

Camera-based fall detection using a particle filter

Glen Debard^{*1,2,3}, Greet Baldewijns², Toon Goedemé^{3,4}, Tinne Tuytelaars^{3,5} and Bart Vanrumste^{2,6,7}

Abstract—More than thirty percent of persons over 65 years fall at least once a year and are often not able to get up again. The lack of timely aid after such a fall incident can lead to severe complications. This timely aid can however be assured by a camera-based fall detection system triggering an alarm when a fall occurs. Most algorithms described in literature use the biggest object detected using background subtraction to extract the fall features. In this paper we compare the performance of our state-of-the-art fall detection algorithm when using only background subtraction, when using a particle filter to track the person and a hybrid method in which the particle filter is only used to enhance the background subtraction and not for the feature extraction. We tested this using our simulation data set containing reenactments of real-life falls. This comparison shows that this hybrid method significantly increases the sensitivity and robustness of the fall detection algorithm resulting in a sensitivity of 76.1% and a PPV of 41.2%.

I. INTRODUCTION

As thirty to forty-five percent of people older than 65 and living at home fall at least once a year [1], fall incidents are considered a major cause of health related problems for older persons. Following a fall incident the lack of timely aid can lead to both physical complications such as dehydration, pressure ulcers, and even death as well as to psychological consequences such as fear of falling, loss of self-confidence, and loss of independence [1], [2]. Accurate fall detection systems that can help insure timely aid and can therefore reduce the consequences of a fall incident, are therefore needed.

An overview of the existing research concerning these accurate fall detection systems given in [3] indicates that both contactless and wearable sensors are used. However as a market study from SeniorWatch showed that wearable sensors are not worn at all times [4] many fall incidents remain undetected when using this technology. Contactless methods, such as the in this research used camera-based system, can overcome this limitation as no user intervention is needed.

*Corresponding author. E-mail: glen.debard@thomasmore.be.

¹Thomas More Kempen, MOBILAB, Geel, Belgium

²KU Leuven Technology Campus Geel, AdvISE, Belgium

³KU Leuven, ESAT-PSI, Leuven, Belgium

⁴KU Leuven Campus De Nayer, EAVISE, Sint-Katelijne-Waver, Belgium

⁵iMinds Multimedia Technology Department, Leuven, Belgium

⁶KU Leuven, ESAT-STADIUS, Leuven, Belgium

⁷iMinds Medical Information Technology Department, Leuven, Belgium

This work is funded by the FWO via project G039811N: "Multi-camera human behavior monitoring and unusual event detection", by the IWT via TETRA project 80150 "Fallcam", by the EU via ERASME (FP7) project IWT 100404 "AMACS" and by the iMinds FallRisk project. The authors would also like to acknowledge the ProFouND network and the COST AAPELE network.

Because of this advantage, several research groups have focused on the development of camera-based fall detection algorithms. In this research two different approaches are most commonly used. Firstly there are the algorithms who detect unusual events, like changes in the person's life pattern [7], [8]. These algorithms rely on indirect evidence, such as prolonged inactivity at unusual locations, to infer fall incidents. As a certain amount of time is needed to detect abnormal inactivities these systems typically have a longer response time. Secondly there are those trying to detect the action of falling directly [5], [6] having a shorter response time. For this fall action detection the most frequently used approach is a combination of low-level cues combined with available domain knowledge (e.g., exploiting the high speed or posture changes of a person during a fall). In these cases, firstly background subtraction is often applied to find the moving foreground objects, including the person [5], [6], [7], [9], [10], [11]. After this, domain knowledge is used to implement simple yet robust fall features, such as the aspect ratio of the foreground region [5], [10], or the speed of the head of a person [5].

Although it is often assumed that these background subtraction techniques result in an accurate silhouette extraction of the person, (e.g. [5], [6], [9]) this is not always the case. Other objects and persons, occlusions, and changing illumination conditions often interfere with the segmentation [12].

These problems could be overcome by using a tracker to follow the person. Several approaches are currently researched such as the use of a $\alpha - \beta - \gamma$ -filter combined with a hybrid matching process based on spatio-temporal templates and color features [6], the use of an overhead view in which a particle filter tracks the person using background subtraction as input [8] or the use of a Kalman filter used to track the head of a person through a combination of background subtraction and ellipse and skin color matching [9]. Although these approaches show promising results they rely on the use of color images. The color in an image is however influenced by the source and amount of the light present in the room which can change during the day.

For our research we chose not to rely on color information but to use gray-level images enabling the use of near-infrared images during the night. Hence we use a particle filter based on three coefficients namely foreground, weighted structural histogram, and person detection.

In order to assess the advantages of the tracker based approach the in this paper presented research compares the performance of our state-of-the-art fall detection algorithm when using only background subtraction, with the usage of

a particle filter to track the person. The results of these methods are also compared with the results of a hybrid method in which the particle filter is only used to enhance the background subtraction and not for the feature extraction.

II. METHODS

The performance of three different algorithms aimed to calculate the input of the fall feature calculation algorithm were compared to each other. As most algorithms described in the literature only use background subtraction to extract the features we firstly used our foreground detection method without the usage of the particle filter, detecting the biggest object visible in the image, as described in [12] as input for the fall detection feature calculation. Subsequently we tested the fall detection algorithm using the prediction of the particle filter as input to calculate the same features. Lastly we used the particle filter as feedback for the background subtraction method. The fall detection features were again calculated using the biggest object in the image.

As more detailed information about our foreground detection method used in the first experiment can be found in [12] we will not discuss this in this section. We will however firstly discuss the implementation of our particle filter and the used measurement coefficients. This is followed by the construction of the feature vectors used for the fall detection. At the end of this section, we describe the used data set.

A. Particle Filter

Particle filters [13] estimate the probability distribution $p(S_t|Z[1..t])$ of the state vector S_t of the tracked object given Z_t representing all the observations. This probability can be approximated from a set of N weighted samples or particles, 50 in our case. Each particle of our filter is an ellipse, with centre coordinates, height, width, and angle and the rate of change of each of these values. The weight of each particle is based on foreground, weighted structural histogram, and person detection coefficients. Fig. 1 shows an overview of the used algorithm. We use a Bootstrap filter implemented using the Bayesian Filtering Library [14]. The tracker was initialized when a foreground object of over 4000 pixels was detected.

1) *Foreground Coefficient C_F* : The foreground coefficient used the foreground that was segmented from the image using background subtraction based on an approximate median filter. To remove shadows, we used cross correlation. After this, an erosion/dilation step was used on all foreground pixels to remove small noisy patches.

A high value for the foreground coefficient was given to an ellipse that covers the detected foreground without including too much background or excluding too much foreground. C_F was calculated with the following formula:

$$C_F = 1.2 \times \frac{FG_{BE}}{Z_{BE}} - 0.6 \times \frac{FG_{LO}}{Z_{LO}} - 0.4 \times \frac{FG_{LOO}}{Z_{LOO}} \quad (1)$$

with FG_{BE} the amount of foreground pixels contained in the bounding ellipse (BE), Z_{BE} the surface of BE, LO is a first layer surrounding BE measuring 1.5 times BE, LOO is a second surrounding layer twice the size of BE.

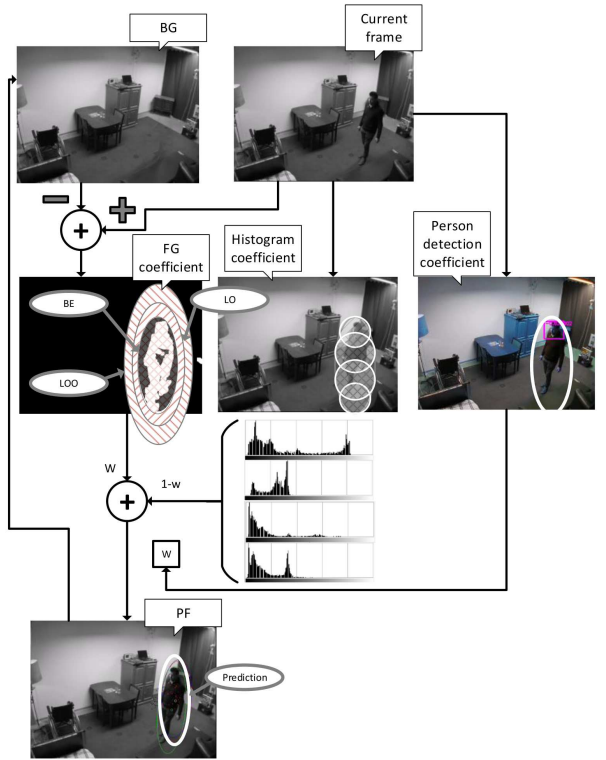


Fig. 1. Overview of implementation of particle filter (PF) with the different measurement coefficients. (BG: background; FG: foreground; BE: bounding ellipse; LO: first outer layer; LOO: second outer layer; W: weightfactor)

2) *Weighted Structural Histogram Coefficient C_H* : The second measurement function for our particle filter was based on histogram matching of the bounding ellipse around the person. Mostly a color histogram is used for this, but since we also used near-infrared during the night, we only could use grayscale values. To make the histogram more distinctive, we used a weighted structural histogram. To measure the correspondence of a given particle, we used correlation with a histogram model H_M . At initialization H_M was calculated from the biggest object. During tracking, H_M was updated every frame with 5% of the current prediction. To calculate the histogram of this particle, we divided the bounding ellipse in four overlapping circles, representing the head and shoulders, the chest, the abdomen and hips, and the lower legs. For each circle a histogram was calculated in which the center had a higher weight than the edges. One exception to this rule was the circle containing the legs. Since legs move a lot and the center could therefore contain background information, the weights were evenly distributed over the whole circle. We calculated this measure for each part of the ellipse and combined them as given in

$$C_H = 0.3 \times C_{head} + 0.35 \times C_{chest} + 0.25 \times C_{abdomen} + 0.1 \times C_{legs} \quad (2)$$

3) *Person Detection Coefficient C_D* : To stop erroneously tracking other objects, we used an upper-body detector [15]. To limit the processing time the detection was only executed every five frames on the region of the detected foreground

object. When a detection was available, H_M was updated accordingly to the score of the detection. Also some low weight particles were replaced by this detection. This is normally not done in the particle filter paradigm, but it increased the robustness of the tracker.

4) *Final Coefficient*: Finally, the total coefficient was calculated as a combination of C_F , C_H and C_D . The initial formula for the final coefficient was given by

$$C_{total} = 0.65 \times C_F + 0.35 \times C_H \quad (3)$$

When a detection was available, the weight was shifted to C_H according to the score of the detection, with a limit of 0.25 for C_F and 0.75 for C_H . When no detection was available, the weight was gradually shifted back to C_F again. If no detection was available for 20 frames, the initial weight was used again.

5) *Predicted State Of Person*: The particle filter represents a probability density. The predicted state of the tracked person was calculated as the mean of the five best predictions. This provided a more stable prediction than only using the particle with the highest weight or using a weighted mean of all particles. For the latter, if there are two peaks in the distribution, a state in between of these two peaks is given as a prediction. This predicted ellipse was also used as feedback for the update of the background discussed in Sec. II-A.1. The background was updated very fast outside of this ellipse and more slowly inside of it. This reduced the formation of a ghost figure while other changes (e.g., changes in lighting, other moving objects...) were integrated faster.

B. Fall Detection Features

Five features were extracted to detect a fall: aspect ratio (AR) [5], [10], change in AR (CAR), fall angle (FA) [7], center speed (CS) [7], and head speed (HS) [5]. These features were also the most widely used in the literature. More information on how to calculate these can be found in [12].

As Noury *et al.* [16] states a fall consists of four phases: the pre-fall, critical, post-fall, and recovery phase. To cover this, we also included information from before and after the fall. We created a feature vector that contained the mean and maximum values of the five fall features over different time slots from before, during, and after the fall. For each time slot of one second such a feature vector was created.

These feature vectors were then used to train a Support Vector Machine (SVM) to classify them as a fall or non-fall. To prevent classifying all vectors as non-fall, different weights were used for positive (w) and negative data ($1 - w$). To find the best combination of the weight w and the regularization parameter of the SVM, we used the F_β -measure as cost-function. This allowed us to give appropriate weights to the sensitivity (SENS) and positive predictive value (PPV).

$$SENS = TP / (TP + FN) \quad (4)$$

$$PPV = TP / (TP + FP) \quad (5)$$

$$F_\beta = (1 + \beta^2) \times \frac{PPV \times SENS}{\beta^2 \times PPV + SENS} \quad (6)$$

TABLE I

RESULTS FOR $\beta = 4$. (A) BIGGEST BOUNDING BOX (BB), NO TRACKER USED; (B) BB PREDICTED BY TRACKER; (C) TRACKER USED FOR BACKGROUND UPDATE, BIGGEST BB USED FOR FALL DETECTION.

Method	SENS	PPV	TP	FN	FP	AUC
(A)	58.7 %	31.4 %	27	19	59	0.526
(B)	65.2 %	43.5 %	30	16	39	0.695
(C)	76.1 %	41.2 %	35	11	50	0.764

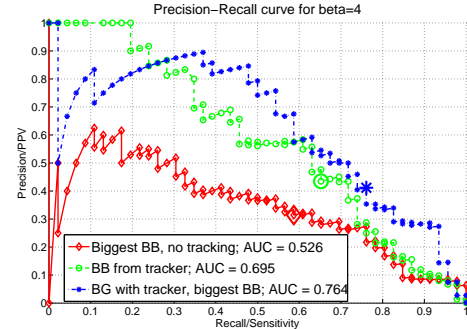


Fig. 2. Precision-Recall curve for $\beta = 4$. The larger markings indicate the trained model without changes of the threshold. (BB = Bounding Box, AUC = Area Under Curve)

The detector was evaluated using ten-fold cross-validation over the complete data set. To reduce the false alarm rate, we first executed a median filter over all time slots of each video to remove single detections, followed by non-maximum suppression to group bursts of detections.

C. Data set

During previous research we monitored seven older persons for several months at their place of residence. During this period, we collected an extensive real-life dataset. However due to privacy concerns, this real-life dataset can not be made publicly available. In order to bridge the gap between currently existing simulation datasets and this real-life one, we created a dataset that re-enacts these falls together with several causes for false alarms that were identified before [12].

For this purpose one lab room was furnished as a nursing home room. The room was further equipped with five wall-mounted IP camera's. Ten different actors were used to re-enact these fall incidents and other Activities of Daily life. From this dataset we used 70 videos, 46 containing one fall each and 24 containing numerous other activities. The fall recordings are each approximately three minutes long. The other recordings are 8.5 hours long in total. This complete simulation dataset will be made publicly available in the near future.

III. RESULTS

Table I shows the results of the three experiments for $\beta = 4$ for the F_β -measure on our simulation data set. Fig. 2 shows the corresponding Precision-Recall curves.

Using no tracker and only relying on background subtraction to follow the person gave the worst results. Only 27 out of 46 falls were detected and 59 false alarms were generated. When using the full prediction of the particle filter, 30 falls were detected while generating 39 false alarms. The best results were found when using the particle filter to enhance the foreground segmentation and then extracting the features from the biggest object in the image. In this case, 35 falls were detected while 50 false alarms were generated. This gave a sensitivity of 76.1% and a PPV of 41.2%. The area under the curve (AUC), which gives a measure of how well the classifier is working, was also the highest with a score of 0.764.

Although these results show an important improvement in fall detection performance can be achieved using tracker data, not all fall incidents were detected due to various reasons which were among others: low fall speed and the formation of a ghost figure that interfered in the feature calculation.

28% of the false alarms were caused by the person leaving or entering the field of view of the camera from below. This caused rapid changes in size and aspect ratio of the BB. Switching the light on or off originated in 20% of the false alarms. In 14% there were two persons or objects moving in the image. Another 12% of the errors was caused by the continuous update of the background, which included non-moving persons in the background and caused a ghost figure. The final 10% of the known causes was because of a misclassification of bending over or sitting on the knees. Correctly detecting if a person leaves or enters the view could therefore reduce the number of false alarms. This could be done by checking the prediction at the edge or out of the image plane. Also detecting large light changes by monitoring the average intensity level can reduce the false alarm further.

IV. CONCLUSION

In this paper the fall detection results of a state-of-the-art fall detector were compared using three different inputs. The best results were obtained using a particle filter to enhance the foreground segmentation and then extracting the features from the biggest object in the image. Visual inspection of the foreground detected using the particle filter showed a definite improvement. The integration of a particle filter improved the performance and robustness of a fall detection algorithm. However using the tracker only for the background update gave better results than using it for the calculation of the fall detection features itself. This could be explained by the fact that a tracker smooths out sudden abrupt movements. Increasing the reaction speed of the tracker would cause it to better follow the falls, but it also would cause the tracker to loose track more often.

These results show us that camera-based fall detection is still a very challenging topic. Certainly the broad range of different falls and difference in fall speed has to be taken into account. We believe that the usage of our particle filter is a good step forward. The particle filter is able to track a person

consistent through the room, while background subtraction is able to discriminate the fast movements well. However still 24% of the falls remain undetected, while a large amount of false alarms were generated. A combination of detecting unusual events, like mentioned in the introduction, using the output of the particle filter and our hybrid solution using background subtraction with the feedback of the particle filter, could again bring us a step closer to solving the fall detection challenge using a camera system.

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] SeniorWatch, "Fall detector: Case study of european ist seniorwatch project," tech. rep., SeniorWatch, 2001.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] M. Shoab, 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.
- [10] 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.
- [11] 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.
- [12] G. Debar, 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.
- [13] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2), pp. 174 – 188, 2002.
- [14] K. Gadeyne, "BFL: Bayesian Filtering Library." <http://www.orocos.org/bfl>, 2001.
- [15] M. Eichner, M. Marin-Jimenez, A. Zisserman and V. Ferrari, "Articulated human pose estimation and search in (almost) unconstrained still images," in *ETH Zurich, D-ITET BIWI Technical Report No 272*, pp. 52–57, 2010.
- [16] 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.