characterized by yes and no crisp outputs. The Sugeno FIS adopts probabilistic or as fuzzy operator, product for implication, and weighted sum to aggregate all outputs.

The presented fuzzy rules for fall event modeling are consistent with our intuition. They were designed to prevent from undesirable generation of fall alarms. For instance, if person is lying on the floor, but the preceding person's motion was slow, no fall alarm is triggered. The slow motion means that the action has been performed

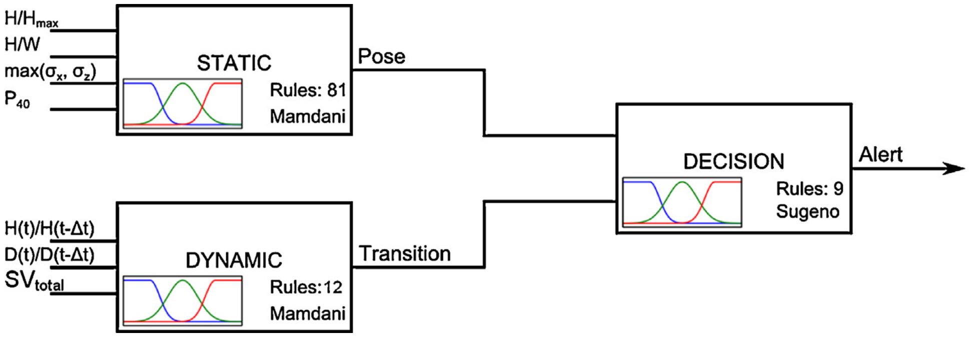

Fig. 7. Diagram of the fuzzy engine for discrimination between ADLs and fall events.

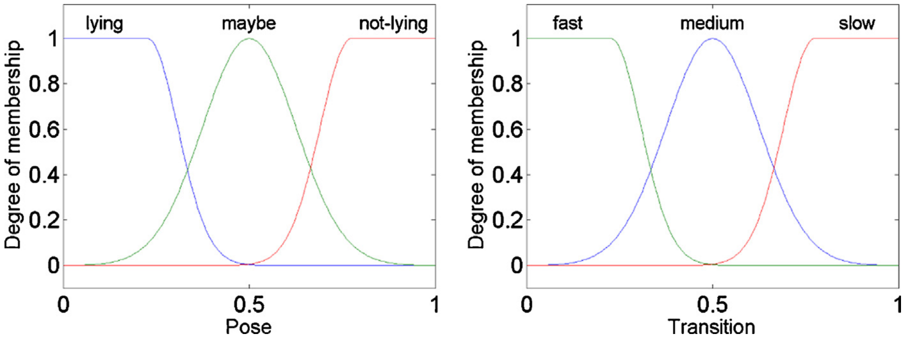

Fig. 8. Lying pose confidence membership function (left) and transition confidence membership function (right).

in time longer than 0.7 s. It is worth noting that the time interval between the loss of balance during the quite upright stance and the impact with the floor is longer than about 0.7 s [43]. The same decision is taken when the system is not sure about the person pose. Such a situation can take place in many everyday occurrences, including occlusions. In consequence, the everyday occurrences like lying or playing on the floor will not cause undesirable false alarms. On the other hand, even if a movement preceding the fall was fast, but a person is not in lying pose the alarm is not triggered. Such a situation can happen in case of quick sitting on a chair and in many similar occurrences.

5.4. *Pose FIS*

In order to describe various kinds of lying poses, four linguistic variables corresponding to the descriptors have been defined: H/W, H/H_{max}, $max(\sigma_x, \sigma_z)$ and P_{40}. They are described by three fuzzy sets: high, medium, and low. Fig. 9 depicts the plots of the membership functions for the discussed linguistic variables. The lying pose confidence membership function that is utilized in a Mamdani FIS is shown in Fig. 8. The membership functions are described by Gaussian curve, which has the advantage of being smooth and nonzero at all points. The parameters of the membership functions were manually tuned.

Table 2 shows the selected fuzzy rules that were used for modeling of lying pose. In total, 81 fuzzy rules were formulated to model such an event. The utilized descriptors of falls with corresponding linguistic variables pose sufficient redundancy to deal with person occlusions and imperfect observations.

Fig. 10 shows the input–output mapping for some pairs of the linguistic variables. In the discussed surface views the colors change according to the output values. As we can observe, the surfaces are more or less irregular. The horizontal plateaus are due to flat areas on the input sets, see for instance Fig. 9 and fuzzy set high of the linguistic variable $max(\sigma_x, \sigma_z)$.

5.5. *Transition FIS*

Motion features extracted on the basis of video sequences are used quite rarely in fall detection algorithms. On the other hand, the motion features are very useful since fall is a dynamic action, which is accompanied by changes of the person's shape. In this work, the motion features are utilized by a separate Mamdani-type FIS, which delivers three fuzzy sets describing the transition linguistic terms, see also Figs. 7 and 8 and Table 1. Fig. 11 depicts a sample plot of $D(t)$ vs. frame number during a person's fall. The vertical red line denotes the moment of the impact, which has been determined on the basis of the thresholded SV_{total}. As expected, during a typical fall there is a considerable change of $D(t)$ in a short time. On the basis of depth maps from our URFD dataset[1] we prepared the plots of $D(t)$ vs. time around the moment of the impact and then analyzed them in terms of determining the period of time in which a typical fall event takes place. After examining several such plots we reached the conclusion that the time Δt equal to 700 ms will be a good choose to describe the fall event. The experimental results in [44] show that falls can be detected with an average lead-time of 700 ms before the impact occurs, with no false alarms and sensitivity of 95.2%.

Given the Δt determined in such a way, we examined several ratios of the features with values determined in time t and $t - \Delta t$. The analysis of such plots showed that the features $D(t)/D(t - \Delta t)$ and $H(t)/H(t - \Delta t)$ have the best discrimination power in distinguishing between dynamic and slow body transitions. The discussed features together with SV_{total} signal from the accelerometer were used to define three linguistic variables, which in turn are described by two fuzzy sets: low and fast. Fig. 12 illustrates the partitioning of the transition domain into the discussed two fuzzy sets. As we see on the discussed figure, the acceleration domain is divided into three fuzzy sets, namely, low, medium and high. As

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

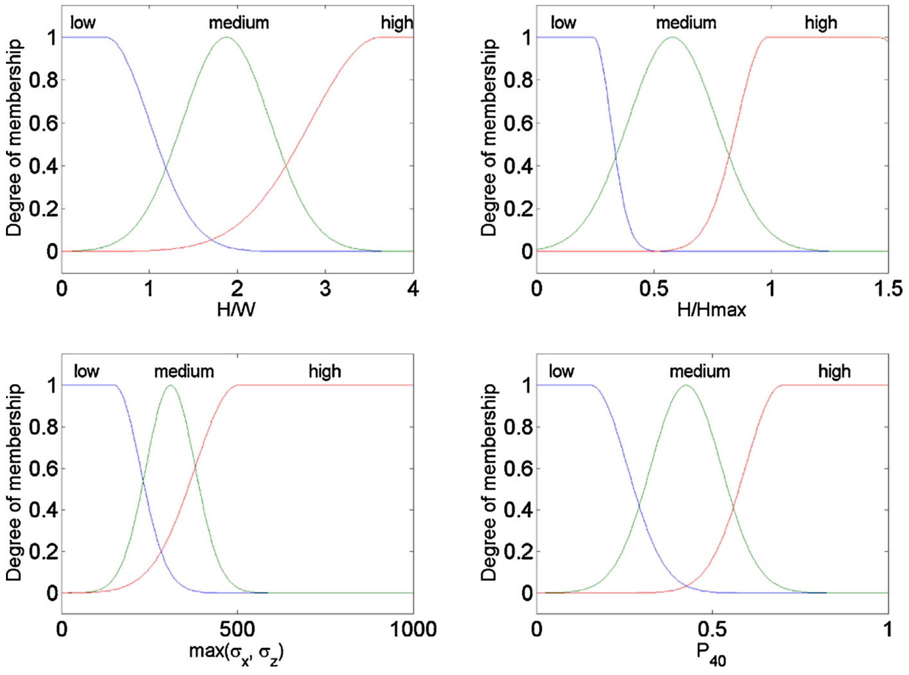

Fig. 9. Membership functions for the input linguistic variables H/W, H/H_{max}, $max(\sigma_x, \sigma_z)$ and P_{40}.

Table 2
Fuzzy rules for modeling of lying pose.

Rule		P_{40}	H/W	$max(\sigma_x, \sigma_z)$	H/H_{max}		Pose
1		Low	High	Low	High		Not-lying
2		Low	High	Low	Medium		Not-lying
3		Low	High	Low	Low		Not-lying
4		Low	High	Medium	High		Not-lying
5		Low	High	Medium	Medium		Not-lying
6		Low	High	Medium	Low		Not-lying
...	if					then	
76		High	Low	Medium	High		Maybe
77		High	Low	Medium	Medium		Lying
78		High	Low	Medium	Low		Lying
79		High	Low	Low	High		Lying
80		High	Low	Low	Medium		Lying
81		High	Low	Low	Low		Lying

Table 3
Fuzzy rules for modeling of body transitions.

Rule		$\frac{H(t)}{H(t-\Delta t)}$	$\frac{D(t)}{D(t-\Delta t)}$	SV_{total}		Transition
1		Low	Low	Low		Fast
2		Low	Low	Medium		Fast
3		Low	Low	Low		Medium
4		Low	High	High		Fast
5		Low	High	Medium		Medium
6		Low	High	Low		Slow
7	if	High	Low	High	then	Fast
8		High	Low	Medium		Medium
9		High	Low	Low		Slow
10		High	High	High		Medium
11		High	High	Medium		Slow
12		High	High	Low		Slow

we can notice, the membership functions are described by Gaussian curve. The depth based transition domain is divided into two fuzzy sets due to possible imprecision in observations of the body motion, for instance due to occlusion, etc. The transition confidence membership function that is utilized in a Mamdani FIS is shown in Fig. 8.

Table 3 shows rules for modeling body transitions just before the body impact. In total, twelve fuzzy rules were formulated to model such body transitions. The data delivered by two different sensors undergo fusion. On the other hand, the depth data are processed in the context, which is provided by accelerometric data. What's more, the fuzzy inference engine is executed only if the thresholded SV_{total} is larger than a presumed threshold, see Fig. 1. That means that such a binary rule is used to decide if the fuzzy inference is needed to authenticate the hypothesis. The discussed rules can deal with person occlusions and imperfect observations. They allow us to extract different kind of the knowledge in comparison to the knowledge inferred by the pose FIS.

Fig. 13 illustrates the input–output mapping for some pairs of the linguistic variables, which are used in reasoning by the transition FIS. The discussed surface views result from a rule base with four and six rules, respectively. As we can observe, the resulting surfaces are relatively regular.

In Mamdani engines we selected max and algebraic product for the T-norm and T-conorm operators and employed max-product composition in the rule-base.

6. Experimental results

The fuzzy system has been evaluated on our freely available UR Fall Detection dataset. It consists of 30 image sequences with

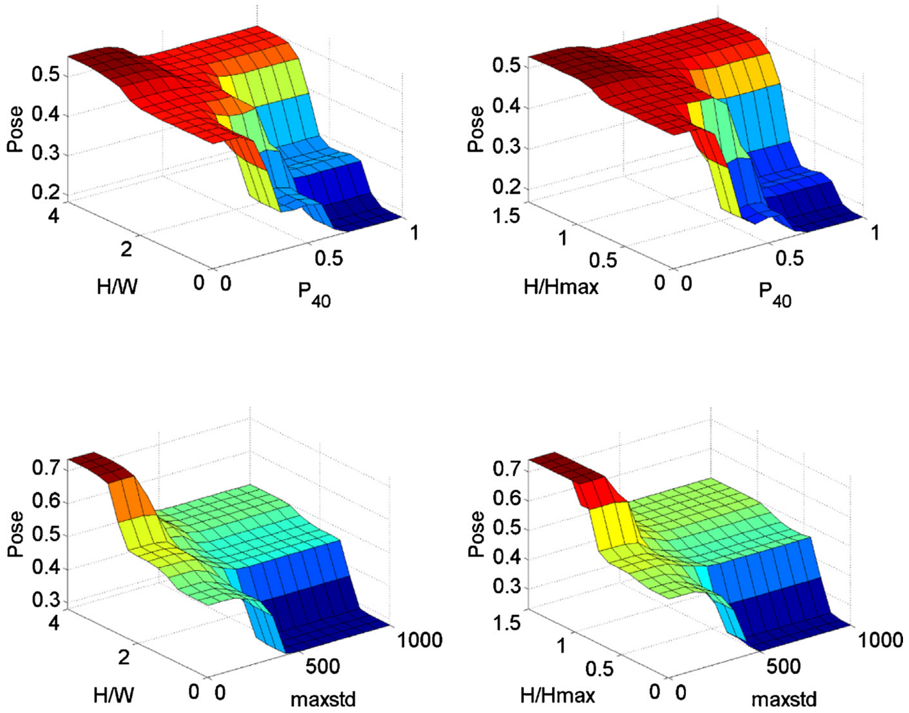

Fig. 10. Input–output mapping surface views for pairs of the linguistic variables.

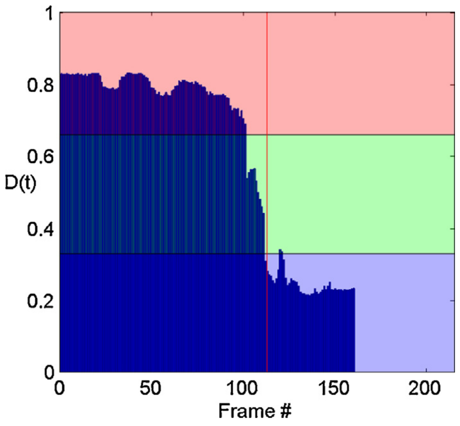

Fig. 11. $D(t)$ vs. frame number during a fall. The vertical red line denotes the moment of the impact. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

simulated falls, 30 sequences with some daily activities like walking, sitting down, crouching down, and 10 sequences with fall-like activities as quick lying on the floor and lying on the bed/couch. Two kinds of falls were performed by five persons with different age, namely from standing position and from sitting on the chair. The number of images in the sequences with falls is equal to 3000, whereas the number of images from sequences with ADLs is equal to 10,000. All RGB-D images are synchronized with motion data, which were acquired by the x-IMU inertial device. The sensing unit was worn near the spine on the lower back using an elastic belt around the waist. The motion data contains the acceleration over time in the x-, y-, and z-axes together with the precalculated SV_{total} values.

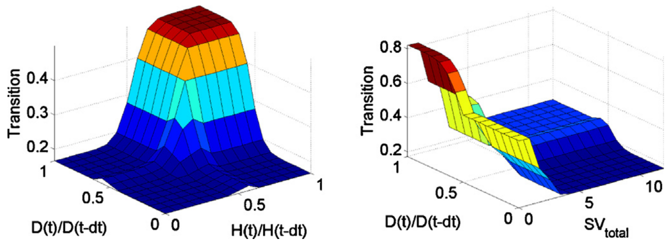

Fig. 13. Surface views of rule base in Table 3.

In order to evaluate the effectiveness of the pose FIS as well as to assess the usefulness of the descriptors responsible for separation of the lying pose from typical activities we added a defuzzification to the pose FIS and evaluated it on a set of depth maps. In total 2395 images were selected from our URFD dataset and other image sequences, which were recorded in typical rooms, like office, classroom, etc. The selected image set consists of 1492 images

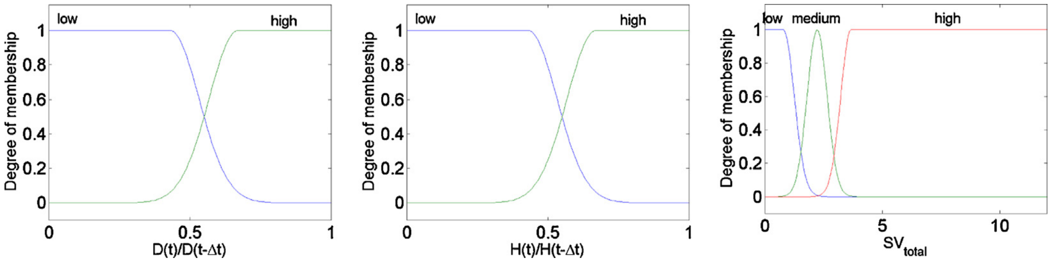

Fig. 12. Membership functions for the input linguistic variables $D(t)/D(t-\Delta t)$, $H(t)/H(t-\Delta t)$ and SV_{total}.

Table 4
Performance of lying pose classification using static variables and Mamdani FIS with center-of-gravity defuzzification.

	True		
	Fall	Not fall	
Estimated			
Fall	903	19	Accuracy=99.22%
Not fall	0	1503	Precision=97.94%
	Sens.=100%	Spec.=98.75%	

Table 5
Performance of fall detection on URFD data (sequences 1–70).

	Method			
	Fuzzy static+dyn. var.	Fuzzy dyn. var.	Fuzzy ANFIS	SVM
Results				
Accuracy	97.14%	92.86%	95.71%	95.71%
Precision	93.75%	85.71%	90.90%	90.90%
Sensitivity	100.00%	100.00%	100.00%	100.00%
Specificity	95.00%	87.50%	92.50%	92.50%

with typical ADLs like walking, sitting down and crouching down, whereas 903 images depict a person lying on the floor. The aforementioned depth maps collection was employed to determine the features discussed in Section 4.2. Table 4 shows the classification performance and the potential of the lying pose descriptors. The discussed results were obtained in 10-fold cross-validation [45]. It is worth noting that the presented results concern crisp outputs, whereas in our fuzzy system we utilize fuzzy sets and the inferred membership grades. As we can observe, all fall events were distinguished correctly, whereas some ADLs were classified as falls. The role of the Sugeno engine, operating both on membership grades to possible body poses and the membership grades to the possible values of dynamic transitions, is to reduce the number of such misclassifications.

Table 5 shows the performances of fall detection, which were obtained by our fuzzy system operating both on static and dynamic variables, a Mamdani FIS with a center of gravity defuzzification and operating on dynamic variables only, a fuzzy ANFIS [46], and a linear SVM [47], respectively. Given limited amount of the training data the SVM has been chosen as a representative classifier since it performs structural risk minimization to achieve good generalization performance. All features used to train and test the ANFIS and the SVM classifiers were normalized to zero mean and unit variance. As we see, our system achieves the best results in terms of accuracy, precision, sensitivity and specificity. Three key performance metrics are sensitivity, specificity and precision [45], because the accuracy always lies between the sensitivity and the specificity. All classifiers achieve 100% sensitivity and this means that all falls are assigned to the fall class. A perfect specificity implies that no ADL may be erroneously recognized as a fall event. The best 95% specificity is achieved by our fuzzy system. This is relatively good result given that the dataset contains ten hard sequences, in which the actions were done quickly and which motion patterns are similar to fall ones. The precision is defined as the ratio of true positives among all the positively labeled activities. On the utilized dataset we did not notice statistically significant difference in the detection performance between falls from upright position and those from sitting position.

As we can notice in Table 5, the ANFIS and SVM operating on seven features, i.e. H/W, H/H_{max}, $max(\sigma_x, \sigma_z)$, P_{40}, $D(t)/D(t-\Delta t)$, $H(t)/H(t-\Delta t)$ and SV_{total}, achieve identical results. They are worse in comparison to results achieved by our fuzzy system and operating on seven linguistic variables. The classifiers were trained on the collection of aforementioned collection of depth maps. The number of rules utilized by ANFIS is equal to 256. The rules are quite different from the rules utilized by our system. Moreover, the interpretation of such ANFIS rules is very hard. What's more, the membership grades inferred by our system are connected with physical behavior of the observed attributes and can be used at higher level of linguistic summarization. If even in some real environments the SVM may achieve better classification results, the advantage of the proposed linguistically understandable classifiers is that they are better suited to the summarization of human behaviors. As we can see in Table 5, the Mamdani FIS operating on dynamic variables only, i.e. on $D(t)/D(t-\Delta t)$, $H(t)/H(t-\Delta t)$ and SV_{total} achieves quite good results. The presented results confirm the importance of dynamic features in fall detection.

The system has been designed to consume least amount energy while achieving reliable fall detection in real-time. This aim has been achieved thanks to the use of accelerometer to filter at low computational cost most of the non-fall activities. The system has been implemented on low-cost PandaBoard. Details about communication between processes to achieve real-time processing on an embedded platform are given in [48].

The system as well as the presented algorithms were compared with relevant approaches. Another state-of-the-art classifiers, including the k-nn classifier, provide worse or at most the same results as SVM/ANFIS classifiers. A k-nn with 5 neighbors has been built on the same set of identically scaled features. To the best of our knowledge, the URFD dataset is the only publicly available dataset for evaluation of the fall detection algorithms using depth and/or accelerometric data. We also evaluated the performance of the fall detection using features, which were used in the relevant work. However, the early approaches to fall detection using Kinect used simple cues [10], like distance of the person's gravity center to the floor [18], and in consequence they trigger too much false alarms. The research results from [22] indicate that the body velocity prior to the occlusion, which was employed in the work of Rougier et al., can trigger a vast number of false alarms (on the order of 20 per day) for a person walking out of the scene. They also demonstrated that after disabling the velocity component, the discussed algorithm triggered a large number of false alarms at low detection rates. The false alarms were caused by a variety of everyday occurrences, including pets moving on the floor, items dropped or moved on the floor, and residents and visitors lying or playing on the floor. Such typical everyday occurrences are filtered reliably by our algorithm at low computational cost.

The performance of the fall detector achieved on benchmark data is very important. Last but not least is the architecture of the system. In our opinion, decisions about the fall alarm should be made taking into account the context of the situation. Our systems considers the context of the situation, i.e. an alert is triggered in the case of rapid movement proceeding the body impact. The proposed architecture is flexible and the system can be extended about additional contextual inputs like sound and/or floor vibrations. One of the major advantages of fuzzy logic over other existing fusion methods is that it permits easy fusion by using linguistic variables and the rules. It is worth noting that falls are accidents, so it is not easy to collect data of real falls, particularly of the elderly. As we already mentioned, since this approach is not exclusively reliant on a dataset, a model with high level of generalization can be obtained, even when a small amount of multimodal data is in disposal.

One of the limitations of the utilized depth sensor is relatively narrow field-of-view. However, as demonstrated in [49], the observed area can be expanded by the use of depth sensor mounted on a pan-tilt unit. Future work includes further improvement of the person extraction algorithm along with verification whether a detected blob belongs to the person undergoing monitoring. Multiple depth sensors per room, including ToF cameras will be utilized to improve dealing with occlusions.

7. Conclusions

We have demonstrated a flexible framework for combining different modalities in order to improve the fall detection. The features are extracted on both depth maps and accelerometric data and then used along with fuzzy inference to determine the state of the resident. In the proposed architecture an accelerometer is utilized to indicate an eventual fall. A fall hypothesis is then authenticated by a two-stage fuzzy system, which fuses depth maps and accelerometric data. The aim of the first stage, which is composed of two Mamdani engines, is to infer separately the static pose and dynamic transition. The second stage of Takagi-Sugeno engine provides a crisp decision on either fall or no-fall. This resulted in reduced computation complexity due to the disjoint linguistic variables at the first stage, and reduced computation cost due to extracting of the depth features only when a fall is likely to have just happened as indicated by the acceleration measurement values. The proposed linguistically understandable classifier can be generalized to other applications especially when sensor fusion is involved or human activity summarization is required. As demonstrated experimentally, it can be particularly useful in fall detection since frequently a reduced amount of training data is in disposal. We showed that fusion and manipulation of linguistic variables and rules is easy. We demonstrated experimentally that the proposed framework permits reliable and unobtrusive fall detection in real-time and at low computational cost.

Acknowledgements

This work was partially supported by Faculty of Computer Science, Electronics and Telecommunications of AGH Univ., by the Interdisciplinary Centre for Computational Modelling of Univ. of Rzeszow as well as by Polish National Science Center (NCN) under a research grant conducted over years 2015–2018.

References

[1] P. Durso, R. Massari, Fuzzy clustering of human activity patterns, Fuzzy Sets Syst. 215 (March) (2013) 29–54.
[2] T. Banerjee, J. Keller, M. Skubic, E. Stone, Day or night activity recognition from video using fuzzy clustering techniques, IEEE Trans. Fuzzy Syst. 22 (June (3)) (2014) 483–493.
[3] L. Zadeh, A fuzzy-algorithmic approach to the definition of complex or imprecise concepts, Int. J. Man-Mach. Stud. 8 (3) (1976) 249–291.
[4] O. Masoud, N. Papanikolopoulos, A method for human action recognition, Image Vis. Comput. 21 (8) (2003) 729–743.
[5] H. Wang, A. Klaser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: IEEE Int. Conf. on Comp. Vision and Pattern Rec. (CVPR), 2011, pp. 3169–3176.
[6] P. Borges, N. Conci, A. Cavallaro, Video-based human behavior understanding: a survey, IEEE Trans. Circuits Syst. Video Technol. 23 (November (11)) (2013) 1993–2008.
[7] Y. Zigel, D. Litvak, I. Gannot, A method for automatic fall detection of elderly people using floor vibrations and sound-proof of concept on human mimicking doll falls, IEEE Trans. Biomed. Eng. 56 (12) (2009) 2858–2867.
[8] P. Rashidi, A. Mihailidis, A survey on ambient-assisted living tools for older adults, IEEE J. Biomed. Health Inform. 17 (3) (2013) 579–590.
[9] Y. Li, K.C. Ho, M. Popescu, Efficient source separation algorithms for acoustic fall detection using a Microsoft Kinect, IEEE Trans. Biomed. Eng. 61 (3) (2014) 745–755.
[10] D. Webster, O. Celik, Systematic review of Kinect applications in elderly care and stroke rehabilitation, J. NeuroEng. Rehabil. 11 (2014).
[11] R. Igual, C. Medrano, I. Plaza, Challenges, issues and trends in fall detection systems, BioMed. Eng. OnLine 12 (2013).
[12] A. Bourke, J. O'Brien, G. Lyons, Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm, Gait Posture 26 (2) (2007) 194–199.
[13] Q. Li, J.A. Stankovic, M.A. Hanson, A.T. Barth, J. Lach, G. Zhou, Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information, in: 2009 Int Conf. on Body Sensor Networks, vol. 9, 2009, pp. 138–143.
[14] M. Kepski, B. Kwolek, I. Austvoll, Fuzzy inference-based reliable fall detection using Kinect and accelerometer, in: The 11th Int. Conf. on Artificial Intelligence and Soft Computing, ser. LNCS, vol. 7267, April 29–May 3, 2012, Springer, 2012, pp. 266–273.
[15] M.V. Sokolova, J. Serrano-Cuerda, J.C. Castillo, A. Fernández-Caballero, A fuzzy model for human fall detection in infrared video, J. Intell. Fuzzy Syst.: Appl. Eng. Technol. 24 (March (2)) (2013) 215–228.
[16] B.E. Demiroz, A. Salah, L. Akarun, Coupling fall detection and tracking in omnidirectional cameras Human Behavior Understanding, ser. LNCS, vol. 8749, Springer, 2014, pp. 73–85.
[17] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, R. Moore, Real-time human pose recognition in parts from single depth images, Commun. ACM 56 (January (1)) (2013) 116–124.
[18] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, J. Meunier, Fall detection from depth map video sequences Toward Useful Services for Elderly and People with Disabilities, ser. LNCS, vol. 6719, Springer, 2011, pp. 121–128.
[19] C. Chen, R. Jafari, N. Kehtarnavaz, Improving human action recognition using fusion of depth camera and inertial sensors, IEEE Trans. Hum. Mach. Syst. 45 (1) (2014).
[20] K. Cuppens, C.-W. Chen, K.B.-Y. Wong, A. Van de Vel, L. Lagae, B. Ceulemans, T. Tuytelaars, S. Van Huffel, B. Vanrumste, H. Aghajan, Integrating video and accelerometer signals for nocturnal epileptic seizure detection, in: Proc. of the 14th ACM Int. Conf. on Multimodal Interaction, ACM, New York, NY, USA, 2012, pp. 161–164.
[21] B. Delachaux, J. Rebetez, A. Perez-Uribe, H.F. Satizábal Mejia, Indoor activity recognition by combining one-vs.-all neural network classifiers exploiting wearable and depth sensors Advances in Comp. Intelligence, ser. LNCS, vol. 7903, Springer, 2013, pp. 216–223.
[22] E.E. Stone, M. Skubic, Fall detection in homes of older adults using the Microsoft Kinect, IEEE J. Biomed. Health Inform. 19 (2015).
[23] B. Kwolek, Face tracking system based on color, stereovision and elliptical shape features, in: Proc. of the IEEE Conf. on Advanced Video and Signal Based Surveillance, IEEE Computer Society, Washington, DC, USA, 2003, pp. 21–26.
[24] C. Rougier, J. Meunier, A. St-Arnaud, J. Rousseau, 3D head tracking for fall detection using a single calibrated camera, Image Vis. Comput. 31 (3) (2013) 246–254.
[25] G.V. Raju, J. Zhou, R.A. Kisner, Hierarchical fuzzy control, Int. J. Control 54 (1991) 1201–1216.
[26] L.-X. Wang, Universal approximation by hierarchical fuzzy systems, Fuzzy Sets Syst. 93 (2) (1998) 223–230.
[27] M.G. Joo, J.S. Lee, Universal approximation by hierarchical fuzzy system with constraints on the fuzzy rule, Fuzzy Sets Syst. 130 (2) (2002) 175–188.
[28] L.I. Kuncheva, Fuzzy Classifier Design, ser. Studies in Fuzziness and Soft Computing, Physica-Verlag GmbH, Heidelberg, Germany, 2000.
[29] D. Brulin, Y. Benezeth, E. Courtial, Posture recognition based on fuzzy logic for home monitoring of the elderly, IEEE Trans. Inf. Technol. Biomed. 16 (September (5)) (2012) 974–982.
[30] R. Planinc, M. Kampel, Robust fall detection by combining 3D data and fuzzy logic, in: Proc. of the 11th Asian Conf. on Computer Vision, Springer, 2013, November, pp. 121–132.
[31] T.-T.-H. Tran, T.-L. Le, J. Morel, An analysis on human fall detection using skeleton from Microsoft Kinect, in: IEEE Fifth Int. Conf. on Communications and Electronics, 2014, July, pp. 484–489.
[32] T. Banerjee, J.M. Keller, M. Popescu, M. Skubic, Recognizing complex instrumental activities of daily living using scene information and fuzzy logic, Comput. Vis. Image Understand. 140 (2015) 68–82.
[33] D. Anderson, R.H. Luke, J.M. Keller, M. Skubic, M.J. Rantz, M.A. Aud, Modeling human activity from voxel person using fuzzy logic, IEEE Trans. Fuzzy Syst. 17 (February (1)) (2009) 39–49.
[34] The x-IMU Inertial Measurement Unit. Available from: http://www.x-io.co.uk/products/x-imu/.
[35] J. Chen, K. Kwong, D. Chang, J. Luk, R. Bajcsy, Wearable sensors for reliable fall detection, in: 27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005, pp. 3551–3554.
[36] C. Stauffer, W.E.L. Grimson, Learning patterns of activity using real-time tracking, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 747–757.
[37] 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 (June (6)) (1981) 381–395.
[38] B. Kosko, Fuzzy systems as universal approximators, IEEE Trans. Comput. 43 (November (11)) (1994) 1329–1333.
[39] A. Abraham, Rule-Based Expert Systems, John Wiley & Sons, Ltd, 2005.
[40] E. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man-Mach. Stud. 7 (1) (1975) 1–13.
[41] A. Tewari, M.-U. Macdonald, Knowledge-based parameter identification of TSK fuzzy models, Appl. Soft Comput. 10 (2) (2010) 481–489.
[42] L. Sanchez, I. Couso, Advocating the use of imprecisely observed data in genetic fuzzy systems, IEEE Trans. Fuzzy Syst. (2007) 551–562.
[43] 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.
[44] M. Nyan, F.E. Tay, E. Murugasu, A wearable system for pre-impact fall detection, J. Biomech. 41 (16) (2008) 3475–3481.
[45] T.J. Hastie, R.J. Tibshirani, J.H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, ser. Springer Series in Statistics, Springer, New York, 2009.

[46] J.-S. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybern. 23 (May (3)) (1993) 665–685.

[47] C.-C. Chang, C.-J. Lin, LIBSVM: A library for Support Vector Machines, ACM Trans. Intell. Syst. Technol. 2 (May (3)) (2011) 1–27.

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

[49] M. Kepski, B. Kwolek, Detecting human falls with 3-axis accelerometer and depth sensor, in: 36th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC), 2014, pp. 770–773.