Video-Based Fall Detection in the Home Using Principal Component Analysis

Lykele Hazelhoff[1], Jungong Han[1], and Peter H.N. de With[1,2]

[1] University of Technology Eindhoven, the Netherlands
[2] CycloMedia Technology B.V., Waardenburg, the Netherlands

Abstract. This paper presents the design and real-time implementation of a fall-detection system, aiming at detecting fall incidents in unobserved home situations. The setup employs two fixed, uncalibrated, perpendicular cameras. The foreground region is extracted from both cameras and for each object, principal component analysis is employed to determine the direction of the main axis of the body and the ratio of the variances in x and y direction. A Gaussian multi-frame classifier helps to recognize fall events using the above two features. The robustness of the system is increased by a head-tracking module, that can reject false positives. We evaluate both performance and efficiency of the system for a variety of scenes: unoccluded situations, cases where the person carries objects and occluded situations. Experiments show that our algorithm can operate at *real-time* speed with more than 85% fall-detection rate.

1 Introduction

Falls are a major health risk for senior citizens, since almost one-third of them fall once a year, causing physical injuries and further health complications, both physically and mentally. Therefore, surveillance of such incidents is important and can also help in setting up a more efficient approach for human support from nurses. For fall detection, a number of solutions have been already developed. The most common way is the usage of a wearable button, which can be pressed in case of a fall or another emergency. However, these systems do not help if the person is hit by a heart attack or a stroke. Also, automatic fall detectors are available, consisting of wearable velocity sensors. While the performance of such sensors is reasonable, elderly people should always wear them to be protected. Computer-vision-based fall-detection overcomes these problems, since it provides fully passive monitoring and can also be used for other tasks, such as security and fire-alarm. Therefore, the use of a video-based fall-detection system can be advantageous over active sensors.

1.1 Related Work

In the last years, some research has been carried out on vision-based fall-detection. The requirements for a vision-based fall detection system have been evaluated in [1], where the detection accuracy, response after a detected fall and the privacy

J. Blanc-Talon et al. (Eds.): ACIVS 2008, LNCS 5259, pp. 298–309, 2008.

of the monitored elderly person turned out to be important factors. The first fall-detection technique is to represent the foreground with a bounding box, where the bounding-box aspect ratio is the major visual feature. Based on this technique, the work in [3] is focused on a network of low-cost cameras with embedded processors. When one camera is calibrated, fall incidents can be localized. In this way, an inexpensive and power-efficient fall-detection solution is achieved. In order to further improve the performance of the technique from [3], the proposal in [4] combines the change in aspect ratio with audio information to detect falls using Hidden Markov Models (HMMs). While the performance in the setup is good, a silent environment is required.

Instead of using a bounding box, [5] proposes to fit an ellipse on the foreground area, using a ceiling-placed, wide-angle camera. For fall detection, the motion-history image and the the standard deviations of the angle and aspect ratio of the ellipse are used. After a fall, the authors assume that large motion is absent. The disadvantage is that fast sitting down can be seen also as a fall, and slow falls can even be missed. In [6], authors also use ellipse-fitting on the extracted foreground area. Different from [5], it proposes a technique to identify entry zones and inactivity zones automatically, using a MAP estimation of Gaussian mixture models. Juang *et al.* [8] employ both projection histograms and the bounding-box aspect-ratio associated with a neural fuzzy network. A DFT is applied on both the vertical and horizontal projection histogram, and a self-constructing neural network is used for posture classification. Based upon the classification results, a fall-detection system is proposed. In [2], projection histograms are also used, but the applied classifier is implemented based on an HMM. Moreover, a warping method is proposed to reduce the effects of occlusion when switching between rooms with different calibrated cameras. As a drawback, the use of projection histograms is influenced by shadows, carried objects and occlusions. A complete different approach is given in [7], where the head is tracked in 3D coordinates, using a particle filter and a calibrated camera. A fall is detected when large head motion appears in both horizontal and vertical direction. This approach is not robust in the sense that the speed of the head is also large when people sit down, and during slow falls the speed of the head is low, resulting in an undetected fall. Most of the fall-detection techniques have the same drawback: they only deal with the ideal situation but ignore the variety of the practical cases. For example, there is no system to treat more realistic situations where a person is carrying and using objects, or the situation where a person falls out of a chair or uses walking tools.

1.2 Problem Statement and Contribution

Since the use of objects and walking tools regularly occurs in practical situations, we focus on detecting falls in such real-life situations, and we accept that occlusions by furniture, shadows, bad segmentation results and the use of objects will certainly occur. We contribute by providing a novel and robust technique for

fall detection on single persons, that can handle the above-mentioned occlusions and use of objects. The technical consequences of the previous discussion details the major contributions of this paper into two aspects. First, we investigate the changes of the main axis of a human body to deduce a possible fall event. Comparing with other visual features like bounding-box aspect-ratio, this feature is robust for the situation that a person is partially occluded or carrying objects. Also, we add a head-tracking module to further reject false positives. Second, we propose an low-cost implementation of the algorithm using low-resolution cameras and existing widely available PC components.

The remaining of this paper is organized as follows. In Section 2, the global system overview will be given. Section 3 explains the algorithm design, followed by some implementation details in Section 4. Section 5 states the obtained results, followed by the conclusions.

2 System Overview

Fig. 1 gives an overview of the complete system, where the frames from two uncalibrated, overlapping, perpendicular cameras are taken as input and the fall-detection indicator is the output. The complete system consists of five modules, which are briefly explained below.

1. *Object segmentation:* at each sample moment, an image is taken from both synchronized cameras. Foreground regions are obtained by background subtraction and objects are formed by connecting information components.
2. *Object tracking:* although we focus on single persons, there can be multiple objects in the scene. A tracker is used to track the objects in each camera view. Objects can be marked as non-human, which are identified based on size and absence of both motion and a head region. The tracker also fuses the human objects from both camera views together.
3. *PCA-based feature extraction:* from each object which is not identified as non-human, the direction of the principal component and the variance ratio are extracted for both camera views.
4. *Fall detection:* the previous features are used to determine whether the person has fallen using a multi-frame Gaussian classifier.
5. *Head tracking:* The head position is tracked based on skin-color information. After a detected fall, the head position is used to reject false detections.

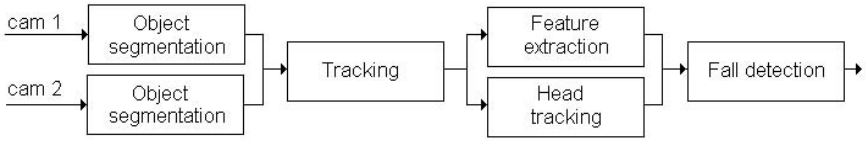

Fig. 1. Schematic overview of our fall-detection system

3 Algorithm Design

3.1 Foreground Extraction, Object Detection and Object Tracking

From both perpendicular cameras, foreground regions are obtained by background subtraction. After this step, small noise regions are removed by filtering and the foreground pixels are grouped into objects. Since small occlusions can occur, two foreground pixels are considered to be part of the same object if their Euclidian distance is less than e.g. a limited amount of pixels. Objects that are partially outside the camera view are discarded to avoid false alarms.

Both cameras are equipped with a tracker, which tracks the detected objects on position and size and can mark objects as human or non-human. Non-human objects are defined as motionless and without a head region, other objects are marked as human. Since non-human objects can not move itself, they are most likely put in the scene by a human object. This results in the detection of an object split, where the smaller, motionless object is marked as non-human, unless a head region is found on this object. The human-marked objects present in both camera views are fused together. At the moment, we focus on observation of single persons and normally both cameras observe the same human object from different viewpoints. Therefore, all combinations of human-marked objects are evaluated. The described tracker can be extended with additional tracking features, such as color information; but it is our opinion that with one person in the scene the proposed tracking algorithm is sufficient. When more than two cameras are used, the human-marked objects present in all camera views can be fused together, however, we focus on two cameras.

3.2 Feature Extraction

For objects marked as human, the orientation of the main axis and the ratio of the variances in horizontal and vertical direction are measured for both cameras. To this end, the covariance matrix Σ is computed, according to the formula in Eqn. (1), where μ denotes the average value and X and Y are the distributions of the object pixels in x and y directions, respectively. This matrix equals:

$$\Sigma = \begin{bmatrix} E\left((X-\mu_x)\left(X-\mu_x\right)\right) & E\left((X-\mu_x)\left(Y-\mu_y\right)\right) \\ E\left((Y-\mu_y)\left(X-\mu_x\right)\right) & E\left((Y-\mu_y)\left(Y-\mu_y\right)\right) \end{bmatrix}. \tag{1}$$

The ratio of the variances in horizontal and vertical direction ρ is formulated as:

$$\rho = \frac{E\left((X-\mu_x)\left(X-\mu_x\right)\right)}{E\left((Y-\mu_y)\left(Y-\mu_y\right)\right)}. \tag{2}$$

The orientation of the main axis can be obtained by a singular value decomposition of the covariance matrix Σ, decomposing Σ into the matrix product $\Sigma = USV'$. The angle between the vertical camera-axis (assumed to be perpendicular to the floor plane) and the main axis ϕ is calculated as:

$$\phi = atan\left(\frac{V(1,1)}{V(1,2)}\right). \tag{3}$$

This process is similar to a *Principal Component Analysis* (PCA): the main directional axis is determined by finding the component with the largest variance. For an unoccluded human, this leads to the main orientation of the human body, where $\phi \approx 0$ for a standing person. One camera can measure the main axis only in the direction perpendicular to the principal camera-axis. Therefore, at least two, almost perpendicular, cameras are needed to measure the main-axis orientation in the two orthogonal directions. The variance ratio can be seen as a pixel-based measurement for the aspect ratio of the object. This feature is added for additional robustness and comes for free, since the covariance matrix is needed for calculation of the main-axis angle ϕ in any case.

The obtained four-dimensional feature-space (containing ϕ_1, ρ_1 from camera 1 and ϕ_2, ρ_2 from camera 2) can be reduced to two dimensions by coupling the main-axis angles and variance ratios from both cameras together. In case of a fall, at least one main-axis angle should be large in absolute value and both variance ratios should be small, indicating that both angles can be coupled together to $\bar{\phi}$ by taking the absolute maximum. Similarly, both variance ratios can be combined to $\bar{\rho}$ by summation. Thus, from all human-marked objects the following two features are extracted:

$$\bar{\phi} = \max\left(|\phi_1|, |\phi_2|\right) \qquad \bar{\rho} = \rho_1 + \rho_2. \tag{4}$$

3.3 Head Detection

To identify objects as humans and to increase the robustness of the fall-detection system, the position of the head is estimated. For this, a CrCb skin-color model is used, which was also used in our previous work [9]. Different skin-colored parts from various photographs are extracted and a Gaussian skin-color model is developed. A pixel with values Cr, Cb is classified as skin-colored if Eqn. (5) holds, where the mean values μ_{Cr}, μ_{Cb} and the covariance matrix C are parameter of the Gaussian model. The detection threshold α is empirically chosen as 10.

$$\left[Cr - \mu_{Cr} \;\; Cb - \mu_{Cb}\right] C^{-1} \left[Cr - \mu_{Cr} \;\; Cb - \mu_{Cb}\right]^T < \alpha. \tag{5}$$

During the head-search phase, the skin-colored pixels are connected to blobs. The largest blob farthest away from the object center along the main axis of the object is taken as head candidate. This candidate is further extended away from the center coordinate to cover the whole head, as depicted in Fig. 2 A-D. The found head candidate is matched against the following constraints:

1. The head candidate should contain at least 15 skin-colored pixels, and at most 5% of the total object pixels.
2. The eccentricity of the head candidate should lie between 0.5 and 2.

Once the head is found, the head can be tracked in the successive frames. After compensation for global object-motion, the skin-colored blob closest to the old position is taken as head, as shown in Fig. 2 E-G. Because the frame rate is low, selecting the hand as head during falls and other large movements is avoided by

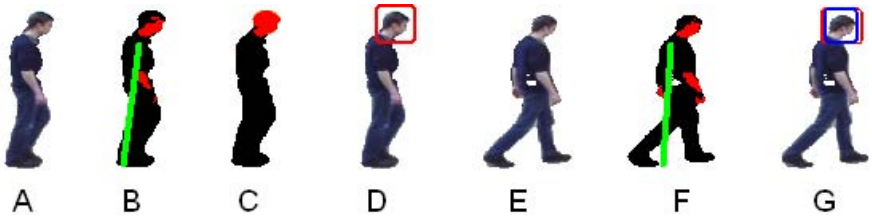

Fig. 2. A and E: input images. B and F: found skin-color blobs (red) and main axis (green). C: most-probably skin-color blob extended to a head candidate. D and G: found heads enclosed in a red frame (the blue frame in G encloses the predicted position).

keeping the search range small. If no blob is close enough to the old position, the head search procedure is started again. Locating the head based on skin-color information works only for a side or a frontal view on the person; since we use two perpendicular cameras, probably one of them is able to track the head.

3.4 Fall Detection

The features described in Section 3.2 are used for detection of fall incidents. Falling is a dynamic process: the time, speed and direction of a fall are strongly varying. Therefore, we aim at the detection of occurred falls by detection of lying persons. Persons can also intentionally lie down, for example on a couch or bed. Therefore, inactivity regions can be defined where lying is permitted. Although inactivity zones are learned automatically in [6], we have marked inactivity zones manually for simplicity.

The behavior of the maximum main-axis angle $\bar{\phi}$ and the variance-ratio sum $\bar{\rho}$ during different activities is investigated by recording a test sequence, containing different activities such as walking, sitting (both on the floor and on a chair), bending forward and falling down. Occlusions are avoided to obtain a clear view. The features measured from this test sequence during the different behavior are plotted in Fig. 3. Since we focus on detection of persons that have fallen down, the transitions between the different activities are not shown. As can be seen, there is a clear difference in feature values between the different activities and between fallen and the other activities in particular. In case of a fall $\bar{\phi}$ is around 85° and $\bar{\rho}$ is small compared to the other activities. When both cameras are not completely perpendicular, $\bar{\phi}$ can be lower during a fall. A miss-alignment of approximately 20° is allowed to maintain the gap between fallen and the other activities, however, at the penalty of sacrificing some robustness.

A classifier is set up to discriminate between two classes: falls and non-falls. Although the data is not normally distributed, we propose to represent the fall class by a Gaussian distribution, as shown in Eqn. (6), where $\mu_{\bar{\phi}}$, $\mu_{\bar{\rho}}$ are the mean values and $\Sigma_{\bar{\phi}-\bar{\rho}}$ is the covariance matrix of both features, thus

$$p_{fall}(\bar{\phi},\bar{\rho}) = \frac{1}{\sqrt{2\pi}|\Sigma_{\bar{\phi}-\bar{\rho}}|} e^{-\frac{1}{2}\left(\begin{bmatrix}\bar{\phi}-\mu_{\bar{\phi}}\\ \bar{\rho}-\mu_{\bar{\rho}}\end{bmatrix}^T \Sigma^{-1}_{\bar{\phi}-\bar{\rho}} \begin{bmatrix}\bar{\phi}-\mu_{\bar{\phi}}\\ \bar{\rho}-\mu_{\bar{\rho}}\end{bmatrix}\right)}. \quad (6)$$

Each feature set is converted to a probability of a fall. This likelihood information can be used to increase the robustness by using a sliding-window-based classification instead of frame-based classification. As window-based classification with window size N, the probabilities are low-pass filtered using a moving-average filter of length N, and the result is thresholded; the threshold equals the lowest probability on a fall measured for the fallen-activity in the test sequence multiplied by N. A larger window size N leads to a smoother classification result. We used $N = 25$ to remove incidental outliers, for example due to bad segmentation results. This results in the detection of a fall within 5 seconds, when a frame rate of 5 frames per second is used. The system can be adapted to detect different activities, for example sitting on the floor, by adjusting the classification parameters.

An additional clue for occurrence of a fall is given by tracking of the head position. During some behavior the main axis flips 90 degrees, while the head position remains about the same (see for example Fig. 4). By detecting these cases, the amount of false alarms can be lowered; when the head position does not move significantly in one of the camera views, the detected fall is rejected. When a fall occurs, the head position should change and head-motion is present (investigated in detail in [7]); detection of significant head-position change or lose of head tracking confirm the detected fall-event.

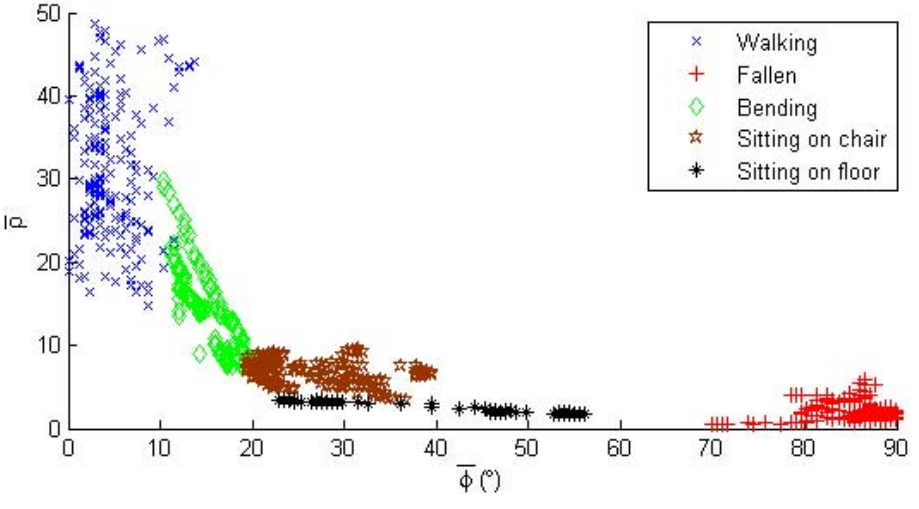

Fig. 3. Plot of the measured features during the different activities in the test sequence

4 Real-Time Implementation

A fall detector should work in real-time to be useful. The proposed algorithm involves large amount of computations, which are also dependent on the amount

of foreground pixels. Thus real-time implementation with a more or less constant frame rate is a challenging task, especially because the computational load grows quadratic with the size of the object and is thus dependent on the distance to the camera. Therefore, an adaptive-resolution approach is applied to flatten the computation time per frame and to achieve an acceptable frame rate.

The software implementation is done following the scheme of Fig. 1. Background subtraction is carried out by modeling the background with a Gaussian Mixture Model. The obtained foreground mask is subsequently down sampled 20 times and connected components is applied on the down-sampled mask; the found objects are tracked. From objects marked as human, the head region is found by looking for skin-colored pixels in the corresponding foreground object (for this, the original foreground mask is used). The use of a 512×512 boolean lookup table in Cr, Cb color space reduces the impact on the frame rate. After that, the covariance matrix is computed on a down-sampled version of the object, where the down-sampling factor is dependent on the number of object pixels in the 20-times down-sampled foreground-mask. In order to maintain a good accuracy, each object should consist of at least Q_{min} pixels. From the obtained covariance matrix the features are extracted. This process is done for both camera views at the same time, and afterwards fusion of object features and classification is applied. An appropriate value for Q_{min} is determined by calculating the main-axis angle and variance ratio for different values of Q_{min} for all frames of the test sequence and analyzing the occurring deviations, which are shown in Table 1. A minimum object size of about 2,000 pixels is both accurate and achieves a flatter computation time.

Using this approach, the fall-detection system is implemented on a 2.8-GHz Intel P-IV PC using OpenCV. Sequences with a resolution of 640×480 pixels are used; foreground mask retrieval can be done with 11 frames per second, extracting features lowers this to 6.9 frames per second and the complete process runs with 5.7 frames per second on the average, where each frame is processed within 0.27 seconds. For 320×240 sequences, processing the same sequences is achieved with more than 15 frames per second on the average; but when objects are far away from the camera, the head region can be missed due to its small size. Therefore, we prefer using 640×480 sequences; since we focus on detection of fallen persons instead of falling persons, an average frame-rate of more than 5 frames per second is acceptable and we call our system a *real-time* system.

Table 1. Minimal object size Q_{min} influencing the accuracy of the estimated features. The shown computation time is measured by Matlab. Processing the original object results in an average computation time of 2.33 seconds.

Q_{min}	1,000	2,000	3,000	4,000	5,000	6,000
Avg. error in $\bar{\phi}$ (°)	0.67	0.45	0.46	0.24	0.18	0.03
Avg. error in $\bar{\rho}$	0.17	0.11	0.15	0.13	0.11	0.05
Avg. computation time (s)	0.47	0.50	0.64	0.97	1.11	1.96

5 Experiments and Results

The presented video fall-detection system has been tested on a number of videos captured at our office, since test videos of elderly people are scarce, especially when they contain falls. It is noted that these videos are captured under different lighting conditions and with different backgrounds. In our dataset, there are three classes of sequences. The first one contains normal activities, such as walking, bending, sitting and falling down and consists of 4 sequences with a total of 2,085 frames. The second type of sequences describes more realistic motion patterns, such as reading the newspaper, relaxing on a chair, cleaning, placing items in the scene and using a chair as a walking tool. This class contains 5 sequences with 1,456 frames in total. The last type of the sequences portrays scenes with bad segmentation results and occlusions by chairs and tables. This class consists of 4 sequences containing 2,535 frames. Obviously, the last two cases are more challenging and close to the real-life situation.

In the first class of test videos, containing mainly unoccluded behavior, the proposed fall-detection algorithm is able to detect all falling incidents, while no false alarms are given.

In the second class of test videos, more realistic situations are addressed. Falls are still detected accurately and false alarms are not given. Fig. 4 gives some snapshots from a typical sequence. The behavior of the main axis is not always equal to the main orientation of the body. For example, when bending and placing an object, the main axis flips 90° for one camera (see most-left images in Fig. 4). In a single camera setup, a fall is detected; but camera 2 observes a normal situation and also the head is trackable, both can reject the seen fall. Only in case the view of the second camera was occluding and the head tracking failed, or the head moved significantly, a fall can be detected erroneously.

The third class of test videos contains situations including bad segmentation results and occlusions, where the most challenging problems occur. Fig. 5 shows examples from two of the investigated occluded situations. When an outlier part (for example the feet) of the human is occluded or not segmented well, the variance ratio is lowered, while the main-axis orientation is influenced only a little bit. When the center part of the body is occluded, the object can be split in two smaller blobs, as shown in Fig. 6. In this case, both blobs can be treated as separate objects or can be seen as part of the same object. The latter gives reasonable results, but the merging distance cannot be chosen arbitrary large, since merging different objects together is disadvantageous. Also occlusions lower

Table 2. Fall-detection accuracy of our fall-detection system (left columns) and the bounding-box fall-detection method (right columns)

Activity type	Sequences	Detected falls	False positives
Normal activity	4	100% — 100%	0 — 5
Realistic activity	5	100% — 91%	0 — 4
Occluding activity	4	55% — 44%	1 — 6

Fig. 4. Typical images from a sequence where objects are used; Top view: input images, bottom view: segmented objects. Left: a bag is carried and put down ($\bar{\phi} = 80.8°, \bar{\rho} = 21.3$); right: a chair is used as a walking tool ($\bar{\phi} = 24.6°, \bar{\rho} = 8.5$). The main axis is drawn in green.

Fig. 5. Typical images from sequences where occlusions occur; Top view: input images, bottom view: segmented objects. Left: a person is reading and is partially occluded for both cameras ($\bar{\phi} = 21.19°, \bar{\rho} = 5$); right: a person has fallen and is occluded partially for both cameras ($\bar{\phi}_{max} = 87.7°, \bar{\rho} = 0.3$, the fall is detected). The main axis is drawn in green.

the amount of involved foreground blocks, leading to a smaller object size. In this way, human objects are sometimes seen as small, not important objects. Using temporal information can correct this. Other errors occur when a person becomes occluded after a fall. Now, no person is present in the foreground area, and no fall is detected. This problem can be solved by defining exit zones in which the person is allowed to exit, for example by setting exit zones at the right and left side of the camera view.

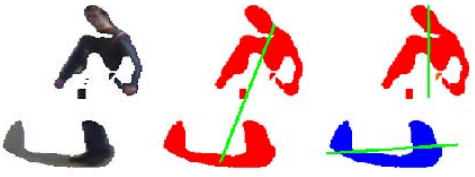

Fig. 6. Example of a bad segmentation result; the middle of the person is not detected. Left: foreground mask; middle: result when both blobs are seen as one object; right: result when both blobs are seen as separate objects. The main axis is drawn in green.

Table 2 summarizes the number of sequences for each class and shows the detection results. In the left columns the result of our approach is given, at the right side the result of the bounding-box fall-detection technique using the minimum bounding-box ratio of both cameras and a threshold of 0.8, as proposed in [3]. Our approach can detect all falls in both normal situations and in situations where carried objects are used, leading to a good performance. When occlusions occur, the performance decreases. Compared to the bounding-box method, the proposed fall-detection algorithm achieves better detection results in cases where objects are carried by the persons and gives much less false alarms.

6 Conclusions and Future Work

In this paper, a new system for detecting fall events of single persons has been introduced, with an emphasis on robustness with respect to carrying additional objects and a high computation efficiency. We have contributed in two ways. First, we have investigated whether the main axis of the object can be used as a reliable fall-detection indicator. Second, we have proposed an extensible and low-cost implementation, based on off-the-shelf PC-components. The implementation is scalable and allows that more cameras can easily be added to cover the whole room. For this, only the tracker should be changed.

A specific feature of the proposed algorithm is that the head of the person is detected and tracked. The position of the head is taken into account in order to obtain a high robustness and avoid false detections. Even when occlusions occur for both cameras, no false alarms are given, as long as the head is trackable and not fast moving. Because velocity during falling is only taken into account by the presence of change in the head position, slow falls are detected correctly, in contrast with already existing solutions in literature.

The proposed fall-detection algorithm achieves a 100% detection rate when large occlusions are absent, regardless whether objects are carried or a chair is used as a walking tool. In case large occlusions are present, still falls can be detected, as long as no more than half the object is occluded. Otherwise, the fall can be missed and false alarms can be given. Adding more cameras can be a solution for this problem.

In a practical implementation, the person should be monitored by at least two perpendicular cameras at every place in the room, implying that four cameras are needed to cover a square room completely. However, a part of the room may not be used frequently, therefore, a one-camera fall-detection system can be integrated to detect falls in the non-overlapping parts of the camera views. While the detection accuracy in these parts is slightly lower than in the main part, less cameras can be used to detect falls in the whole room. Other possible improvements can be obtained with a more advanced tracker, which also enables to analyze situations with multiple persons. Our future work might also be the usage of the same features to predict whether persons will fall out of a bed, which is a very practical case in a hospital environment.

References

1. Marquis-Faulkes, F., McKenna, S., Newell, A., Gregor, P.: Gathering the requirements for a fall monitor using drama and video with older people. Technology and Disability 17(4), 227–236 (2005)
2. Cucchiara, R., Prati, A., Vezzani, R.: A multi-camera vision system for fall detection and alarm generation. Expert Systems 24, 334–345 (2007)
3. Williams, A., Ganesan, D., Hanson, A.: Aging in place: fall detection and localization in a distributed smart camera network. Proceedings of ACM Multimedia (September 2007)
4. Töreyin, B., Dedeoglu, Y., Çetin, A.: HMM based falling person detection using both audio and video. In: Sebe, N., Lew, M., Huang, T.S. (eds.) HCI/ICCV 2005. LNCS, vol. 3766, pp. 211–220. Springer, Heidelberg (2005)
5. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Fall detection from human shape and motion history using video surveillance. In: AINA Workshops (2) (2007)
6. Nait-Charif, H., McKenna, S.: Activity Summarisation and fall detection in a supportive home environment. In: International Conference on Pattern Recognition (ICPR 2004) (2004)
7. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Monocular 3D head tracking to detect falls of elderly people. In: International Conference of the IEEE Engineering in Medicine and Biology Society (2006)
8. Juang, C., Chang, C.: Human body posture classification by a neural fuzzy network and home care system application. SMC-A 6(37), 984–994 (2007)
9. Zuo, F., de With, P.H.N.: Fast Human Face Detection Using Successive Face Detectors with Incremental Detection Capability. In: Proc. SPIE, vol. 5022 (2003)