Journal of Ambient Intelligence and Smart Environments 1 (2015) 1–19
IOS Press

Camera-based fall detection using real-world versus simulated data: how far are we from the solution?

Glen Debard [a,b,c,*], Marc Mertens [a,d], Mieke Deschodt [e,f], Ellen Vlaeyen [e], Els Devriendt [e,f], Eddy Dejaeger [f,g], Koen Milisen [e,f], Jos Tournoy [f,g], Tom Croonenborghs [b,d], Toon Goedemé [c,h], Tinne Tuytelaars [c,i] and Bart Vanrumste [b,j,k]

[a] *Thomas More Kempen, MOBILAB, Geel, Belgium*
[b] *KU Leuven Technology Campus Geel, AdvISe, Geel, Belgium*
[c] *KU Leuven, ESAT-PSI, Leuven, Belgium*
[d] *KU Leuven, Department of Computer Science, DTAI, Leuven, Belgium*
[e] *KU Leuven, Department of Public Health and Primary Care, Health Services and Nursing Research, Belgium*
[f] *University Hospitals Leuven, Geriatric Medicine, Leuven, Belgium*
[g] *KU Leuven, Department of Clinical and Experimental Medicine, Leuven, Belgium*
[h] *KU Leuven Campus De Nayer, EAVISE, Sint-Katelijne-Waver, Belgium*
[i] *iMinds Multimedia Technologies Department, Leuven, Belgium*
[j] *KU Leuven, ESAT-STADIUS, Leuven, Belgium*
[k] *iMinds Medical Information Technology Department, Leuven, Belgium*

Abstract. Several new algorithms for camera-based fall detection have been proposed, with the aim of reliably alerting caregivers about older persons' falls at home. These algorithms have been evaluated almost exclusively using brief segments of video data captured in artificial environments under optimal conditions and with falls simulated by actors. By contrast, we collected real-life video data recorded over several months at seven older persons' residences. Here, we report on our fall-detection algorithm based on the state-of-the-art, and we present an analysis of the real-life video data. The performance of our detection algorithm was compared with the performance of three previously reported algorithms that used a publicly available simulation data set. All four algorithms produced similar results when using the simulated data. However, the performance of our algorithm degraded drastically when evaluating falls in the real-life data. The false alarm rate was especially high, showing that some challenges still need to be met to make the system sufficiently robust to deploy in real-world situations. We conclude that using more realistic data sets that include longer video recordings and a broad range of activities are essential to reveal weaknesses in fall-detection algorithms.

Keywords: Fall Detection, Video Surveillance, Ambient Assisted Living, Evaluation

1. Introduction

Between 30 and 45% of the community-dwelling persons aged 65 or over and more than half of the institutionalized older persons fall at least once a year [33,34,55]. Severe injuries are suffered in 10 to 15% of these falls [34]. A large number of persons are not able to get up without help after falling. This occurs in 20% to 43% of community-dwelling older persons and 66% to 100% of residents in institutional settings [10,36,56]. Not receiving aid in time can lead to fur-

*Corresponding author: Kleinhoefstraat 4, 2440 Geel, Belgium +3214802296, glen.debard@thomasmore.be.

ther complications such as dehydration, pressure ulcers, and even death. Not only physical injuries but also psychological consequences are important, particularly fear of falling which leads to loss of self-confidence and independence [10,34]. Taking into account the ever-increasing aging of the population, it is obvious that rapid and accurate detection of fall incidents is becoming increasingly important to reduce the clinical risks associated with lying on the floor for long periods.

Different types of fall detectors are presently available, an overview of which can be found in [35] and [62]. Currently wearable sensors are used most frequently. However, a market study conducted by SeniorWatch [49] discovered that fall sensors are not worn continuously (e.g., removed at bedtime). Also, for devices requiring a button to be pressed, as with a panic button, older persons and especially those with cognitive impairment are not always capable of activating the alarm system through a button press [10]. As a result, many falls remain undetected. A camera-based system, on the other hand, has the potential to overcome these limitations, because it is contactless (i.e., mounted in the environment, not on the person) and does not require initiative of the person to be operated.

Because of these advantages, several research groups have focused on camera-based fall detection algorithms in the last decade [1,3–6,11,15,16,18,20,23–25,29,31,32,37,38,44–47,50,53,54,57]. However, to the best of our knowledge, performance of all systems published to date has always been evaluated using simulated data sets. Moreover, the falls were recorded in artificial environments, and the simulators were mostly younger persons. More has to be considered than simply the difference between simulated falls and real-world falls when evaluating fall-detection systems. Also, it is important to consider that most fallers fall only once a year (although one-third of seniors fall at least twice or more a year) [34]. Table 1 shows some examples of the number of fall- versus non-fall-related data in the literature that report using simulation data sets. Most research has focused almost exclusively on detecting falls using data sets that comprise mostly fall simulations, without verifying the systems' robustness with regard to rejecting non-fall-related activities. Also, the length of video fragments used previously appears to be especially short, given that a real-world fall-detection system has to work continuously.

By contrast, the present study used an extensive real-life video data set. We installed cameras to monitor seven older persons at their residence for a period ranging from 3 months to 1.5 years. Four of the persons had a documented high risk of falling. For a number of the fall events that were recorded, an analysis was performed from a clinical point of view, the results of which can be found in [59].

The aim of this study was threefold: Can conclusions derived from the analyses of simulated data be extrapolated to real-life data? Can we reliably detect real-life falls using a state-of-the-art system that is benchmarked only on simulated data? What is the performance of a state-of-the-art fall-detection algorithm using real-life data?

In the remainder of the paper, we provide an overview of the literature on camera-based fall detection. In Section 3, the data sets we used in this study are discussed and an overview of the real-life falls is given. Our new fall-detection algorithm is discussed in detail in Section 4, while Section 5 presents the validation results of the proposed algorithm, which used both simulated and real-life video data. A discussion of the results and some possible improvements in the fall-detection system are presented in Section 6. We conclude in Section 7.

2. Related work

Two main approaches have been used in most fall detection systems reported in the literature: (1) one that attempts to detect the action of falling directly (e.g., [1,4–6,11,15,16,18,20,23–25,29,32,44,46,47,53,54,57]); and (2) one that is designed to detect unusual events, like changes in a person's life pattern, in general (e.g., [3,31,37,38,45,50]). Approaches in the latter category rely on collecting indirect evidence to infer fall incidents, such as prolonged inactivity at unusual locations. The disadvantage of this approach is that a longer response time is required, since a certain amount of time needs to pass in order to detect abnormal inactivity.

More recently, some research groups have started using multi-camera networks of four or more cameras in the same room for fall detection (e.g., [3]). Also, 3D video acquisition has gained much attention recently ([8,24,26,27,47]). Although these approaches provide more information about the scene, a disadvantage is that often calibration of the system is needed, and the cost is higher than a system using regular cameras with fewer constraints of placement and overlap of view.

Table 1

Overview of events used in different studies in the literature compared to ours using real-life data.

Study	Number of falls	Other events	Amount of data	Type of data	Publicly available
Auvinet *et al.* [3], Hung *et al.* [18], Rougier *et al.* [46]	25	24	10 minutes over 24 fragments	Simulation	Yes
Charfi *et al.* [4]	192	57	± 125 minutes over 249 fragments	Simulation	Yes
Chua *et al.* [5]	21	30	na[a]	Simulation	No
Foroughi *et al.* [11]	375	875	na	Simulation	No
Hartmann *et al.* [15]	45	45	na	Simulation	No
Htike *et al.* [16]	43	30	na	Simulation	No
Miao *et al.* [31]	240	240	na	Simulation	No
Shieh *et al.* [50]	60	40	na	Simulation	No
Young *et al.* [23]	5	49	na	Simulation	No
Young *et al.* [24]	92	56	na	Simulation	No
Our real-life data set used in the present study	21	Virtually unlimited	504 hours used	Real life	No

[a] na: not available

Ozcan *et al.* [41] have proposed using a wearable camera. This approach has the potential of overcoming some of the challenges faced by fixed camera systems. Certainly, the problem of the subject falling out of the camera's view is solved in this case, but the major disadvantage is that it is not contactless, and so has the same disadvantages of other wearable systems as mentioned above.

Currently, the most frequently used approach for fixed camera-based fall detection reported in the literature is combining relatively simple, low-level cues with available domain knowledge (e.g., exploiting the fact that the speed of a person falling is mostly greater than his speed during normal activities, or the fact that the posture of a person changes during a fall, etc.). The use of more complex methods, like action recognition (e.g., [53]) and person detection (e.g., [61]), seems promising, but still needs further study and probably will require more training data to be reliable. Since the cameras are static, background subtraction can be applied to find the moving foreground objects, including a person. Likewise, one can build on domain knowledge to design simple yet robust fall features, such as the aspect ratio of the foreground region [1,4,6,11,20,24,25,32,60] , or the speed of the person's head [4,11,29,44,47] (exploiting the fact that the head remains mostly visible during a fall). Background subtraction, in which the background is modeled using previously visible information in the video and then the foreground is found by calculating the difference between the current image and this model, has been used by many systems (e.g., [1,3,5,6,11,15,16,18,20,23–26,29–32,38,44,45, 50,51,53,54,57,60]). In many cases, it is assumed that this results in an accurate silhouette of the person, based on which the person's posture can be determined (e.g., [1,5,6,11,16,20,23,25,26,31,38,50,51]). However, observations on real-life data show that an accurate silhouette is often unavailable. Due to the lower image quality, as well as problems with overexposure, occlusions, and changing illumination conditions, background subtraction only provides a rough idea of where the person might be (even after shadow removal). Also, the fact that older persons often remain seated at the same place over long periods of time does not help in this respect [7]. In this case, due to the nature of background subtraction, the person is sometimes unintentionally integrated into the background, which causes ghost figures when the person starts moving again.

Methods exploiting relatively low-level cues (e.g., [22,44,45,54]) seem most promising in a real-life context. They are robust, fast to compute, and relatively generic (i.e., no need for retraining or calibration for each new camera setup).

The new object-detection method proposed in this paper is based on background subtraction to find the person in the image. This will be described in detail in Section 4.1. From this image of the detected person, several low-level features are extracted and combined. Additionally, temporal information is included in the algorithm to detect a fall.

3. Video data

This section gives an overview of our real-life data set and other publicly available, simulated data sets. We compare and contrast the most important characteristics of these data sets.

3.1. *Real-life fall data set*

This study was approved by the Medical Ethics Committee of the Leuven University Hospitals in Leuven, Belgium. From 2009-2013, camera systems were installed at the residences of seven older persons: two in a house of community-dwelling older persons, one in an older person's private living room at his nursing home, and four in an assisted living room (ALR). All participants received oral and written information about the study and the camera system, and all gave their informed consent. The participants' age ranged from 74 to 95 years. For each residence, 4 to 6 wall-mounted, internet protocol (IP) cameras were used. They had a resolution of 640 by 480 pixels and a frame rate of 12 frames per second. Images were recorded continuously, at night, and in low light conditions using near-infrared lights. The cameras were installed only in the corners of the room, with the aim of covering the entire room in the camera's field of view. For privacy reasons, a camera was not installed in the bathrooms, except for the bathroom of one person, who insisted because he had fallen twice at that location previously. A control panel was provided to allow the participants to switch off the system whenever they desired.

The participants' fall risk was measured using the Timed Get-Up-and-Go Test (TGUGT) [21]. During this test, the subject is asked to rise from a chair, walk three meters, turn around, return to the chair, and sit down. The manually recorded time needed to complete the test and observations by the health-care worker (e.g., shuffling or unstable gait) are used to quantify fall risk. The TGUGT indicates that the subject has an increased fall risk when more than 14 seconds are needed to complete the test [21]. A result of over 20 seconds represents a high risk of falling. Based on this test, four participants had a high risk of falling, while one had an increased risk of falling. The two other subjects had a normal walking pattern, and little risk for falling, as assessed with the TGUGT.

Camera recordings were made 24 hours a day, 7 days a week. In total, over 21,000 hours of video were stored. The required storage for these data was over 180 TB. For privacy reasons, the recordings were always stored on a pc lacking internet access in the house of the participant. Every two weeks, the researchers exchanged the computers' filled hard drives for fresh ones to continue the recordings. During the video acquisition period, 29 falls were registered, and a huge amount of non-fall related data were recorded. Two persons fell a combined 25 times. Two participants never fell and were therefore excluded from the remainder of this study. More detailed information about the remaining participants can be found in Table 2.

To the best of our knowledge, the data we collected comprise a unique data set. Robinovitch *et al.* conducted an observational study on real-life falls, but only common areas (e.g., dining rooms, lounges, and hallways) were monitored [43]. In our study, on the other hand, falls were recorded in the actual living quarters of these persons. The videos of Robinovitch *et al.* were, to the best of our knowledge, also not used for automatic fall detection, only for researcher-viewed observations.

It was impractical to use the complete real-life data set of over 21,000 hours for this validation study. We decided instead to use a fragment of the 24 hours of video recorded before each fall. The fall was included at the end of the video fragment. The camera, on which the fall was best visible, was manually selected.

Several inclusion criteria were formulated that had to be met for the video to be included in this study. The first was that 24 hours of video before the fall had to be available. Additionally, the fall had to occur while no one else was present in the same room. Someone else was allowed to be present in another room of the house or another portion of the video to remain eligible. And finally, the person had to fall in the camera's field of view. Occlusions, as shown in the upper panel of Figure 2, were allowed. Thus, only 21 of the 29 falls recorded as part of the real-life data set were eligible for inclusion. One fall was excluded, because the 24 hours of video before the fall were lost due to a hard drive failure. Three falls were excluded, because another person was in the same room when the falls occurred. Note that, in real-world situations, the detection of such fall incidents is less critical, since someone is present to immediately offer help. Four other falls were excluded, because part of the fall took place beyond the camera's field of view.

Figure 1 gives an overview of all included falls truncated into groups of three informative video frames: one immediately before the fall, one during the fall, and one after the fall. Table 3 gives an overview of the

Table 2

Baseline characteristics of the participants that fell during the video acquisition period.

Participant	Age	Sex	Home	TGUGT results	Walking aid	Acquisition period	Fall incidents	Falls on camera	Falls fulfilling inclusion criteria
A	95	f	ALR	24 sec	walking frame	5 months	1	1	1
B	95	f	ALR	32 sec	rollator	7 months	12	10	6
C	85	f	RCR	27 sec	rollator	5 months	17	15	11
D	74	m	CD	11 sec	none	20 months	2	2	2
E	75	f	ALR	16 sec	walker, cane, none[a]	3 months	2	1	1

[a] Participant E switched between walker, cane, and not using a walking aid during the day.

(TGUGT: Timed Get-Up-and-Go Test; ALR: Assisted Living Room; RCR: Residential Care Room; CD: Community Dwelling)

most important fall characteristics. In 38% of the falls, a walking aid was being used. In 29% and 38% of the falls, the person was partially or completely occluded in the video after falling, respectively. Thus, in 67% of the falls, it was not possible to extract a complete and accurate silhouette of the person for fall detection. Furniture was shifted in 67% of the cases. The direction in which a fall happened, as viewed from a given camera position, was also diverse. In 67% of the falls, the person fell sideways as viewed from the camera angle, while in 33%, the person fell towards or away from the camera. Most of the falls started from an upright posture (76%), while in 14%, the person was sitting before falling. In one fall, the person was crouching near the bed, and in another, the person was bending over to pick up something from the floor. As mentioned before, it is most important to generate an alarm if the person is unable to get up. In 86% of the falls, the person remained laying on the floor for an extended time and had to be aided to get up again.

Although we manually selected fall incidents and camera viewpoints, a fair comparison could still be made, since the goal of this study was to evaluate the fall detection algorithm, not the complete system. Excluded falls might have been acceptable for analysis if more cameras had been available, or if they were placed differently, a more favorable angle may have been chosen. Even after the selection of the best camera view, however, the data set remained very challenging to analyze.

3.2. *Simulated fall data sets*

During our literature search, we discovered only two data sets that are publicly available to evaluate fall detection algorithms. These are provided in the papers of Auvinet *et al.* [2] and Charfi *et al.* [4]. Both data sets were recorded in a simulated environment, in which younger actors simulated both falls and other activities of daily living. The data set of Auvinet *et al.* used a network of eight calibrated cameras for video acquisition. The data set consists of 24 videos, ranging from 30 seconds to over 4 minutes. It contains 25 falls and 24 other events (11 crouching, 9 sitting, and 4 lying on a sofa). Some examples are shown in Figure 6. The data set of Charfi *et al.* used a single camera setup and consists of 249 videos, ranging from 10 seconds up to 45 seconds. One hundred and ninety-two videos contain falls, and 57 contain several normal activities, such as walking in different directions, sitting down, standing up, housekeeping, moving a chair, etc. Some selected characteristics of the falls for these two data sets and several others can be found in Table 1.

For our analysis of simulated falls, we chose the video data set of Auvinet *et al.* [2]. This data set was also used in three other studies [3,18,46], while, to our knowledge, the data set of Charfie *et al.* [4] has been used only once. The data set of Auvinet *et al.* comprises eight different views of the monitored room: four from the corners and four from a position in the middle of the walls. To mimic as closely as possible the real-life setup of our data set, we analyzed video only from the corner cameras. From this subset, we manually selected the video from the camera on which the fall was best visible.

3.3. *Comparison of data set characteristics*

Table 4 shows a more detailed summary of the characteristics of our real-life data set, along with those of the simulated data sets of Charfi *et al.* and Auvinet *et al.* Not all falls occur in the same way. They can occur while walking or picking something up, for example. Often the person falls forward but sometimes also backwards. These different types are contained in all data sets. To produce falls that are more diverse in simulated data sets, multiple participants can be used, but Auvinet *et al.* used only one person to simulate the

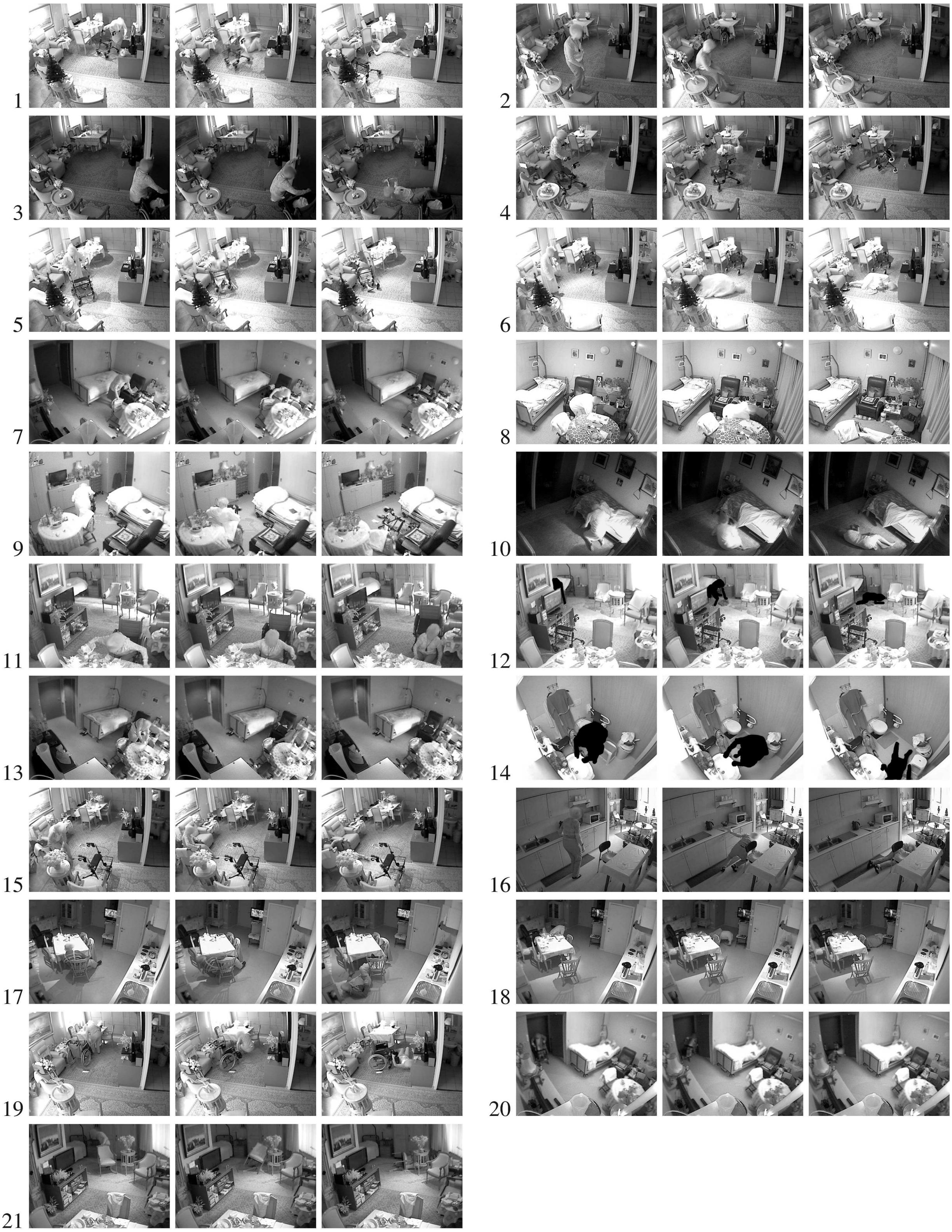

Fig. 1. Overview of all falls meeting the inclusion criteria in the real-life data set. The participants' heads are intentionally blurred in post-editing to maintain anonymity. For falls 12 and 14, only the silhouette of the person is shown to maintain anonymity.

Table 3

Overview of the most important characteristics of the included real-life falls. The last two columns show whether the fall was correctly detected using our algorithm for a given β-value[a].

Fall	Fall direction	Direction seen from camera	Walking aid used?	Occluded after fall	Furniture moved	Posture at start	Remained on floor	Fall detected $\beta = 20$	Fall detected $\beta = 40$
1	Backwards	Sideways	Rollator	No	Yes	Standing	Yes	Yes	Yes
2	Backwards	Sideways	No	Yes	Yes	Standing	Yes	No	Yes
3	Sideways	Sideways	Wheelchair	No	Short distance	Standing	Yes	Yes	Yes
4	Sideways	Away	Rollator	Yes	Rollator	Standing	Yes	No	Yes
5	Backwards	Away	Rollator	Yes	Yes	Standing	Yes	No	No
6	Frontal	Sideways	No	No	No	Standing	Yes	Yes	Yes
7	Sideways	Sideways	No	Yes	Yes	Standing	Yes	No	Yes
8	Sideways	Towards	No	Partially	Yes	Standing	Yes	Yes	Yes
9	Backwards	Sideways	Rollator	Partially	Yes	Standing	Yes	Yes	Yes
10	Sideways	Sideways	No	No	No	Crouching	Yes	Yes	Yes
11	Downwards	Towards	Wheelchair	Partially	Wheelchair	Sitting	Yes	No	No
12	Sideways	Sideways	No	No	No	Standing	No	No	No
13	Sideways	Sideways	No	Yes	Yes	Standing	Yes	No	No
14	Frontal	Towards	No	Partially	Yes	Bending over	Yes	No	Yes
15	Backwards	Sideways	No	Yes	Yes	Standing	Yes	Yes	Yes
16	Frontal	Sideways	No	Partially	No	Standing	Yes	Yes	Yes
17	Sideways	Sideways	No	No	No	Sitting	No	No	No
18	Frontal	Sideways	No	Partially	Yes	Sitting	No	No	No
19	Backwards	Towards	Wheelchair	No	No	Standing	Yes	No	Yes
20	Backwards	Away	Yes	Yes	No	Standing	Yes	No	No
21	Backwards	Sideways	No	Yes	Yes	Standing	Yes	No	No

[a] See Section 4.3

falls. Charfi *et al.* used multiple younger persons as actors. Our data set contained falls from five older persons, but as mentioned before, 86% of the incidents occurred in two people.

The real-life recordings showed that while falling, a person might shift the furniture, as shown in Figure 2. This can hinder the accurate detection of a fall. From the two simulated data sets, only Auvinet *et al.* took this into account in a small proportion of their falls. Additionally, our data set contained falls in which the person is (partly) occluded, or contained falls while the participant was using a walking aid, as shown at the bottom of Figure 2. During daytime, many different activities naturally occur, some of which can cause false alarms. However, in contrast to the real-life data set, both simulated data sets included only a limited number of different activities in (sometimes very) short fragments. In a real-life situation, the living quarters of a person changes constantly, for example, furniture shifting, doors and curtains opening and closing, and televisions being switched on or off. Also, the lighting conditions change during the day. These challenging conditions were included in the real-life data set, but are only meagerly present in the simulated data sets.

4. Methods

First, we provide an explanation about how a person is detected in the videos using our new algorithm. Then, we explain how the different features are extracted from the detected person. Finally, we show how these features are combined to determine whether a fall has occurred.

4.1. Person Detection

The videos were first converted to gray-scale images to standardize the processing of images acquired during the day and during the night when near-infrared light was used. The disadvantage of this conversion is that no color information is then available that might be used to distinguish features of the room from the person. The first step in the new algorithm involves identifying the person using foreground detec-

Fig. 2. Some challenges for camera-based fall detection. Upper panels: The table and chairs are pushed away while falling, and the upper body of the person is occluded at the end of the fall. Lower panels: While falling, the rollator walker is pushed and rolls away from the person.

Table 4

Most important characteristics of the different data sets

	Simulated data set of Charfi *et al.*	Simulated data set of Auvinet *et al.*	Our real-world data set
Falls	Several different kinds of falls	Includes falls against furniture	Different falls, against furniture, partly occluded, with walking aids
Normal activities	Some, but not lying down	Some, but also some that seem unnatural for older persons	Normal activities during complete day
Setting	Several different rooms; only few pieces of furniture present	Only one room, with only small amount of furniture	5 different houses with different rooms; most full of furniture
Participants	Several actors with different types of clothing	One actor wearing different shirts	5 older persons, even while changing clothes in view
Lighting	Different light sources, including (sun)light through window	Only one fixed light source	Several light sources; changing conditions during day
Video length	Very short (10-45 sec)	Rather short (30-60 sec), with only 2 longer videos (2-4 min)	Continuous (evaluation via fragments of 21 videos of 24 hours)
Furniture	Shifted chairs and some other small objects; also opening of doors	Moving sofas, chairs, and other objects, even during falls	Moving furniture, opening of doors, TV switched on, etc.
Camera perspective	Only one perspective; sometimes very close to fall itself	8 different fixed perspectives, similar to placement in real room	Fixed perspectives; always from upper corner of rooms
Occlusions	Occlusions present; sometimes person is only partially in view of camera	Occlusions caused by furniture	Numerous occlusions caused by furniture and person exiting view of camera

tion. This is accomplished through several steps, as shown in Figure 3. First, the foreground is segmented out from the background using background subtraction. After this, the shadow is removed and the remaining foreground is cleaned up. From this foreground, the region of interest (ROI) is determined.

4.1.1. Background Subtraction

First, the foreground is segmented out from the background. For this, a background subtraction technique is used based on an approximate median filter [28]. This technique approximates the median pixel intensity over all images acquired over a certain amount of time. Median filtering for background subtraction is based on the assumption that the background is visible more than half of the time. The advantages of using the approximate median filter are its low memory consumption, fast computation, and robustness. The drawbacks are its rather slow update in response to large changes in illumination, and as with any background subtraction method in which a dynamic background is present, the influence of the foreground on the background. This influence can lead to the appearance of a ghost figure in the video processing results (see Figure 8 f-h). When a person is sitting still on a couch for a long period, for example, the background updating process incorporates the person into the background. If he stands up, the region of the couch that was occluded previously will also differ from the background, and it is detected as foreground. This could influence the extraction of features used to detect a fall.

4.1.2. Shadow Removal

In video processing, a shadow that is cast by a moving object is falsely detected as being part of the foreground, since it makes the underlying pixels appear darker. This makes the detected foreground region larger than it should be. To remove this shadow, we use the property that a shadow only changes the intensity of the pixels while the texture of the covered region does not change [13]. As a result, the texture of the shadow is correlated with the corresponding texture of the background image. Jacques and Jung [17] describe the use of cross correlation to determine how the detected foreground pixels match the background pixels. With this approach, when the cross correlation is greater than a certain threshold, and the pixel is darker in the current image, then a pixel is classified as being part of the shadow. Also, other changes in illumination can be eliminated using this technique when removing the constraint that the pixel has to be darker in the current image.

Fig. 4. Extraction of salient subject features used for fall detection. Purple rectangle: bounding box of subject; white fine segmented line: best-fitting ellipse within subject bounding box; double green diamond: center of mass; blue octagon: head position. (Small, filled black rectangle is not part of feature extraction set; it is inserted in the figure presentation here to maintain subject anonymity).

4.1.3. Post-processing

The resulting foreground after shadow removal is sometimes still noisy. To reduce this noise, first an erosion/dilation step is used on all foreground pixels to remove small noisy patches. Next, a connected components analysis [48] is applied to determine all the foreground objects.

4.1.4. ROI Detection

The next step in our fall-detection algorithm is determining a ROI. In our case, the largest foreground region is selected and determined to correspond to the monitored person. To minimize noise and interference, the ROI must be greater than a certain threshold. A minimum threshold of 17,500 pixels resulted in the best performance. From this ROI, the features used to detect a fall are extracted from the image, as explained in the next subsection.

4.2. Fall Detection Features

For our algorithm, five features are extracted from an image in order to detect a fall: (1) aspect ratio (AR) [1,6,32,60]; (2) change in AR (CAR); (3) fall angle (FA) [45,60]; (4) center speed (CS) [45]; and (5) head speed (HS) [11,29] (see Figure 4). These features were chosen based on domain knowledge, i.e. in such a way that they captured relevant information to discriminate falls from other actions, while at the same time were sufficiently robust to avoid inaccuracies in person detection. These features are also the most widely used in other algorithms reported in the literature for fall detection in video images.

Aspect ratio The AR is calculated as the ratio of the width to the height of the bounding box (BB) around the foreground object (i.e., the subject). A per-

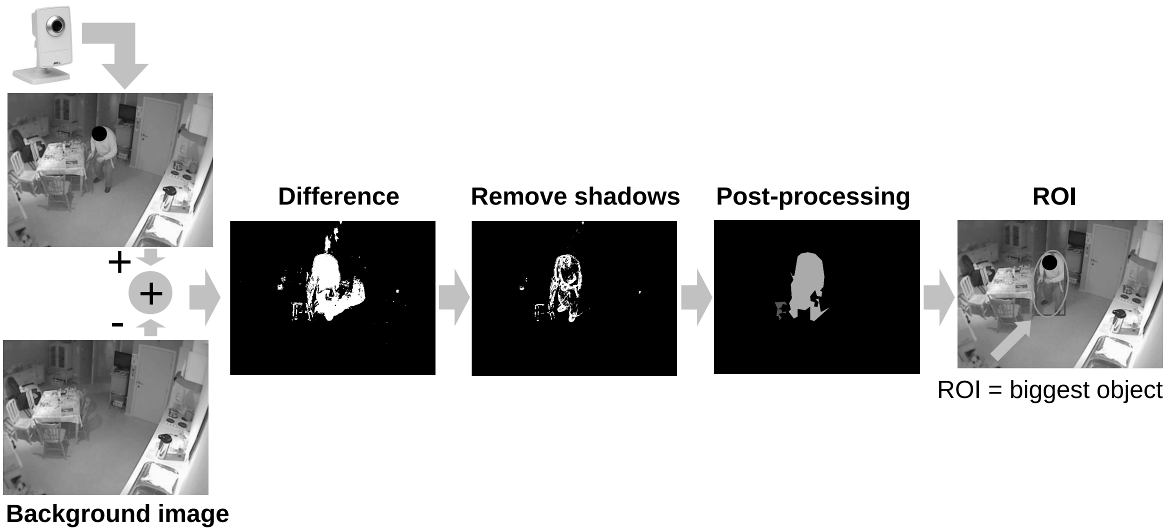

Fig. 3. Overview of the person-detection algorithm (ROI : region of interest).

son standing upright has a small AR, while a person lying down would have a large AR. Also, the change in AR (CAR) is used. This is calculated as the difference between the current AR and the AR calculated one second before the current AR.

Fall angle The angle of the person is measured as the angle between the long axis of the best-fitting ellipse and the horizontal direction in the image. A standing person has an angle of approximately ± 90 degrees, while a person lying down has an angle of approximately ± 0 degrees (from a side-view perspective). The fall angle is the difference of the current angle and the angle one second before the current angle.

Center speed and head speed A person, and certainly most older, frail persons, typically have low-paced movements. With most falls, however, a portion of the movement has relatively higher speeds. Based on this observation, the speed of the center of mass of the foreground object and the speed of the head defined as the highest point of the bounding ellipse as in [11] are key parameters for fall detection in our algorithm. The head is a more stable feature in the video images, since it is less prone to being occluded and less influenced by occlusions.

Feature vector Given that the selected features are based on domain knowledge, each of them can be used on their own as a basic fall detector by choosing an appropriate threshold (e.g., in [60]). However, better results can be obtained if they are combined in a single classifier. Most systems detect a fall in a single video frame. In some cases (e.g., [4]), a fall is only detected if a couple of adjacent frames are classified as "fall," reducing the number of false detections.

According to Noury *et al.* [40], a fall consists of four phases: the pre-fall, critical, post-fall, and recovery. Using this scheme, and considering even more data in the period just before the fall and certainly more in the period after the fall, could improve the detection. With this additional knowledge, a feature vector is constructed that contains information in the period in which a fall could be present with information in the period before and after this. Figure 5 shows the structure of this feature vector.

First for each video frame, the five fall detection features discussed in Section 4.2 are calculated. Note that the features are normalized to have a zero mean and unit standard deviation. Then, the complete video fragment is divided into time slots of 1 second. For each time slot, one feature vector is created. The maximum and mean value of each fall detection feature is calculated for each time slot. The features of the current time slot are stored along with those of the previous and the next slot. Additionally, other combinations, such as the combination of the previous time slot with the current and the next two slots, are used. These feature vectors are then used to detect a fall.

4.3. Fall Detection with SVM

To classify a time slot as containing a fall or non-fall, a Support Vector Machine (SVM) [58] is used. A SVM is a universal classifier that maximizes the mar-

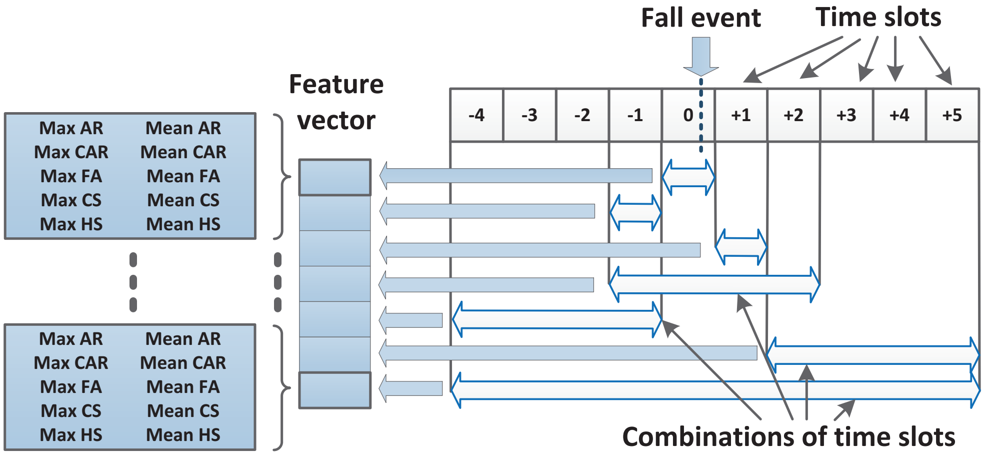

Fig. 5. Overview of the contents of the feature vector (FV) used by the Support Vector Machine (SVM; Section4.3). The complete video is split into discrete, one-second time slots. One FV is created for each time slot. A FV contains information about the current time slot and (combinations of) other time slots. Each FV part consists of 10 features as shown. (AR = Aspect ratio, CAR = Change of AR, FA = Fall angle, CS = Center speed, HS = Head speed).

gin between positive and negative data. Given the limited amount of fall-related training data, a linear kernel is chosen. To prevent classifying all vectors as non-fall, different weights are used for positive (w) and negative data $(1 - w)$. The detector is evaluated using ten-fold cross-validation. The available videos are partitioned into 10 different non-overlapping parts. This way there are 10 different test sets. With each test set, one training set corresponds, consisting of all videos that are not part of the specified test set. The best overall SVM model parameters are needed to train the ten SVMs of each training set. The parameters are (a) weight w and (b) regularization parameter of the SVM. A wide range of combinations of these parameters were tested. As a cost-function, the F_β-measure was used, which allows one to set appropriate weights to the sensitivity and positive predictive value (PPV). The sensitivity is the ratio of detected falls to all falls, while the PPV is the ratio of true alarms to all alarms. (TP: True positive; FN: False Negative; FP: False Positive)

$$Sensitivity = \frac{TP}{TP + FN}$$

$$PPV = \frac{TP}{TP + FP}$$

$$F_\beta = (1 + \beta^2).\frac{PPV.sensitivity}{\beta^2.PPV + sensitivity}$$

A β greater than one puts more weight on sensitivity than on PPV. The best combination of parameters is the one that gives the best mean performance over all ten training sets. Using this best combination, a model can be trained for each of the 10 training sets. This trained model can then be used to classify falls in its corresponding test set. This way, all time slots in all videos are classified as fall or non-fall just once. Also, different settings for β are tested to determine which setting gives the best result.

To reduce the false alarm rate, two post-processing steps are executed over the sequence of each video: median filtering and non-maximum suppression [39]. First, if a time slot is classified as a fall, but none of its neighbors are, then it is more probable that this is a false alarm and not a true fall. To reduce the number of these single detections, a median filter of length three is applied over the sequence. After this, non-maximum suppression with a window of three is used to combine the detections in single detections. Non-maximum suppression searches for a local maximum in the defined neighborhood. The detections adjacent to this maximum are set to zero, or in this case, to a non-fall. A time frame starting three seconds before the fall and ending three seconds after the fall is used to determine if a fall incident is truly detected. If at least one time slot inside of this time frame is classified as a fall, the fall incident is detected. This is counted as one TP. If no time slot is classified as a fall inside this time frame, this is one FN. Each detection outside of this time frame counts as one FP.

5. Results

Our fall detection algorithm was tested on a simulation data set that is publicly available and on a real-life data set we collected. First, the performance of our algorithm using the simulation data set is presented, along with the performance of other fall detection algorithms reported in the literature. Also, the results from using the separate features in our algorithm and the results from using a combination of all features are presented. Finally, we present the results of our fall detection algorithm using real-life data, along with the performance of the different parts of the algorithm.

5.1. Simulations

Table 5 shows the results of our fall detection algorithm together with those of other studies using the same simulation data set. A value of 2 for β in our algorithm produced the best overall performance for the F_β-measure. This compromise resulted in a sensitivity of 88% and a specificity of 95.6% for fall detection. Three (12%) falls in the simulation data were undetected with our algorithm, and five (4.4%) events were incorrectly identified as falls (i.e., false alarms). All false alarms occurred in just one video (video 23) of the simulated data set of 24 videos. The first missed fall went undetected, because the person fell onto a chair and stood up immediately after falling. Our system likely failed on this example, because it uses information from after the fall. This simulated fall differs from all other falls of the data set. In all other falls, the actor remains on the floor for several seconds after falling. For the second missed "fall," the actor was sitting on a chair and then dropped a bit lower onto the mattress. In the final missed fall, the actor fell and then slid over the floor for several meters. The false alarms produced by our algorithm were caused by errors in the foreground segmentation (3 alarms) or by misclassifications (2 alarms). In two cases, the actor was "split" into two parts in the foreground segmentation, causing erroneous features to be extracted. In one case, a chair was shifted close to the camera, which resulted in the chair being the largest foreground object rather than the actor walking on the other side of the room (see upper panel of Figure 6). The erroneous shift of the bounding box from the actor to this chair produced a false alarm. The two misclassifications were both from similar scenarios, in which the actor was simply laying down onto the sofa by letting himself "fall" into it (see lower panel of Figure 6).

Table 5

Performance of our proposed algorithm against others reported in the literature using the same simulation data set.

Method	Sensitivity	Specificity
Our method	88 %	95.6 %
Hung *et al.* [18]	95.8 %	100 %
Auvinet *et al.*[a] [3]	80.6 %	100 %
Rougier *et al.*[b] [46]	95.4 %	95.8 %
Our method with same annotations as used in [3][a]	82.6%	93.9%
Our method on same subset as used in [46][b]	90.9 %	100 %

[a] Auvinet *et al.* use a different annotation.

[b] Rougier *et al.* use a subset of the simulation data set.

Figure 7 presents receiver operating characteristic (ROC) curves, showing performance of the complete algorithm versus several the single feature classifiers, tested as baselines. The increase of performance of combining the different features and using time-informataion in our algorithm is evident in the ROC curves. First, the separate features with and without time information were used to classify the different time slots as a fall. Only the best performing features are shown in the figure. Then all the features were combined in one feature vector, again with and without time information. The performance of single features are shown when a threshold is applied and no time information is included; the SVM is not used, and test or training data were not used. In this case, the most optimal results for the different thresholds are shown. The combination of the features without time uses a test and training data set, in combination with the SVM. This explains why it seems to perform worse than with just the separate features.

The mean value of the aspect ratio performs the best of all of the separate features. This is followed closely by the mean value of the change of the AR. This was expected, as the AR is reported to be the best feature of other algorithms in the literature. The addition of time information increases the performance of all classifiers. Also, combining the different features increases performance of the algorithm. The complete classifier has the best overall performance, resulting in an area under the curve (AUC) of 0.821. The mean AR combined with time information is a close second, having an AUC of 0.798.

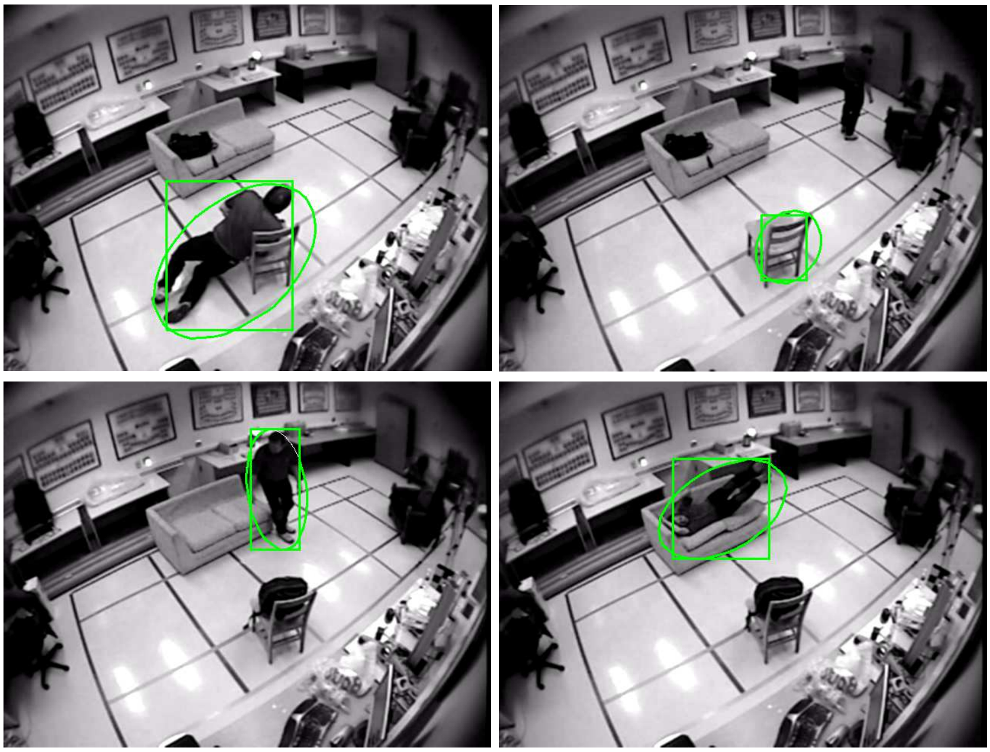

Fig. 6. Example of erroneous fall detections of the proposed algorithm using the simulation video data. Upper left: Actor falls on chair and gets up immediately. Upper right: Moved chair is largest foreground object. Lower panel: Swift "drop" onto sofa, annotated in the simulation data set as being normal activity.

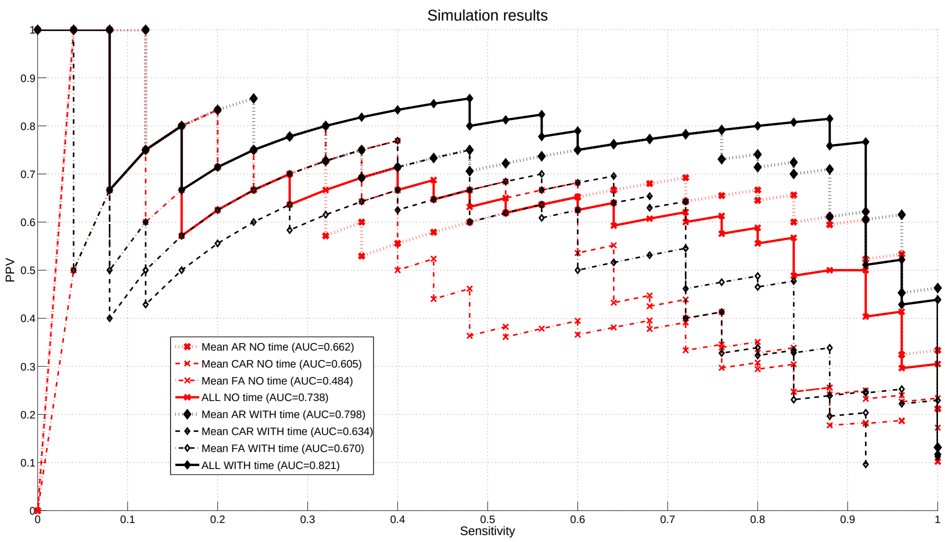

Fig. 7. ROC curves of tests on the simulation data using separate and combined features in the proposed algorithm. Performance with and without time information is considered. Only the best performing methods are shown.

5.2. Real-life data

Table 6 presents the results of our fall detection algorithm using the real-life data set we described earlier. The results of Auvinet et al., which used the simulation data set, are also included for comparison. Table 3 shows which falls exactly are detected for $\beta = 20$ and $\beta = 40$. The specificity is not presented in Table 6 for the real-life data set, because specificity is the ratio of true negatives divided by all non-fall activities. It is nearly impossible to determine the number of non-fall activities in the real-life data set of 504 hours, since the notion of "non-fall event" is ill defined. Additionally, even if it could be calculated, specificity is not the best measure to show the robustness of the system. A better measure for this is the PPV.

Since there was a huge amount of non-fall-related data and only a small number of falls in the real-life data set, the F_2-measure, as used in the simulations, put too much weight on the PPV. The result was a very low sensitivity. The tests using $\beta = 40$ produced a more or less acceptable sensitivity of 61.9% and a PPV of 0.35%. With this setting, 8 out of 21 (38%) falls were not detected. Five falls were missed because of the formation of a ghost figure in the video processing, which interfered with the correct extraction of the features. The presence of a walking aid in the images caused two falls to remain undetected: In one case, the person was occluded behind a rollator walker; in the other case, the person was partially occluded by a wheelchair that also generated a ghost figure. In five falls, the person was partially or completely occluded during or after the fall. In three falls, the person got up or was aided in getting up soon after falling. These circumstances, or a combination of them, prevented these falls from being correctly identified by the algorithm as a fall.

Also, 3753 false detections occurred, or $178.7(\pm 84.9)$ false alarms per day. These mainly happened during the daytime, when the person moved more frequently. Visual inspection of a random sample of 10% of these false detections showed that these had nine main causes. The most important causes are shown in Figure 8. In 18% of the errors, another foreground object with a similar size as the person was present, while in 16% of the cases another person was present in the room (see Figure 8 a). In 15% of the false alarms, the person's image consisted of two foreground objects of almost the same size. Because of these three causes, the system often erroneously switched to the other person or object, resulting in large motions and changes in aspect ratio and angle. Another 17% of the errors were caused by the continuous update of the background, which included non-moving persons in the background (see Figure 8 c-e). During this update, the size of the object changed, and sometimes the objects were split into several blobs. However, more than just people entering the background produced false alarms. Ten percent of the errors were caused by ghost figures that resulted from a person moving after being included in the background (see Figure 8 f-h). With a single camera system, a person moving through the room could become occluded behind furniture, again causing large changes of the bounding box and ellipse. This scenario accounted for 4% of the errors. A person could also become (partially) hidden by leaving the field of view, because the view angle of the cameras was limited. When the person closely approached the camera, or walked below it (as seen in Figure 8 b), a distortion of the bounding box and ellipse could be seen. This situation accounted for 13% of the errors. Another 5% of the errors was caused by changes in illumination (see Figure 8 i+j); shifted furniture was the origin of 2% of the false alarms.

Figure 9 again shows ROC curves of the different classifiers. Also, in the case of the real-life data set, the proposed algorithm outperforms its component parts considered alone. As mentioned above, we can see that performance using the real-life data set is a lot worse than that using the simulation data set. Only 38% of the falls can be detected before the PPV drops below 0.5%. This point corresponds with a $\beta = 20$.

6. Discussion

6.1. Simulations

As described in Section 5.1, our algorithm erroneously detected two activities as being falls that were annotated in the simulated data set as being a normal activity. These were two cases in which the person let himself "fall" into the sofa in order to lay down (see Figure 6 lower panels), as if he had just completed a hard day of work. These can be described as "intended" falls, but of course, could also be potentially dangerous for an older person. The three missed falls were mainly related to the fact that only one similar fall was present in the simulation data set. The use of a SVM-based fall detector, as for most other machine-learning approaches, has the constraint that it needs a sufficient amount of training data to learn about the possible range of detection activities. If a particular

Fig. 8. Examples of false alarms in the real-life video data set. a: More than one person present in room. b: Person leaves camera view. c-e: Person, remaining in same place for a long time, starts to disappear in background. c: Background, d: Binary foreground with detected bounding box, e: Foreground. f-h: Interference of ghost figure. f: Background, g: Binary foreground with detected bounding box, h: Foreground. i+j: Detected bounding box before (i) and after (j) switching on the lights (note that in this case, there is camera sensor saturation visible that cannot be solved by the cross correlation). (All heads are blurred to maintain anonymity.)

Table 6

Results of our proposed fall-detection algorithm using different values for β

Type of video data set	β	Time slot length	Sensitivity	PPV[a]	TP[b]	FN[c]	FP[d]
Auvinet *et al.*[e]	2	1 sec	0.88	0.82	22	3	5
Real life	2	1 sec	0.1905	0.0296	4	17	131
Real life	10	1 sec	0.2381	0.0217	5	16	225
Real life	20	1 sec	0.3810	0.0058	8	13	1360
Real life	40	1 sec	0.6190	0.0035	13	8	3753

[a] PPV: Positive Predictive Value
[b] TP: True Positive
[c] FN: False Negative
[d] FP: False Positive
[e] Results of our algorithm on simulation dataset added as reference.

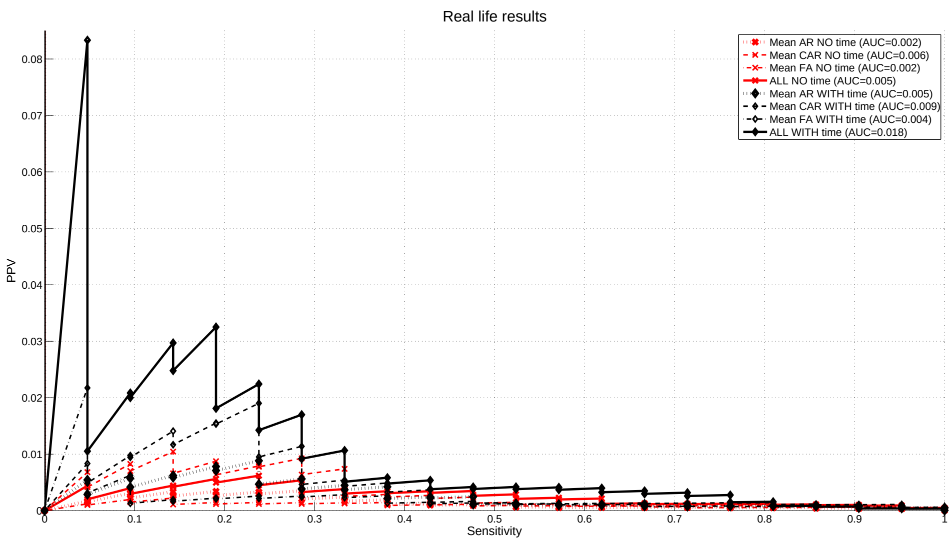

Fig. 9. ROC curves of tests using real-life data of individual and combined features, with and without time information added. Only the best performing methods are shown.

fall instance differs too much from the examples in the training set, it is possible that it will not be detected correctly.

Comparing the performance of our proposed algorithm using the simulation data set of Auvinet *et al.* with alternative algorithms described in the literature using the same data set (Table 5), we see that ours performed roughly similarly. Note that the other systems combine information from several calibrated cameras, while in our system only a single uncalibrated camera is used (the one on which the fall is best visible). Moreover, while several methods (e.g., [3,46]) report results on the same data set, it should be noted that they do not all use the same evaluation protocol, making it difficult to compare results across studies. This issue is detailed below.

In the detection system of Auvinet *et al.* [3], foreground detection results from multiple calibrated cameras are fused into a 3D volume, and a fall is detected whenever the largest part of this volume dips below 40 cm from the ground plane for at least five seconds. We report their results when merging information from three cameras. Using more cameras improves the performance of their system to a sensitivity and specificity of 100%. However, in their evaluation, they used different activity annotations than the ones provided in the description of the data set[2]. When using these same adjusted annotations of Auvinet *et al.*, our algorithm resulted in a sensitivity of 82.6% and a specificity of 93.9% on the simulated data set. This lower performance could be explained by the presence in the data of a few falls in which the person recovered quickly. Our algorithm detected these activities as falls, but they are not labeled as falls in their ground-truth annotations.

In the detection system of Hung *et al.* [18], two orthogonally placed calibrated cameras are used, which measure the height of the person. This parameter is then used to classify the person as standing, sitting, or lying down. A fall is detected if a person's state directly transitions from standing to lying, without first transitioning to sitting. This is why their algorithm did not detect the fall that began from a sitting position in the chair.

Rougier *et al.* [46] performed shape matching on silhouettes obtained by background subtraction to quantify shape deformation. Abnormal shape deformations followed by a period of one to five seconds of insignificant motion are detected as falls. Input from multiple cameras is integrated using majority voting. To validate their system, only the first 22 video fragments of the simulated data set were used. Using the same subset of data, our algorithm generated no false alarms and only two falls went undetected. In this case, the performance of our algorithm resulted in a sensitivity of 90.9% and a specificity of 100%, as shown in Table 5.

With these considerations, the performance of our proposed new algorithm could be considered to be similar to state-of-the-art performance of other algorithms.

6.2. Real life

Now, looking at the performance of our algorithm using the same settings on the real-life data set in Table 6, a completely different picture emerges. With this data set, our algorithm had a low detection rate and a high false alarm rate. One or two false alarms on a small data set of 20 minutes of video might seem acceptable. However, in a real-world situation, this rate translates to three or more false alarms per hour. Moreover, the more complex nature of real-life data caused a higher false alarm rate. Tuning the system to accommodate the amount of non-fall-related data by increasing β to 40 produced better results. Unfortunately, this adjustment was still not sufficient to produce a usable system. Most of the situations causing false alarms in the real-life data set were absent in the simulation data sets.

6.3. General Discussion

Unfortunately, it was impossible to test the three other published algorithms on the real-life data set. There is no implementation available for these, and even if there were, the system of Auvinet *et al.* needs at least three calibrated cameras with overlapping views. This is similar to the case for the Hung *et al.* system, which needs at least two orthogonally calibrated cameras. In our real-life data set, there are several rooms with only one camera, and even when two cameras are available, they are not orthogonally arranged or calibrated, and have only limited overlap.

The algorithm comparisons presented in section 5.1 ([3,18,46]), all start with background subtraction. It can thus be expected that all algorithms, including our own, are equally affected by the more challenging conditions found in real-life data. The use of real-life data certainly shows the importance of correctly validating the performance of the system and testing its robustness. Evaluation data should be representative of real-world situations, or even better, validations should use real-life data, like the present study did. The disadvantage is that the use of real-life data makes development of new algorithms more difficult. Moreover, it is difficult and expensive to collect real-life data.

Another problem arises with making real-life data publicly available, since it has several constraints, e.g., maintaining anonymity. Finding persons that want to cooperate in continuous recordings intended for use as a public data set is even more difficult than for a private data set. Anonymizing the data could make it easier to distribute the videos, but one problem is that certain fall-detection techniques rely critically on correctly identifying a person's head, which obviously will fail after applying current video anonymization procedures. It is clear that the use of simulation data offers several advantages and is still the best way to make progress now, but it is imperative to create simulation data sets that include a range of activities and situations closer to reality. For future work, we plan to produce and publically release a fall data set in which actors will reenact the falls from the real-life data set we presented and analyzed here. Most current algorithms using background subtraction work well if the videos are short and if the person does not remain in the same place for extensive periods of time. These restrictions are often only discovered after examining longer videos of real-world situations. An important point to take into account is the amount of data.

6.4. Possible improvements

Most of the false alarms produced by our algorithm could possibly be avoided by using more advanced techniques. Implementing more advanced foreground detection that is robust to continuous changes in illumination, slow movement of older persons, different types of light sources, and possible over-illumination, could improve performance. However, exploratory tests using two background subtraction algorithms available in OpenCV, an improved and adaptive mixture of a Gaussian model [63] and a probabilistic method that uses Bayesian inference [12], produced only minor improvements. Thus, using these advancements in this context does not justify the increase in processing time and memory consumption.

The largest improvement could be expected from the use of a tracking algorithm like e.g., a particle filter, to follow the person in the image instead of selecting the largest blob for each video frame. This avoids large motions and changes in appearance caused by jumping back and forth between different foreground blobs corresponding to different (parts of) persons or other objects. The integration of a person detector like e.g., [9] could also provide a means to verify the tracked object and to reinitialize the tracker.

7. Conclusion

Fall detection is becoming more and more important in efforts to ease the fears of an older person or someone with an increased fall risk. With reliable implementation, these persons are able to live longer independent lives in a more comfortable and safe way. In the present study, we introduced and evaluated a novel fall-detection algorithm comparable to state-of-the-art detection algorithms in terms of performance. Most importantly, we compared its performance on real-life data of older persons at their residences and publicly available simulation data. We showed that the system performs similarly as the current state-of-the-art systems on simulation data. However, when using real-life data, it becomes clear that it is not yet robust enough to employ in a real environment. Examining the detailed results produced by our system and the similarities in the preprocessing steps of the other algorithms reported in the literature suggests that their performance will also degrade similarly when used with real-life data. The available simulation data sets are a good start for developing fall-detection systems, but they are not yet close enough to reality. It is better to use real-life data, but the constraints in distributing them makes it impossible to use them for benchmarking different fall-detection algorithms. The creation of simulation data sets that include more real-life challenges and a more realistic ratios of fall to non-fall activities is important.

8. Acknowledgements

This work is funded by the "Fonds voor Wetenschappelijk Onderzoek" (FWO) via project G039811N: "Multi-camera human behavior monitoring and unusual event detection," by the "Agentschap voor Innovatie door Wetenschap en Technologie" IWT via "Technology Transfer" (TETRA) project 80150 "Fallcam: Detection of fall in older persons with a camera system," and by the EU via EraSME (FP7) project (IWT) 100404 "AMACS: Automatic Monitoring of Activities using Contactless Sensors." It is also funded by the Interdisciplinary Institute for Technology (iMinds) FallRisk project. The iMinds FallRisk project is co-funded by iMinds, a research institute founded by the Flemish Government. Companies and organizations involved in the project are COMmeto, Televic Healthcare, TP Vision, Verhaert, and Wit-Gele Kruis Limburg, with project support of IWT.

The authors thank the persons who participated in the research by giving their permission to be monitored over several months.

The authors would also like to acknowledge the Prevention of Falls Network for Dissemination (ProFouND) and the Cooperation in Science and Technology (COST) Algorithms, Architectures and Platforms for Enhanced Living Environments (AAPELE) network.

References

[1] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He, "Recognizing falls from silhouettes," in *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pp. 6388 –6391, 2006.

[2] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Multiple cameras fall dataset," *DIRO-Universite de Montreal, Tech. Rep*, vol. 1350, 2010.

[3] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier, "Fall detection with multiple cameras: An occlusion-resistant method based on 3-D silhouette vertical distribution," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 15, no. 2, pp. 290–300, 2011.

[4] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Definition and performance evaluation of a robust svm based fall detection solution," in *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*, pp. 218–224, 2012.

[5] J. L. Chua, Y. C. Chang, and W. K. Lim, "Intelligent visual based fall detection technique for home surveillance," in *Computer, Consumer and Control (IS3C), 2012 International Symposium on*, pp. 183–187, 2012.

[6] R. Cucchiara, A. Prati, and R. Vezzani, "An intelligent surveillance system for dangerous situation detection in home environments," *Intelligenza Artificiale*, vol. 1, no. 1, pp. 11–15, 2004.

[7] G. Debard, P. Karsmakers, M. Deschodt, E. Vlaeyen, E. Dejaeger, K. Milisen, T. Goedemé, B. Vanrumste, and T. Tuytelaars, "Camera-based fall detection on real world data," in *Outdoor and Large-Scale Real-World Scene Analysis*, vol. 7474 of *Lecture Notes in Computer Science*, pp. 356–375, Springer Berlin Heidelberg, 2012.

[8] R. Dubey, B. Ni, and P. Moulin, "A depth camera based fall recognition system for the elderly," in *Image Analysis and Recognition* (A. Campilho and M. Kamel, eds.), vol. 7325 of *Lecture Notes in Computer Science*, pp. 106–113, Springer Berlin Heidelberg, 2012.

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Anchorage, Alaska, June 2008.*, 2008.

[10] J. Fleming and C. Brayne, "Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90," *British Medicine Journal*, vol. 337, no. v17 1, p. 2227, 2008.

[11] H. Foroughi, B. Aski, and H. Pourreza, "Intelligent video surveillance for monitoring fall detection of elderly in home environments," in *Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference on*, pp. 219–224, 2008.

[12] A. Godbehere, A. Matsukawa and K. Goldberg, "Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation," in *American Control Conference (ACC), 2012*, pp. 4305–4312, 2012.

[13] D. Grest, J. Frahm, and R. Koch, "A color similarity measure for robust shadow removal in real-time," Vision, Modeling and Vizualization, 2003.

[14] K. A. Hartholt, N. van der Velde, C. W. N. Looman, E. M. M. van Lieshout, M. J. M. Panneman, E. F. van Beeck, P. Patka, and T. J. M. van der Cammen, "Trends in fall-related hospital admissions in older persons in the Netherlands," *Arch Intern Med*, vol. 170, no. 10, pp. 905–911, 2010.

[15] R. Hartmann, F. Al Machot, P. Mahr, and C. Bobda, "Camera-based system for tracking and position estimation of humans," in *Design and Architectures for Signal and Image Processing (DASIP), 2010 Conference on*, pp. 62–67, 2010.

[16] Z. Htike, S. Egerton, and K. Y. Chow, "A monocular view-invariant fall detection system for the elderly in assisted home environments," in *Intelligent Environments (IE), 2011 7th International Conference on*, pp. 40–46, 2011.

[17] J. C. S. Jacques and C. R. Jung, "Background subtraction and shadow detection in grayscale video sequences," The XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05), 2005.

[18] D. H. Hung and H. Saito, "Fall detection with two cameras based on occupied areas," in *Proc. of 18th Japan-Korea Joint Workshop on Frontier in Computer Vision*, pp. 33–39, 2012.

[19] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," *Real-time imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[20] M. Krekovic, P. Ceric, T. Dominko, M. Ilijas, K. Ivancic, V. Skolan, and J. Sarlija, "A method for real-time detection of human fall from video," in *MIPRO, 2012 Proceedings of the 35th International Convention*, pp. 1709–1712, 2012.

[21] J. Large, N. Gan, D. Basic, and N. Jennings, "Using the timed up and go test to stratify elderly inpatients at risk of falls," *Clinical rehabilitation*, vol. 20, no. 5, pp. 421–428, 2006.

[22] T. Lee and A. Mihailidis, "An intelligent emergency response system: preliminary development and testing of automated fall detection," *Journal of Telemedicine and Telecare*, vol. 11, no. 4, pp. 194–198, 2005.

[23] Y.-S. Lee and H. Lee, "Multiple object tracking for fall detection in real-time surveillance system," in *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, vol. 03, pp. 2308–2312, 2009.

[24] Y.-S. Lee and W.-Y. Chung, "Automated abnormal behavior detection for ubiquitous healthcare application in daytime and nighttime," in *Biomedical and Health Informatics (BHI), 2012 IEEE-EMBS International Conference on*, pp. 204–207, 2012.

[25] H. Liu and C. Zuo, "An improved algorithm of automatic fall detection," *AASRI Procedia*, vol. 1, no. 0, pp. 353 – 358, 2012. AASRI Conference on Computational Intelligence and Bioinformatics.

[26] K. Makantasis, E. Protopapadakis, A. Doulamis, L. Grammatikopoulos, and C. Stentoumis, "Monocular camera fall detection system exploiting 3d measures: A semi-supervised learning approach," in *Computer Vision - ECCV 2012. Workshops and Demonstrations*, vol. 7585 of *Lecture Notes in Computer Science*, pp. 81–90, Springer Berlin Heidelberg, 2012.

[27] G. Mastorakis and D. Makris, "Fall detection system using kinect's infrared sensor," *Journal of Real-Time Image Processing*, pp. 1–12, 2012.

[28] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, vol. 8, no. 3, pp. 187–193, 1995.

[29] Y. Miao, S. Naqvi, and J. Chambers, "Fall detection in the elderly by head tracking," in *Statistical Signal Processing, 2009. SSP '09. IEEE/SP 15th Workshop on*, pp. 357 –360, 2009.

[30] Y. Miao, A. Rhuma, S. Naqvi, and J. Chambers, "Fall detection for the elderly in a smart room by using an enhanced one class support vector machine," in *Digital Signal Processing (DSP), 2011 17th International Conference on*, pp. 1–6, 2011.

[31] Y. Miao, 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," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16, no. 6, pp. 1274–1286, 2012.

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

[33] K. Milisen, A. Geeraerts, and E. Dejaeger, "Use of a fall prevention practice guideline for community-dwelling older persons at risk for falling: a feasibility study," *Gerontology*, vol. 55,

no. 2, pp. 169–178, 2008.

[34] K. Milisen, E. Detroch, K. Bellens, T. Braes, K. Dierickx, W. Smeulders, S. Teughels, E. Dejaeger, S. Boonen, and W. Pelemans, “Falls among community-dwelling elderly: a pilot study of prevalence, circumstances and consequences in Flanders,” *Tijdschr Gerontol Geriatr*, vol. 35, no. 1, pp. 15–20, 2004.

[35] M. Mubashir, L. Shao, and L. Seed, “A survey on fall detection: Principles and approaches,” *Neurocomputing*, vol. 100, no. 0, pp. 144 – 152, 2013. Special issue: Behaviours in video.

[36] N. M. Nachreiner, M. J. Findorff, J. F. Wyman, and T. C. McCarthy, “Circumstances and consequences of falls in community-dwelling older women,” *Journal of Women’s Health*, vol. 16, no. 10, pp. 1437–1446, 2007.

[37] H. Nait-Charif and S. J. McKenna, “Activity summarisation and fall detection in a supportive home environment,” in *ICPR ’04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR’04) Volume 4*, (Washington, DC, USA), pp. 323–326, IEEE Computer Society, 2004.

[38] F. Nater, H. Grabner, and L. Van Gool, “Visual abnormal event detection for prolonged independent living,” in *International mobile health (mHealth) workshop*, 2010.

[39] A. Neubeck and L. Van Gool, “Efficient non-maximum suppression,” in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, pp. 850–855, IEEE, 2006.

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

[41] K. Ozcan, A.K. Mahabalagiri, M. Casares and S. Velipasalar, “Automatic fall detection and activity classification by a wearable embedded smart camera,” in *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 3, pp. 125–136, IEEE, 2013.

[42] M. Piccardi, “Background subtraction techniques: a review,” in *Systems, man and cybernetics, 2004 IEEE international conference on*, vol. 4, pp. 3099–3104, IEEE, 2004.

[43] S. N. Robinovitch, F. Feldman, Y. Yang, R. Schonnop, P. M. Leung, T. Sarraf, J. Sims-Gould, and M. Loughin, “Video capture of the circumstances of falls in elderly people residing in long-term care: an observational study,” *The Lancet*, vol. 381, no. 9860, pp. 47 – 54, 2013.

[44] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, “Monocular 3d head tracking to detect falls of elderly people,” Annual International Conference of the IEEE Engineering in Medicine and Biology Society., 2006.

[45] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, “Fall detection from human shape and motion history using video surveillance,” in *Advanced Information Networking and Applications Workshops, 2007, AINAW ’07. 21st International Conference on*, vol. 2, pp. 875–880, 2007.

[46] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, “Robust video surveillance for fall detection based on human shape deformation,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 5, pp. 611–622, 2011.

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

[48] H. Samet and M. Tamminen, “Efficient component labeling of images of arbitrary dimension represented by linear bintrees,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, pp. 579–586, Jul 1988.

[49] SeniorWatch, “Fall detector: Case study of European IST SeniorWatch Project,” tech. rep., SeniorWatch, 2001.

[50] W.-Y. Shieh and J.-C. Huang, “Falling-incident detection and throughput enhancement in a multi-camera video-surveillance system,” *Medical Engineering & Physics*, vol. 34, no. 7, pp. 954 – 963, 2012.

[51] M. Shoaib, R. Dragon, and J. Ostermann, “View-invariant fall detection for elderly in real home environment,” in *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pp. 52–57, 2010.

[52] C. Stauffer and W. E. L. Grimson, “Learning patterns of activity using real-time tracking,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 747–757, 2000.

[53] E. Syngelakis and J. Collomosse, “A bag of features approach to ambient fall detection for domestic elder-care,” in *Proc. Intl. Symp. on Ambient Technologies (AMBIENT 2011)*, 2011.

[54] N. Thome, S. Miguet, and S. Ambellouis, “A real-time, multiview fall detection system: A LHMM-based approach,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 11, pp. 1522–1532, 2008.

[55] M. E. Tinetti, “Preventing falls in elderly persons,” *New England Journal of Medicine*, vol. 348, no. 1, pp. 42–49, 2003.

[56] M. E. Tinetti, W. Lui, and E. Claus, “Predictors and prognosis of inability to get up after falls among elderly persons,” *JAMA*, vol. 269, pp. 65–70, 1993.

[57] B. U. Toreyin, Y. Dedeoglu, and A. E. Cetin, “HMM based falling person detection using both audio and video,” in *Computer Vision in Human-Computer Interaction, Proceedings*, vol. 3766 of *Lecture Notes in Computer Science*, pp. 211–220, Springer Verlag Berlin, 2005.

[58] V. Vapnik, *Statistical learning theory*. New York: Wiley, 1998.

[59] E. Vlaeyen, M. Deschodt, G. Debard, E. Dejaeger, S. Boonen, T. Goedemé, B. Vanrumste, and K. Milisen, “Fall incidents unraveled: a series of 26 video-based real-life fall events in three frail older persons,” *BMC geriatrics*, vol. 13, no. 1, p. 103, 2013.

[60] J. Willems, G. Debard, B. Vanrumste, and T. Goedemé, “A video-based algorithm for elderly fall detection,” Medical Physics and Biomedical Engineering World Congress, WC2009, 2009.

[61] Y. Yang and D. Ramanan, “Articulated pose estimation with flexible mixtures-of-parts,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1385 – 1392, 2011.

[62] G. Yongli, O. S. Yin, and P. Y. Han, “State of the art: a study on fall detection,” *Proceedings of World Academy of Science, Engineering and Technology*, no. 62, 2012.

[63] Z. Zivkovic, “Improved adaptive Gaussian mixture model for background subtraction,” in *Pattern Recognition, ICPR 2004. Proceedings of the 17th International Conference on*, pp. 28–31, IEEE, 2004.