Journal of Biomechanics 88 (2019) 25–32

Contents lists available at ScienceDirect

Journal of Biomechanics

journal homepage: www.elsevier.com/locate/jbiomech
www.JBiomech.com

Automated remote fall detection using impact features from video and audio

Evelien E. Geertsema [a,b], Gerhard H. Visser [a], Max A. Viergever [b,c], Stiliyan N. Kalitzin [a,b,*]

[a] Stichting Epilepsie Instellingen Nederland (SEIN), the Netherlands
[b] Image Sciences Institute, University Medical Center Utrecht, Utrecht, the Netherlands
[c] Utrecht University, Utrecht, the Netherlands

ARTICLE INFO

Article history:
Accepted 4 March 2019

Keywords:
Fall detection
Remote sensing
Pattern recognition
Video analysis

ABSTRACT

Elderly people and people with epilepsy may need assistance after falling, but may be unable to summon help due to injuries or impairment of consciousness. Several wearable fall detection devices have been developed, but these are not used by all people at risk. We present an automated analysis algorithm for remote detection of high impact falls, based on a physical model of a fall, aiming at universality and robustness. Candidate events are automatically detected and event features are used as classifier input. The algorithm uses vertical velocity and acceleration features from optical flow outputs, corrected for distance from the camera using moving object size estimation. A sound amplitude feature is used to increase detector specificity. We tested the performance and robustness of our trained algorithm using acted data from a public database and real life data with falls resulting from epilepsy and with daily life activities. Applying the trained algorithm to the acted dataset resulted in 90% sensitivity for detection of falls, with 92% specificity. In the real life data, six/nine falls were detected with a specificity of 99.7%; there is a plausible explanation for not detecting each of the falls missed. These results reflect the algorithm's robustness and confirms the feasibility of detecting falls using this algorithm.

1. Introduction

Like age-related health problems in elderly people, some seizure types in people with epilepsy cause loss of balance which may result in a fall (Krumholz and Hopp, 2008; Rubenstein, 2006). Falls can cause serious injuries, especially when there is high impact with the floor or some other hard surface (Russell-Jones and Shorvon, 1989). People who have fallen may be unable to summon help due to injuries or impairment of consciousness. Automated detection may help alert caregivers to seizure-related falls.

Several solutions have been proposed for automated detection of falls, often using accelerometers to measure movement patterns. Such devices need to be worn at all times, however, and this may be forgotten or not well tolerated by the subject. An alternative solution is remote fall detection.

Monocular video cameras can be used as sensitive, versatile and relatively cheap sensors to quantify movement, and automated online analysis of the video stream may enable remote detection of falls. Common features for detection include shape-related features such as changes in the person's shape (Belshaw et al., 2011; Goudelis et al., 2015; Hazelhoff et al., 2008; Mirmahboub et al., 2013; Rougier et al., 2011; Yu et al., 2013), bounding box dimension ratios (Alhimale et al., 2014; Liu et al., 2010; Miaou et al., 2006; Tao et al., 2005; Vishwakarma et al., 2007), and ellipsoids modelling posture changes (Debard et al., 2016; Feng et al., 2014; Foroughi et al., 2008). Motion pattern features such as velocity and motion energy of the individual segmented in the frame have also been used to detects falls (Huang et al., 2007; Khawandi et al., 2011; Liu et al., 2008; Olivieri et al., 2012; Rougier et al., 2006; Wu, 2000; Zhang et al., 2010). In many recent studies a combination of motion and shape related features has been used (Charfi et al., 2013; De Miguel et al., 2017; Debard et al., 2016, 2015, 2012; Feng et al., 2014; Liao et al., 2012; Makantasis et al., 2016; Senouci et al., 2016; Yun and Gu, 2016; Zweng et al., 2010). Other methods to detect falls in video recordings include inactivity detection (Nait-Charif and McKenna, 2004) or anomaly/unusual event detection (Nater et al., 2009; Zerrouki et al., 2016).

Remote fall detection has also been performed using audio analysis (Li et al., 2012; Popescu et al., 2008; Salman Khan et al., 2015), and combining video and audio features could potentially increase

* Corresponding author at: Achterweg 5, Heemstede, the Netherlands (Visiting address) and Postbus 540, 2130 AM Hoofddorp, the Netherlands (Mailing address).
E-mail address: skalitzin@sein.nl (S.N. Kalitzin).

https://doi.org/10.1016/j.jbiomech.2019.03.007
0021-9290/

detection performance. Video and audio classifiers have previously been combined, detecting a fall only if both classifiers provided a high fall probability (Toreyin et al., 2005).

There is currently no working system for remote fall detection that has shown good detection performance on real-life data. Promising performance results have been reported in benchmark video datasets with acted falls, with sensitivity and specificity values >95% (Charfi et al., 2013; De Miguel et al., 2017; Fan et al., 2017; Feng et al., 2014; Senouci et al., 2016; Wang et al., 2016; Yun and Gu, 2016). The generalizability of these results is questionable, as results on new data not used for training and, particularly, on real-life data are often lacking. When performed, application of trained algorithms on new data shows more modest performance results (Debard et al., 2016, 2012; Fan et al., 2017). It has been suggested that real-life falls differ from acted falls, for example in terms of impact (Kangas et al., 2012). To our knowledge, there exist no studies on remote detection of seizure-related falls, which may differ from acted falls, or falls not resulting from seizures.

We propose an algorithm for remote detection of high impact falls, aiming for universality (applicability to other vision-based sensors) and robustness to new data and real-life epilepsy-related falls. We used half of a benchmark dataset with acted falls and normal activities to train the algorithm. We then tested the trained algorithm's performance and robustness on the other half of the benchmark dataset and on a newly collected test set containing real-life falls resulting from seizures.

2. Materials and methods

Rule-based analysis of video and audio signals provides us with possible fall events. Unlike most fall detection algorithms that convert the entire time series to signal features for machine learning-based classification, our algorithm uses only signal features at the time of possible fall events. In this way we use prior knowledge of event properties to reduce the quantity of observations to classify. The algorithm does not require segmentation of the person in the video frames, but uses vertical velocity and acceleration analysis from optical flow outputs, complemented with a sound amplitude feature for increased detector specificity. The algorithm consists of three parts: preprocessing of the video and audio signals; event detection; and event classification. An overview of the algorithm is shown in Fig. 1. All calculations were performed in Matlab (version 2017a, Mathworks Inc., Natic, USA).

2.1. Datasets

We used two datasets for the development and testing of the fall detection algorithm; the publicly available Le2i fall detection database (Charfi et al., 2013) and the SEIN fall database, a video database of recordings of genuine falls from people with epilepsy, collected at our center. Recording information is summarized in Table 1.

The Le2i database contains 221 videos simulated by actors, with falls in all directions, various normal activities and challenges such as variable illumination and occlusions. Videos were recorded from four room settings; 'coffee room', 'home', 'lecture room', and 'office', with 320 × 240 pixel resolution and a frame rate of 25 frames per second (fps). A subset of this database ('Office2') was excluded for lack of an audio stream in the video files. The remaining dataset (190 videos) was split in two randomly, providing a training set and a test set.

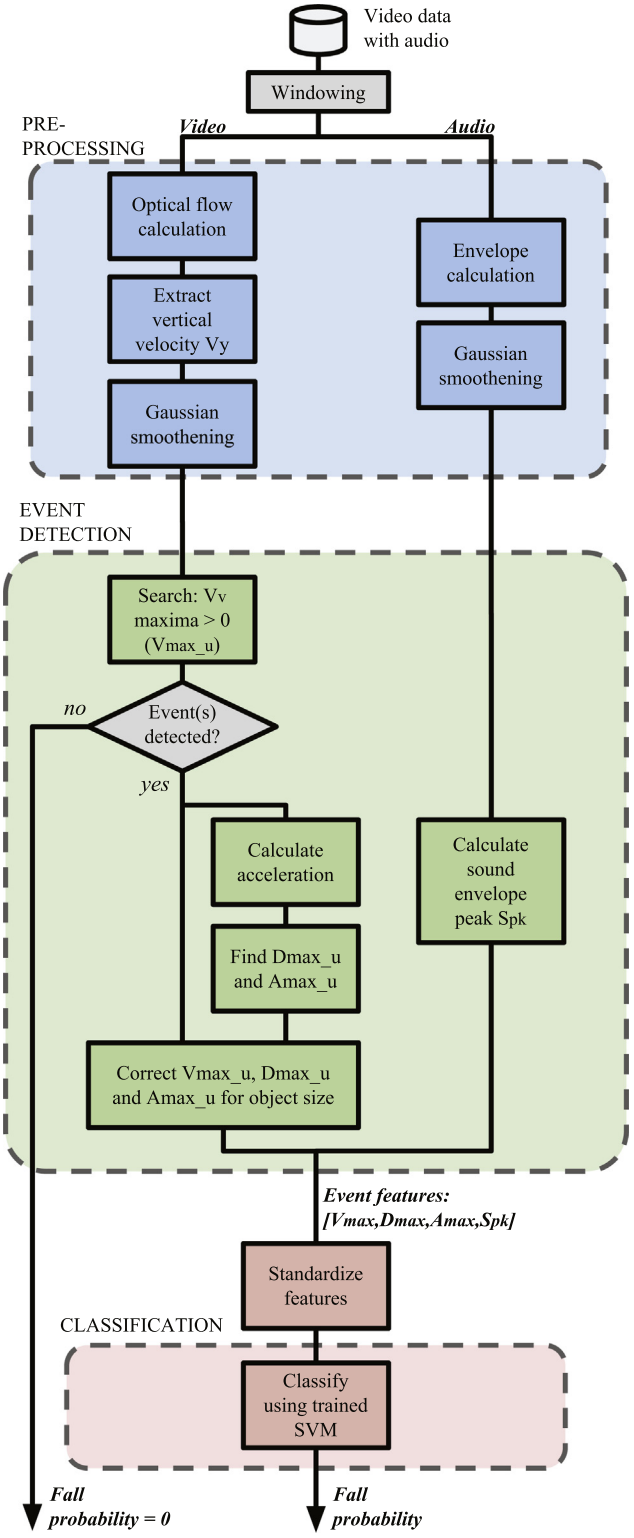

Fig. 1. Fall detection algorithm workflow. Video and audio streams are analyzed separately to provide signal features for events. Event features are only calculated when an event is detected, i.e. when a negative vertical velocity peak is found. Standardization and SVM parameters (red boxes) are obtained with algorithm training. V_y: vertical velocity, V_v: smoothed vertical velocity, V_{max}: positive maximum vertical velocity, D_{max}: deceleration maximum, A_{max}: acceleration maximum (_u for features uncorrected for moving object size), S_{pk}: sound envelope peak, SVM: support vector machine. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Used datasets for training and testing the fall detection algorithm.

Dataset	Subjects (N)	N videos	N falls	Total video length (h)	Mean video length (s) [range]	
Le2i training set	Actors (9)	94	65	0.41	15.7	[5.3–78]
Le2i test set	Actors (9)	96	65	0.44	16.5	[5.6–64]
SEIN test set	Patients (9)	9	9	1.45	578	[11–1600]

The SEIN fall database was selected from a historical video database containing clinical recordings of people who fell because of a seizure. Only videos with the fall happening in view of the camera were included. Five/nine videos were collected at an epilepsy monitoring unit (EMU), had a 1008 × 535 pixel resolution and an audio sampling frequency of 48,000 Hz. EMU staff were able to control camera pan, zoom and tilt. Although EMU-recorded videos do not necessarily represent our monitoring-situation - in which the subject is unattended, and not monitored real-time by another person - they enable estimation of detection sensitivity and provide insight into causes for false detections. The remaining four videos came from several neurologists' personal seizure video databases, had a 352 × 288 pixel resolution and an audio sampling frequency of 44,100 Hz. All videos had a frame rate of 25 fps. The need for written informed consent from the individuals in the videos was waived by our institutional ethics committee.

Fall start and end times in the Le2i dataset were annotated by the makers of the dataset. The time of impact with the floor was annotated for falls in the training set by EG (one of the authors). Fall start and end times in the SEIN fall database were also annotated by EG.

2.2. Preprocessing

Video-audio data in the test sets is windowed to be able to count detector true negatives and estimate fall detection performance realistically. We applied a three-second calculation window shifted with one-second steps (two-second overlap) to obtain algorithm output every second.

Optical flow and subsequent vertical velocity calculations provide subject velocity estimates without having to first segment the subject from the video frames, promoting computational lightness. Choosing this preprocessing step also allows combining a seizure detection algorithm previously developed by our group (Kalitzin et al., 2012) with the fall detection algorithm proposed in this paper into one detection system, without adding much extra computational cost.

Video stream optical flow was estimated using the Horn-Schunck method (Horn and Schunck, 1981) implemented as standard in Matlab, where the velocity vector field is obtained from the intensity field $L(x, y, t)$ as a function of the 2-D spatial coordinates (x, y) and the time t:

$$L(x, y, t) \rightarrow V(x, y, t) = \{V_x(x, y, t), V_y(x, y, t)\} \quad (1)$$

For further calculations we used only the vertical velocity time signal $V_y(t)$, defined as

$$V_y(t) \equiv \langle V_y(x, y, t)\rangle_{x,y} \quad (2)$$

$V_y(t)$ was smoothed along the time samples, or frames, using a one-dimensional Gaussian filter with its scale parameter set to 4 samples (4/25 s), obtaining the smoothened vertical velocity $V_v(t)$. The scale parameter setting was fine-tuned to this value during classifier training (see Section 2.5.). Additionally, vertical acceleration $A_v(t)$ was calculated by taking the first derivative of the smoothed vertical velocity signal $V_v(t)$. For the purpose of this study we selected the downward direction of V_v to be positive.

Audio was preprocessed to provide a smooth amplitude envelope. From the stereo sound recordings, we used only the first audio channel. The signal envelope was calculated using the Hilbert transform and smoothed using a one-dimensional Gaussian filter with its scale parameter set to 0.1 s. This filter scale setting was also fine-tuned during classifier training (see Section 2.5.).

2.3. Event detection and features

To detect possible fall events, we used the following model of a fall, illustrated schematically in Fig. 2. Once an individual starts to fall, downward velocity quickly increases and suddenly decreases again when the individual hits the ground. The quick increase and decrease in velocity are reflected in the maximum downward acceleration A_{max} and deceleration D_{max}. Between these two extrema is the point of maximum vertical downward velocity V_{max}. The impact of the person with the floor is accompanied by a sound. An event is detected when an $A_{max}, V_{max}, D_{max}$ sequence, and an accompanying sound, are found within a calculation window.

To calculate the video event features, we first find positive velocity maxima and for each find its leading acceleration maximum and following deceleration maximum. Feature values are subsequently corrected for distance to the camera. Uncorrected maxima (V_{max_u}, A_{max_u}, D_{max_u}) are calculated at corresponding time points (t_V, t_A, t_D) for each calculation window (w). We define for any function $f(t)$ of the discrete time variable t, the set of time points T_f^w of its positive local maxima within w as:

$$T_f^w = \{t \in w | f(t) > \max(f(t-1), f(t+1)); \quad f(t) > 0\} \quad (3)$$

where $f(t)$ can be replaced with $V_v(t)$ to obtain T_V, with $A_v(t)$ to obtain T_A, and with $-A_v(t)$ to obtain T_D. Only events with a consecutive $t_A < t_V < t_D$ sequence were considered eligible for a fall. The uncorrected features for each remaining event are:

$$A_{max\_u} \equiv A_v(t_A) \quad (4)$$

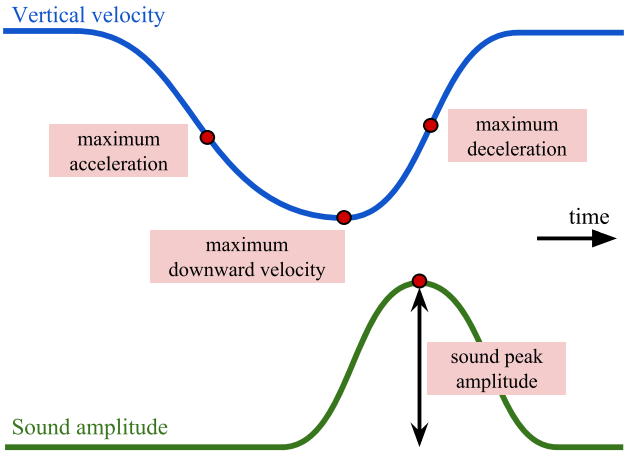

Fig. 2. Schematic illustration of the fall model used.

$$V_{max\_u} \equiv V_v(t_V) \tag{5}$$

$$D_{max\_u} \equiv -A_v(t_D). \tag{6}$$

The values of our video features depend on both the actual size of the object and its (focal) distance to the camera. Assuming that the only motion signal being observed is that of the subject, we can correct our video features for subject-camera distance or zoom angle changes using the size of the image area representing the velocity field of the subject. We can approximate the size of the image-footprint of a moving object from the optical flow velocity field at each time point t, using the determinant of the second moment tensor in the squared field's centroid. First, we spatially smoothed the magnitude of the velocity field $\|V(x,y)\|$, using a two-dimensional Gaussian filter with its scale parameter set to 10 pixels, obtaining the smoothened field magnitude $V_{Ms}(x,y)$. For further calculations we used the squared smoothened magnitude of the velocity field

$$W(x,y) \equiv {V_{Ms}}^2(x,y) \tag{7}$$

emphasizing large-amplitude movements that are presumably the subject's. The total amount of movement at a given time can be quantified by the sum $N_w = \sum_{x,y} W(x,y)$ which will be used as a normalization factor considering (7) as a distribution density. The centroid location (x_c, y_c) of the squared velocity field can be found using:

$$x_c \equiv \frac{1}{N_w}\sum_{x,y} xW(x,y); \quad y_c \equiv \frac{1}{N_w}\sum_{x,y} yW(x,y) \tag{8}$$

The second moment tensor describes the spatial distribution of the velocity field around the centroid and is defined as

$$M \equiv \frac{1}{N_w}\begin{bmatrix} \sum_{x,y}(x-x_c)(x-x_c)W(x,y) & \sum_{x,y}(x-x_c)(y-y_c)W(x,y) \\ \sum_{x,y}(x-x_c)(y-y_c)W(x,y) & \sum_{x,y}(y-y_c)(y-y_c)W(x,y) \end{bmatrix} \tag{9}$$

We can then approximate the moving object area in the field with

$$Area \sim Q \equiv \frac{\sqrt{\det(M)}}{N} \tag{10}$$

where N is the number of pixels. A correction factor was obtained for each event, by averaging the calculated areas over time from the acceleration maximum to the velocity maximum:

$$C_{event} \equiv \langle Q \rangle_{t\in[t_A,t_V]} \tag{11}$$

The three event velocity features can then be corrected as follows:

$$A_{max} \equiv A_{max\_u}/C_{event} \tag{12}$$

$$V_{max} \equiv V_{max\_u}/C_{event} \tag{13}$$

$$D_{max} \equiv D_{max\_u}/C_{event}. \tag{14}$$

For time windows with an event we obtained a fourth feature, based on the sound peak during the event. The window's relative peak amplitude (S_{pk}) was calculated using the smoothed envelope's (S_{env}) maximum and minimum:

$$S_{pk}(w) \equiv \frac{\max(S_{env})}{\min(S_{env})} \tag{15}$$

If a window contained multiple events, these received the same sound feature S_{pk}.

2.4. Classification

We classified events using a support vector machine (SVM) with a radial basis function (rbf) kernel. The training procedure that provided the SVM hyperparameters and model parameters is described in Section 2.5. In each three-second analysis window zero, one, or multiple events may be detected. The V_v fall waveform with corresponding acceleration extrema and sound peak need to occur in the same calculation window for them to constitute a detection. To prevent double detections of a single event owing to overlapping analysis windows, we define that t_V is between t = 1 s and t = 2 s of its corresponding window. If an event was detected, its features were passed to the classifier. The resulting posterior probability was stored as the classifier output for the window. If the window contained multiple events, all were passed separately to the classifier and the maximum posterior probability was stored as output. If no event was present in the analysis window, classifier output was set to zero.

2.5. Algorithm settings and classifier training

Algorithm settings were optimized in the following order: (1) SVM hyperparameters, (2) Gaussian filter scale parameters (used in audio and video preprocessing), (3) feature set standardization parameters. Settings optimization was followed by (4) classifier training, which provided the SVM parameters. In all four steps of this procedure we used only the training set in a ten-fold cross-validation scheme.

The classifier was trained to separate fall events from non-fall events based on event features, aiming for high sensitivity for falls. A sub selection of events was used for training; one event per fall, and a maximum of five non-fall events (selected randomly) per video. The number of non-fall events per video was limited to prevent over-representation of specific situations of a registration in the dataset. To select the right fall event for training (there may be multiple instances of V_{max}), we assumed that the event with the largest velocity within the [impacttime -1 s, impacttime $+1$ s] timeframe indicates the fall. In total, the training set contained 370 events, of which 94 were labeled 'fall'.

We tuned the SVM hyperparameters; the soft margin constant, regulating misclassification penalty, and the rbf kernel scale, determining model flexibility. Tuning aimed to optimize precision and accuracy at 90% sensitivity. To favor high sensitivity over accuracy, misclassification cost for false negative detections was increased until 90% sensitivity (in the cross-validation) was achieved. This process resulted in a subset of suitable hyperparameter settings, as no clear optimum was found. From this subset, we chose the hyperparameter settings that provided the most robust SVM parameters, minimizing variance over the folds.

Next, we fine-tuned the scale parameters for the Gaussian filters used to smooth the vertical velocity and sound envelope signals. The filter scale parameters were originally chosen visually to distinguish signals during falls. Both filter scale parameters were finetuned by optimizing precision and accuracy at 90% sensitivity in the training set.

The feature set was transformed logarithmically and standardized to zero mean and unit standard deviation. The standardization parameters were saved to standardize future observations.

Finally, the SVM was trained on the entire standardized training set, again using a 10-fold cross-validation scheme. Other possible sub-selections of features, classification models, and SVM kernels were also explored, but showed worse performance and were thus pursued no further.

2.6. Performance analysis

Performance was analyzed in the (so far unused) Le2i test set and the SEIN test set. A fall was detected (true positive, TP) when one of the windows overlapping with fall start-end annotations received supra-threshold classifier output. If none of the windows overlapping with the fall annotation received supra-threshold output, the fall was not detected (false negative, FN). Each non-fall window wrongly classified as a fall provided a false positive detection (FP). Non-fall windows correctly classified provided true negative (TN) detections. Detector sensitivity and specificity were calculated using detection counts $N_{...}$:

$$\text{sensitivity} = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{16}$$

$$\text{specificity} = \frac{N_{TN}}{N_{FP} + N_{TN}} \tag{17}$$

A receiver operating characteristics (ROC) curve for the Le2i test set, plotting sensitivity and specificity for different posterior probability thresholds, provided: ROC area under the curve (AUC), and specificity (SPEC) and positive predictive values (PPV) in different ROC working points. We also analyzed the distribution of false positive fraction (FPF) values for individual recordings at 90% sensitivity, defined as the fraction of non-fall windows wrongly classified as a fall. Performance in the SEIN test set was analyzed qualitatively to provide directions for use of the algorithm in practice.

3. Results

We first compared fall V_v waveforms observed in both test sets to assess the feasibility of applying the trained algorithm to the SEIN test set data, with falls that are possibly different. The results in Fig. 3 show that fall waveforms in the SEIN test set are similar to those in the Le2i test set. Smaller median V_v amplitudes were observed in the SEIN test set. This could be caused by an overall larger distance of the subjects from the camera. Because differences between amplitudes of both test sets remained after object size correction (results not shown), we surmise that velocities of the real-life falls in the SEIN test set are smaller than those in experimental falls, as has also been described in (Kangas et al., 2012).

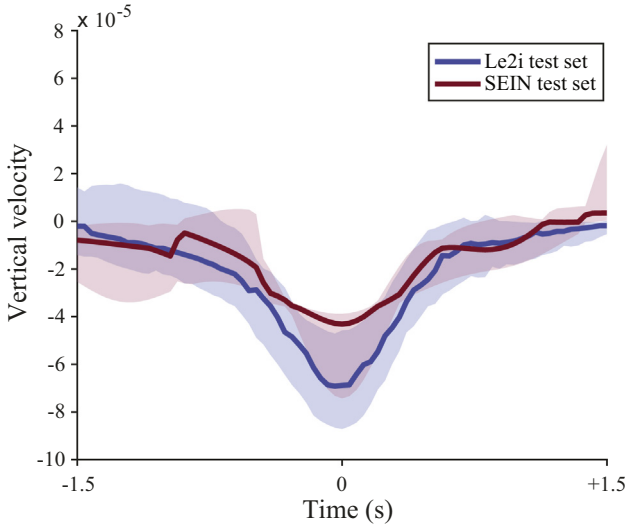

Fig. 3. Median fall V_v waveforms (solid lines) with interquartile ranges (shaded areas), for the Le2i test set (N = 65 falls) and SEIN test set (N = 6 falls with fall waveform). Fall waveforms were synchronized by putting all fall V_v minima at t = 0.

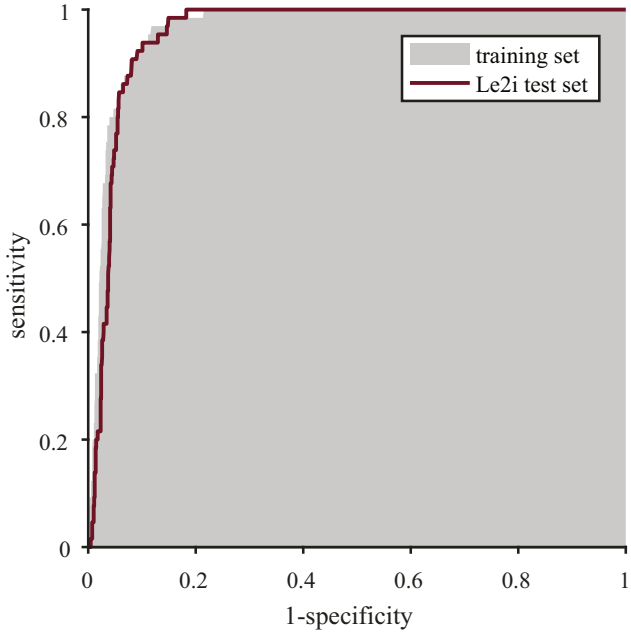

Fig. 4. Receiver operating characteristics (ROC) curve of fall detection algorithm testing results for the Le2i test set. Training set performance is shown in the background for reference. ROC curves were constructed using different classifier posterior probability thresholds.

Applying the trained classifier to the Le2i test set resulted in the ROC curve shown in Fig. 4. Results for different ROC curve working points are summed up in Table 2. Le2i test set results when using a feature set without sound are also shown, to illustrate the added value of the sound feature. FPF distributions for individual Le2i test set recordings obtained with the posterior probability threshold corresponding to 90% sensitivity, are shown in Fig. 5. Most of the videos (61%) show zero false positives, encompassing 43% of the total length of the dataset. In 75% of videos, accounting for 71% of total dataset length, a low FPF < 0.1 was found. Two relatively short videos had a high FPF > 0.3. With a small number of non-fall windows the FP fraction easily becomes high. Most FPs in the Le2i test set occurred when chairs were put down on hard flooring, and video and audio features resembled values of a fall.

Applying the detection threshold that provided 90% sensitivity in the training set in the SEIN test set resulted in detection of six/nine falls. Inspection of the videos and V_v waveforms provided explanations for not detecting these events. Post-hoc analysis of the data revealed that only one of three cases presents a critical and realistic situation that would result in a missed fall. In this case, after taking a few quick steps toward the back of the room in an attempt to regain balance, the patient still falls, which might result in injury. The negative V_v from the quick upward movement (due to camera placement high on the wall) cancelled out the fall movement. In the second case, camera movement (tilt, controlled by staff) impeded detection by cancelling out the fall movement. Camera positions would, however, be static in the intended monitoring system. Also, staff-controlled camera movements imply staff observing the fall. In the third case, a caregiver caught the falling subject before he hit the ground, implying a non-fall. In summary, our algorithm would have missed only one out of seven critical falls presenting realistic situations, which results in 86% sensitivity.

Thirteen FPs in total occurred in the SEIN test set, resulting in an overall FPF of 0.0025 (SPEC = 0.997, PPV = 0.32) in the SEIN test set. In four/nine videos there were no false positives. Two FPs were caused by camera movements (controlled by staff), the other 9 FPs occurred during the presence of extra people (caregivers) in view.

Table 2
Fall detection performance results for the Le2i test set. Results from using the full feature set, and for only video features are shown. Specificity (SPEC) and positive predictive values (PPV) are given for three working points on the ROC curves chosen according to their sensitivity values. ROC AUC: receiver operating characteristic area under the curve.

	ROC AUC	100% sensitivity		90% sensitivity		80% sensitivity	
		SPEC	PPV	SPEC	PPV	SPEC	PPV
Video & sound	0.957	0.818	0.248	0.919	0.401	0.945	0.468
Only video	0.947	0.799	0.231	0.896	0.345	0.923	0.385

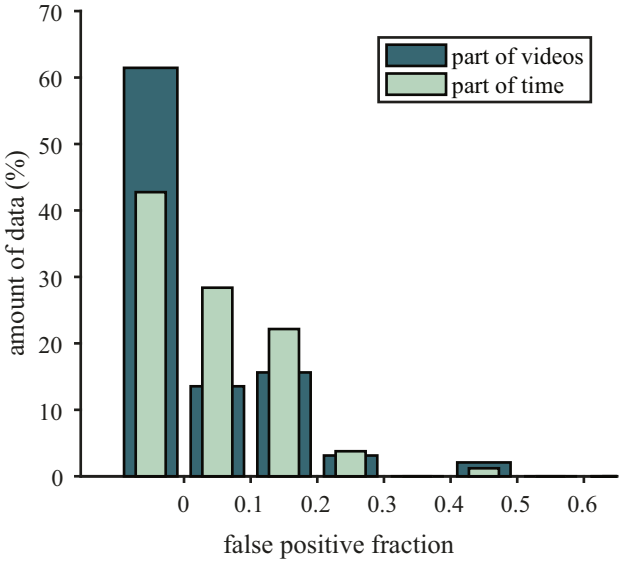

Fig. 5. False positive fraction (FPF) histograms obtained using the 90% sensitivity posterior probability threshold in the Le2i test set. The dark green histograms show the percentage of video registrations. The amount of time encompassed by the subset of videos within this FPF range is given in light green. The first bin of the histogram holds only videos without any false detections. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Discussion

Our fall detection algorithm, trained on a benchmark dataset, is able to detect acted and seizure-related falls in data not used for training. High impact falls occurring in any direction with respect to the camera are detected, as long as the vertical component of the fall is visible. In most records, acceptable false positive fractions are found for 90% sensitivity. We consider these results proof-of-concept of the method, and a promising outcome for eventual use of the method in clinical practice.

Fall detection sensitivity values >90% have been reported with >99% specificity in the Le2i database (Charfi et al., 2013; Fan et al., 2017; Senouci et al., 2016), but when algorithms were applied to new datasets with more challenging scenes, more modest performance results were obtained (up to 69% specificity at 60–75% sensitivity) (Debard et al., 2016, 2012; Fan et al., 2017). Whereas the performance of our algorithm in the Le2i database (92% specificity at 90% sensitivity) seems not fully competitive, we used a realistic performance testing scheme which possibly provides less favorable outcomes. We tested performance using shorter detection windows (leaving relatively more non-fall windows) and a random selection of data from all room settings instead of only a subset. We were able to show robustness of our algorithm in a separate test set and in new, challenging data with real-life falls. Our algorithm might also detect non-seizure falls, but was not tested on real-life non-seizure data. As the implementations of published algorithms are not publicly available, we could not test them on our SEIN test set.

Specificity of our algorithm needs to be improved further and validated in ongoing recordings made in real-life environments, where correct detection can be especially challenging (Debard et al., 2016). Training the algorithm on real-life movement patterns may improve the algorithm's ability to handle real-life events. Subject segmentation before calculation of the video features could possibly improve detection performance, but could also make the algorithm more sensitive to occlusions. Using a more fall-specific sound feature (e.g. one of the features described in (Li et al., 2012; Popescu et al., 2008; Salman Khan et al., 2015)), as opposed to the relatively course amplitude feature in our current algorithm, might also improve performance.

Postprocessing of detection output, for example by waiting after an alarm before generating new alarms, can reduce the false alarm rate when there is some ongoing disturbance. A smart system able to inactivate the detection module when a companion is present could prevent false detections due to multiple individuals in the scene, whereby it is assumed that the companion detects the fall. Similarly, automated inactivation at times of camera movements could prevent false positives. Note that using a 'subject inactivity after a fall' rule, sometimes applied in algorithms to decrease false positives, could decrease sensitivity for seizure-related falls, which may be followed by excess physical (seizure) activity.

Our fall detection algorithm is designed to detect high impact falls with high risk of injury and is less suitable for low impact falls without a clear moment of impact with the floor. An elderly person more sliding than falling off a chair for example, would likely not result in detection by our algorithm. Falls with soundless impact could in theory also be missed, which raises the question if soft flooring could impede detection. Many falls in the Le2i database involved subjects falling on mattresses, which were detected without problems. Visual inspection of training set sound signals confirmed the presence of peaks at times of mattress-falls. This suggests that soft flooring will probably not impede detection.

Our video features are amplitude-based and are dependent on camera placement (vertical component needed to detect a fall), camera zoom and subject distance to the camera. The proposed features derived from the physical fall model are, however, generic, and can be applied to other sensors able to quantify vertical velocity. Feature values may be corrected for (focal) distance using moving object size, but this may also introduce errors; for example in case of falling objects. In the SEIN test set, object size correction increased sensitivity (one extra TP) and specificity (two FPs less). Usage of depth sensors could provide more precise distance information and improve performance (Ma et al., 2014; Stone and Skubic, 2015). Although commercially available and affordable, depth cameras have a limited distance range (<5 m (Horaud et al., 2016)), making them less suitable for application in larger living areas. Depth sensors with a longer range might become more affordable in time.

Conflict of interest statement

The authors declare that they have no financial or personal relationships with other people or organizations that could inappropriately influence (bias) this work.

Acknowledgments

This work was supported by the Margaret Knip Fund, the Christelijke Vereniging voor de Verpleging van Lijders aan Epilepsie, and ZonMW (grant nr. 40-41200-98-9335). The authors would like to thank P. Augustijn, J. Kes and R. Klahn for their help in the collection of the SEIN fall database. We are grateful to J.W. Sander and G. S. Bell for critically reviewing the manuscript.

References

Alhimale, L., Zedan, H., Al-Bayatti, A., 2014. The implementation of an intelligent and video-based fall detection system using a neural network. Appl. Soft Comput. 18, 59–69. https://doi.org/10.1016/j.asoc.2014.01.024.

Belshaw, M., Taati, B., Snoek, J., Mihailidis, A., 2011. Towards a single sensor passive solution for automated fall detection. In: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society. EMBS, pp. 1773–1776. 10.1109/IEMBS.2011.6090506.

Charfi, I., Miteran, J., Dubois, J., Atri, M., Tourki, R., 2013. Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and Adaboost-based classification. J. Electron. Imaging 22,. https://doi.org/10.1117/1.JEI.22.4.041106 041106.

De Miguel, K., Brunete, A., Hernando, M., Gambao, E., 2017. Home camera-based fall detection system for the elderly. Sensors (Switzerland) 17, 2864. https://doi.org/10.3390/s17122864.

Debard, G., Baldewijns, G., Goedem, T., Tuytelaars, T., Vanrumste, B., 2015. Camera-based fall detection using a particle filter. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 6947–6950. 10.1109/EMBC.2015.7319990.

Debard, G., Karsmakers, P., Deschodt, M., Vlaeyen, E., Dejaeger, E., Milisen, K., Goedemé, T., Vanrumste, B., Tuytelaars, T., 2012. Camera-based fall detection on real world data. In: Dellaert, F., Frahm, J.-M., Pollefeys, M., Leal-Taixé, L., Rosenhahn, B. (Eds.), Outdoor and Large-Scale Real-World Scene Analysis. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 356–375. https://doi.org/10.1007/978-3-642-34091-8_16.

Debard, G., Mertens, M., Deschodt, M., Vlaeyen, E., Devriendt, E., Dejaeger, E., Milisen, K., Tournoy, J., Croonenborghs, T., Goedemé, T., Tuytelaars, T., Vanrumste, B., 2016. Camera-based fall detection using real-world versus simulated data: How far are we from the solution? J. Ambient Intell. Smart Environ. 8, 149–168. https://doi.org/10.3233/AIS-160369.

Fan, Y., Levine, M.D., Wen, G., Qiu, S., 2017. A deep neural network for real-time detection of falling humans in naturally occurring scenes. Neurocomputing 260, 43–58. https://doi.org/10.1016/j.neucom.2017.02.082.

Feng, W., Liu, R., Zhu, M., 2014. Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera. Signal, Image Video Process 8, 1129–1138. https://doi.org/10.1007/s11760-014-0645-4.

Foroughi, H., Aski, B.S., Pourreza, H., 2008. Intelligent video surveillance for monitoring fall detection of elderly in home environments. In: Proceedings of 11th International Conference on Computer and Information Technology, ICCIT 2008, pp. 219–224, https://doi.org/10.1109/ICCITECHN.2008.4803020.

Goudelis, G., Tsatiris, G., Karpouzis, K., Kollias, S., 2015. Fall detection using history triple features. In: Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments, https://doi.org/10.1145/2769493.2769562.

Hazelhoff, L., Han, J., de With, P.H.N., 2008. Video-based fall detection in the home using principal component analysis. In: Blanc-Talon, J., Bourennane, S., Philips, W., Popescu, D., Scheunders, P. (Eds.), Advanced Concepts for Intelligent Vision Systems. ACIVS 2008. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 298–309. https://doi.org/10.1007/978-3-540-88458-3.

Horaud, R., Hansard, M., Evangelidis, G., Clément, M., 2016. An overview of depth cameras and range scanners based on time-of-flight technologies. Mach. Vis. Appl. 27, 1005–1020. https://doi.org/10.1007/s00138-016-0784-4.

Horn, B.K.P., Schunck, B.G., 1981. Determining optical flow. Artif. Intell. 17, 185–203. https://doi.org/10.1016/0004-3702(93)90173-9.

Huang, C., Chen, E., Chung, P., 2007. Fall detection using modular neural networks with back-projected optical flow. Biomed. Eng. Appl. Basis Commun. 19, 415–424. https://doi.org/10.1142/S1016237207000525.

Kalitzin, S., Petkov, G., Velis, D., Vledder, B., Lopes da Silva, F., 2012. Automatic segmentation of episodes containing epileptic clonic seizures in video sequences. IEEE Trans. Biomed. Eng. 59, 3379–3385. https://doi.org/10.1109/TBME.2012.2215609.

Kangas, M., Vikman, I., Nyberg, L., Korpelainen, R., Lindblom, J., Jämsä, T., 2012. Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects. Gait Posture 35, 500–505. https://doi.org/10.1016/j.gaitpost.2011.11.016.

Khawandi, S., Daya, B., Chauvet, P., 2011. Implementation of a monitoring system for fall detection in elderly healthcare. Proc. Comput. Sci. 3, 216–220. https://doi.org/10.1016/j.procs.2010.12.036.

Krumholz, A., Hopp, J., 2008. Falls give another reason for taking seizures to heart. Neurology 70, 1874–1875. https://doi.org/10.1212/01.wnl.0000312285.73631.ff.

Li, Y., Ho, K.C., Popescu, M., 2012. A microphone array system for automatic fall detection. IEEE Trans. Biomed. Eng. 59, 1291–1301. https://doi.org/10.1109/TBME.2012.2186449.

Liao, Y.T., Huang, C.-L., Hsu, S.-C., 2012. Slip and fall event detection using Bayesian Belief Network. Pattern Recognit. 45, 24–32. https://doi.org/10.1016/j.patcog.2011.04.017.

Liu, C.-L., Lee, C.-H., Lin, P.-M., 2010. A fall detection system using k-nearest neighbor classifier. Expert Syst. Appl. 37, 7174–7181. https://doi.org/10.1016/j.eswa.2010.04.014.

Liu, T., Yao, H., Ji, R., Liu, Y., Liu, X., Sun, X., Xu, P., Zhang, Z., 2008. Vision-based semi-supervised homecare with spatial constraint. In: Advances in Multimedia Information Processing – PCM 2008, Vol. 5353 of Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 416–425. https://doi.org/10.1007/978-3-540-89796-5_43.

Ma, X., Wang, H., Xue, B., Zhou, M., Ji, B., Li, Y., 2014. Depth-based human fall detection via shape features and improved extreme learning machine. IEEE J. Biomed. Heal. Inform. 18, 1915–1922. https://doi.org/10.1109/JBHI.2014.2304357.

Makantasis, K., Protopapadakis, E., Doulamis, A., Doulamis, N., Matsatsinis, N., 2016. 3D measures exploitation for a monocular semi-supervised fall detection system. Multimed. Tools Appl. 75, 15017–15049. https://doi.org/10.1007/s11042-015-2513-9.

Miaou, S.G., Sung, P.H., Huang, C.Y., 2006. A customized human fall detection system using omni-camera images and personal information. In: Conference Proceedings – 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, D2H2 2006, pp. 39–42, https://doi.org/10.1109/DDHH.2006.1624792.

Mirmahboub, B., Samavi, S., Karimi, N., Shirani, S., 2013. Automatic monocular system for human fall detection based on variations in silhouette area. IEEE Trans. Biomed. Eng. 60, 427–436. https://doi.org/10.1109/TBME.2012.2228262.

Nait-Charif, H., McKenna, S., 2004. Activity summarization and fall detection in a supportive home environment. IEEE International Conference on Pattern Recognition, 20–23. https://doi.org/10.1109/ICPR.2004.1333768.

Nater, F., Grabner, H., Jaeggli, T., Van Gool, L., 2009. Tracker trees for unusual event detection. 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, 1113–1120. https://doi.org/10.1109/ICCVW.2009.5457578.

Olivieri, D.N., Gómez Conde, I., Vila Sobrino, X.A., 2012. Eigenspace-based fall detection and activity recognition from motion templates and machine learning. Expert Syst. Appl. 39, 5935–5945. https://doi.org/10.1016/j.eswa.2011.11.109.

Popescu, M., Li, Y., Skubic, M., Rantz, M., 2008. An acoustic fall detector system that uses sound height information to reduce the false alarm rate. In: 30th Annual International IEEE EMBS Conference, pp. 4628–4631.

Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J., 2011. Robust video surveillance for fall detection based on human shape deformation. IEEE Trans. Circuits Syst. Video Technol. 21, 611–622. https://doi.org/10.1109/TCSVT.2011.2129370.

Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J., 2006. Monocular 3D head tracking to detect falls of elderly people. Annual International Conference of the IEEE Engineering in Medicine and Biology – Proceedings, 6384–6387. https://doi.org/10.1109/IEMBS.2006.260829.

Rubenstein, L.Z., 2006. Falls in older people: epidemiology, risk factors and strategies for prevention. Age Ageing 35, ii37-ii41. https://doi.org/10.1093/ageing/afl084.

Russell-Jones, D.L., Shorvon, S.D., 1989. The frequency and consequences of head injury in epileptic seizures. J. Neurol. Neurosurg. Psychiatry 52, 659–662. https://doi.org/10.1136/jnnp.52.5.659.

Salman Khan, M., Yu, M., Feng, P., Wang, L., Chambers, J., 2015. An unsupervised acoustic fall detection system using source separation for sound interference suppression. Signal Process. 110, 199–210. https://doi.org/10.1016/j.sigpro.2014.08.021.

Senouci, B., Charfi, I., Heyrman, B., Dubois, J., Miteran, J., 2016. Fast prototyping of a SoC-based smart-camera: a real-time fall detection case study. J. Real-Time Image Process. 12, 649–662. https://doi.org/10.1007/s11554-014-0456-4.

Stone, E.E., Skubic, M., 2015. Fall detection in homes of older adults using the microsoft kinect. IEEE J. Biomed. Heal. Inform. 19, 290–301. https://doi.org/10.1109/JBHI.2014.2312180.

Tao, J., Turjo, M., Wong, M.-F., Wang, M., Tan, Y.-P., 2005. Fall incidents detection for intelligent video surveillance. In: 5th International Conference on Information Communications & Signal Processing, pp. 1590–1594. https://doi.org/10.1109/ICICS.2005.1689327.

Toreyin, B.U., Dedeoglu, Y., Cetin, A.E., 2005. HMM based falling person detection using both audio and video. In: Sebe, N., Lew, M., Huang, T. (Eds.), Computer Vision in Human-Computer Interaction. HCI 2005. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 211–220. https://doi.org/10.1109/SIU.2006.1659753.

Vishwakarma, V., Mandal, C., Sural, S., 2007. Automatic detection of human fall in video. International Conference on Pattern Recognition and Machine Intelligence, 616–623. https://doi.org/10.1007/978-3-540-77046-6.

Wang, S., Chen, L., Zhou, Z., Sun, X., Dong, J., 2016. Human fall detection in surveillance video based on PCANet. Multimed. Tools Appl. 75, 11603–11613. https://doi.org/10.1007/s11042-015-2698-y.

Wu, G., 2000. Distinguishing fall activities from normal activities by velocity characteristics. J. Biomech. 33, 1497–1500. https://doi.org/10.1016/S0021-9290(00)00117-2.

Yu, M., Yu, Y., Rhuma, A., Naqvi, S.M.R., Wang, L., Chambers, J.A., 2013. An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment. IEEE J. Biomed. Heal. Inform. 17, 1002–1014. https://doi.org/10.1109/JBHI.2013.2274479.

Yun, Y., Gu, I.Y.H., 2016. Human fall detection in videos by fusing statistical features of shape and motion dynamics on Riemannian manifolds. Neurocomputing 207, 726–734. https://doi.org/10.1016/j.neucom.2016.05.058.

Zerrouki, N., Harrou, F., Sun, Y., Houacine, A., 2016. A data-driven monitoring technique for enhanced fall events detection. IFAC-PapersOnLine 49, 333–338. https://doi.org/10.1016/j.ifacol.2016.07.135.

Zhang, Z., Tong, L.G., Wang, L., 2010. Experiments with computer vision methods for fall detection. In: Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '10), https://doi.org/10.1145/1839294.1839324.

Zweng, A., Zambanini, S., Kampel, M., 2010. Introducing a statistical behavior model into camera-based fall detection. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Chung, R., Hammoud, R., Hussain, M., Kar-Han, T., Crawfis, R., Thalmann, D., Kao, D., Avila, L. (Eds.), Advances in Visual Computing. ISVC 2010. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 163–172. https://doi.org/10.1007/978-3-642-17289-2_16.