Vis Comput (2017) 33:1529–1540
DOI 10.1007/s00371-016-1296-y

ORIGINAL ARTICLE

Percentage of human-occupied areas for fall detection from two views

Mikaël A. Mousse[1,2] · Cina Motamed[1] · Eugène C. Ezin[2]

Published online: 26 July 2016

Abstract Falls are the major causes of fatal injury for the elderly population. To remedy this, several elderly people monitoring systems with fall detection functionality have been proposed. In this work, we investigate a video-based method of detecting fall incidents from multiple cameras. Our goal is to propose a novel method to detect falls on the floor with a multiple-camera system using the percentage of human-occupied areas. We suggest the use of two relatively orthogonal views to estimate the percentage of the surface of the person which is in contact with the ground according to the foreground information of each camera. These features are computed to differentiate by an automatic manner the lying on floor posture which can be considered as fall to other position such as standing up or sitting. This method is evaluated on a public multi-view fall detection dataset which contains videos of a healthy subject who performed 24 realistic scenarios. These scenarios show 22 fall events and 24 confounding events. The results of our experiments show that our proposed algorithm achieved 95.8 % sensitivity and 100 % specificity with less computational costs than state-of-the-art methods.

Keywords Fall detection · Multi-cameras · Human behaviors analysis · Video surveillance

1 Introduction

Analyzing the frames of an input video using the computers to extract any unusual activity is the principal task of an automatic video surveillance system. Vishwakarma and Agrawal [1] present some activity recognition and behavior understanding using an automatic video surveillance system. One of the important human behaviors is the fall. Indeed, the risk of falling rises when people become older. In fact, according to Gillespie et al. [2] 30 % of people over 65 years old fall each year. The fifth of these fall events requires medical intervention because they cause serious injuries such as fractures and head injuries. [3,4]. After the fall, people cannot always contact emergency services. Then fall problem becomes more important for elderly people monitoring. Automatically detecting falls is an essential part of a system for maintaining elderly person. It can enable rapid response against falls and minimize additional complications from a long period in a fallen position. It also can alert caregivers of the need for preemptive measures for a patient. Thus, if automatic fall detection system is accurate, appropriate measures will be taken quickly to accelerate and improve the medical care provided.

It, therefore, becomes essential to understand the characteristics of the fall. We are in the presence of fall when the person suddenly leaves from a normal posture (such as sitting or standing) to lying on the ground position for an extended period. We distinguish two typical scenarios of fall activities [5]:

✉ Mikaël A. Mousse
mousse@lisic.univ-littoral.fr

Cina Motamed
motamed@lisic.univ-littoral.fr

Eugène C. Ezin
eugene.ezin@imsp-uac.org

[1] EA 4491, LISIC, Laboratoire d'Informatique Signal et Image de la Côte d'Opale, Université Littoral Côte d'Opale, 62228 Calais, France

[2] Unité de Recherche en Informatique et Sciences Appliquées, Institut de Mathématiques et de Sciences Physiques, Porto-Novo, Benin

- falling from sleeping or sitting. This fall occurs when people try to get up or stand up. It may due to the dizziness or syncope.
- falling fall from standing or walking. This fall occurs when people perform daily activities (carrying objects, doing housework, etc.) and lost their balance.

It is also essential to discriminate fall from like-fall events. The event such as sitting down brutally on a sofa, and kneeling on the ground should not be considered as fall. Challenges of indoor video surveillance like dynamic lighting conditions, low difference between human appearance and background, and occlusion also pose considerable difficulties in the implementation of a video surveillance system for automatic fall detection.

In this research work, we are interested in suggesting an efficient video surveillance system for automatic fall detection. Our goal is to obtain some reasonable results quickly. Our principle contributions in this paper are:

1. We propose using the estimation of the surface of the person which is in contact with the ground to establish the model that distinguishes lying on the ground position to other positions.
2. We use two relatively orthogonal views to estimate the surface of the person which is in contact with the ground according to the foreground information of each camera. This estimation is performed using homographic projection. In a case of overlapping cameras, homography consists in finding a matrix which corresponds a points $pt(x, y)$ of a camera view to another point $pt'(x', y')$ of another camera view.

This paper consists of five sections. The related works are presented in Sect. 2. Section 3 describes the details of our proposed method. A presentation of the experiment environment and the performance comparison are reported in Sect. 4. We conclude the work in Sect. 5.

2 Related works

Fall detection methods can be subdivided into three categories. The algorithms of the first category are based on wearable devices whereas algorithms in the second category used the ambiance sensors and the third category used cameras. The ambient device-based approaches use pressure sensors to detect and track the subject. The detection and the tracking are based on the subject's weight which is obtained using the pressure sensor. The implementation of these systems is cost effective and less intrusive. But they have a big disadvantage. In fact, the detecting of any other pressure in and around the subject creates a false alarm in case of fall detection. That causes a low fall detection accuracy. To overcome these limitations, some researchers propose to use a computer vision system to detect fall. This kind of system has two major advantages. First, it does not require that the person wears anything. Second, a camera gives more information on the motion of a person and his/her actions than other devices. Thus, a camera-based system does not provide only information on falls, but it also gives information on daily behaviors (medication intake, meal, sleep time and duration, etc.). For that reasons, we propose a method using the cameras. Then, we focus our related work on camera-based methods. A complete review on fall detection algorithms is provided by Mubashir et al. [6].

Most of the research works concerning fall detection relies on a single-view approach [7–11]. This is due to the availability of a single camera surveillance system and to the implementation of these systems which is easy. But, it is difficult to detect efficiently falls of people from a single simple camera view. When the system works well, its complexity is often very large and it takes strong assumptions. To overcome this drawback, several research works propose to use depth cameras which are placed on the ceiling [12–17]. Some research works proposed to use multi-cameras system for people fall detection. The multi-cameras systems take a lot of scope in an automatic visual surveillance system. Indeed, they can serve efficiently to monitor and to supervise significant sites, to control and to estimate flows (car parks, airports, ports, and motorways). Because of the fast growing of data processing, communications and instrumentation, such applications become possible. This kind of systems requires more cameras to cover overall field-of-view. Using several views of the same scene (multi-view) can allow to recover the information that could have been hidden in a specific view and consequently the effects of objects dynamic occlusion are reduced. In this category, Auvinet et al. [18,19] propose to reconstruct the 3D shape of the person for fall detection using a network of multiple calibrated cameras. They extract a feature which represents the volume distribution along the vertical axis. Fall events are then detected by analyzing this feature. They trigger an alarm when the major part of this feature is abnormally near to the floor. In a later work, they define a period of time t before triggering the alarm [20]. Then, the fall alarm is triggered when the major part of the volume distribution along the vertical axis is abnormally near the floor during t. Other researchers have also used the 3D reconstruction to detect fall. For example, Anderson et al. [21,22] have also used 3D reconstruction. They suggest a hierarchy of fuzzy logic to detect falls. Their method consists of two levels. The first level extracts the states of the person at each frame whereas, the second level deals with linguistic summaries of the subject's states called "Voxel Person". The two levels are fused using a fuzzy logic system. These methods provide good results, however, 3D reconstruction process requires more processing time. To overcome this drawback, some researchers have proposed methods with less computational cost. Thome et al. [23] suggest a layered hidden Markov model (LHMM) for modeling motion with fall detection. This modeling is performed with two layers. The first layer models two postures, an upright standing pose and lying pose. They observe the 3D angle relationships and perform an initial image metric rectification. They deduce some theoretical properties from binding the error angle for a standing posture. This differentiates other posture as “non-standing” ones. Thus, falls are accurately detected from other actions, such as walking or sitting. Cucchiara et al. [24] suggest a multiple cameras system to monitor different rooms of a smart home. Each room is controlled by a single camera. They carried out analysis of human behaviors by classifying the posture of the subject and consequently detecting falls using the projection histograms. These histograms are calculated and compared with the stored posture maps which are obtained after training. Cucchiara et al. also propose a tracking algorithm which deals with occlusions. Zweng et al. [25] also use multiple cameras for fall detection without external camera calibration. For each camera of the network, they associate a fall confidence to the subject. Finally, they fuse all single decisions to make an overall fall detection decision. Hung et al. [26,27] propose the using of the measures of humans heights and occupied areas to distinguish three states of humans which are standing, sitting and lying. They prove that two relatively orthogonal views are sufficient to estimate the occupied areas and the height.

In this work, we propose a high-level fusion information for fall detection which is not based on 3D reconstruction with less computational cost. In our method, individual cameras do not extract features but provide foreground bitmap information to the fusion center. Then a fall detection decision is taken by the fusion center using the estimation of the percentage of people surface which is in contact with the ground. In this work, we decided to use the information from two cameras because when we have two cameras with complementary views, they provide sufficient information for decision-making. We choose two relatively orthogonal cameras. Indeed, the estimation of the percentage of human’s surface which is in contact with the ground is highly sensitive to both the occlusions that had frequently happened in the indoor environments mainly by the furniture, the false-positive detection and the true-negative detection. However, under two relatively orthogonal views, we realize that people are partially occluded in one view but likely visible in the other one. The aim of our method is to use the surface of contact between the individual and the ground to detect the fall. The use of the multi-cameras system also reduces the errors because we exploit more information to take a decision. Using Fig. 1, we note that when the two cameras views are complementary, the intersection of the homographic projection of their foreground pixels into reference approximates the contact surface between the individual and the ground. We use this information to extract important features for the detection of the fall.

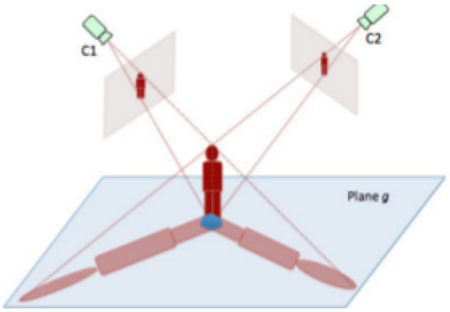

Fig. 1 Person viewed by two overlapping cameras

3 Proposed algorithm for fall detection

In this section, we present our proposed approach. Figure 2 presents the architecture of our proposed system. According to this figure, our fall detection method is divided into five modules:

- single foreground maps detection: this module identifies the foreground pixels of each camera view;
- foreground information fusion: this module merges the two foreground pixels obtained using the first module to get a global information of the scene;
- features extraction: this module extracts some surfaces to characterize the posture of the person. The goal is to discriminate lying on the ground position to other postures;
- tracking: this module is adopted to support people fall detection, by keeping track of people movements and of their identities along time;
- decision-making: this module identifies if the person is in critical position or not.

These modules are described below.

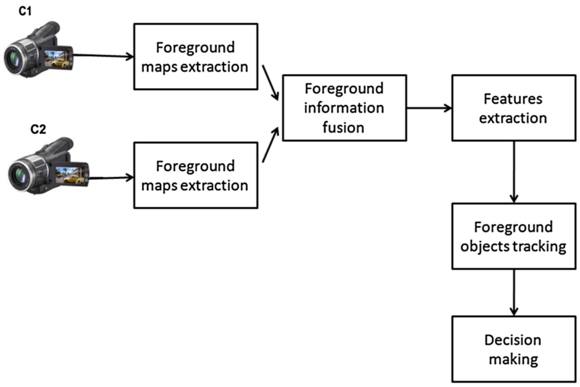
C1
C2
Foreground maps extraction
Foreground maps extraction
Foreground information fusion
Features extraction
Foreground objects tracking
Decision making

Fig. 2 Architecture of our proposed system

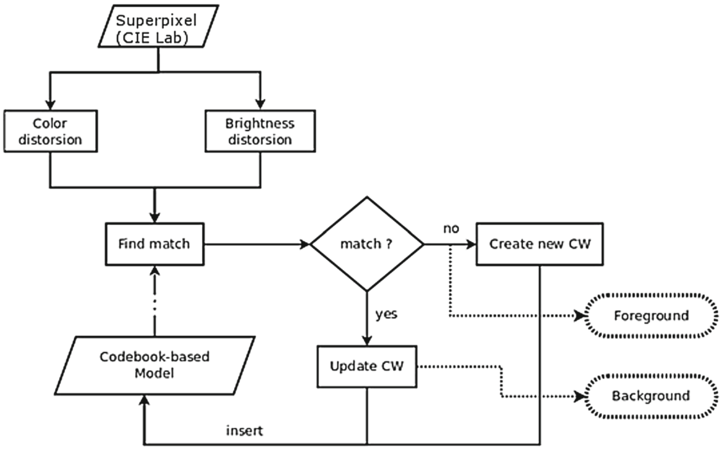

Fig. 3 Flow diagram of the moving object extraction algorithm (the *solid lines* correspond to the learning phase; *dashed lines* correspond to the phase of moving objects detection)

3.1 Foreground map detection

The first step of our fall detection method is the extraction of the foreground map of each camera. This extraction is done in two steps. The first step is the extraction of the foreground pixels and the second is the grouping of these pixels using polygons. The purpose of this grouping is the reduction of information which will be processed. For the foreground pixels detection, we use a background modeling-based algorithm. In fact in a single camera system, many algorithms about object detection exist with different purposes. These algorithms are subdivided into three categories: without background modeling, with background modeling and combined approach. Algorithms based on background modeling are recommended in case of dynamic background observed by a static camera. These algorithms are also robust in the case of illumination variation. In this work, we use the algorithm proposed by Mousse et al. [28] because they proved the efficiency and the efficacy of this algorithm. We chose this algorithm because it has a good performance in moving object detection. This method integrates a region-based information using superpixel segmentation algorithm into the original codebook algorithm and uses CIE L*a*b* color space information. Figure 3 represents the flow diagram of the codebook-based algorithm. For each frame, after the extraction of superpixels, we build a codebook background model. Let $P = \{s_1, s_2, \ldots, s_k\}$ represent the K superpixels obtained after superpixels segmentation. Each superpixel $s_j, j \in \{1, 2, \ldots, k\}$ is composed by m pixels. With each superpixel, we build a codebook $C = \{c_1, c_2, \ldots, c_L\}$ which contains L codewords $c_i, i \in \{1, 2, \ldots, L\}$. Each codewords c_i consists on an vector $v_i = (\bar{a}_i, \bar{b}_i)$ and 6-tuples $\text{aux}_i = \{\check{L}_i, \hat{L}_i, f_i, p_i, \lambda_i, q_i\}$ in which $\check{L}_i, \hat{L}_i$ are the minimum and maximum of luminance value, f_i is the frequency at which the codeword has occurred, λ_i is the maximum negative run length defined as the longest interval during the training period that the codeword has not recurred, p_i and q_i are the first and last access times, respectively, that the codeword has occurred. $\bar{L}, \bar{a}, \bar{b}$ are, respectively, the average value of component L*, a* and b* in a superpixel. The codebook model is created or updated using two criteria. The first criterion is based on color distortion (1) whereas the second is based on brightness distortion (2).

$$\sqrt{||p_t||^2 - C_p^2} \le \varepsilon_1 \tag{1}$$

$$I_{\text{low}} \le I \le I_{\text{hi}} \tag{2}$$

In (1), the autocorrelation value C_p^2 is given by Eq. (4) and $||p_t||^2$ is given by Eq. (3).

$$||p_t||^2 = \bar{a}^2 + \bar{b}^2 \tag{3}$$

$$C_p^2 = \frac{(\bar{a}_i\bar{a} + \bar{b}_i\bar{b})^2}{\bar{a}_i^2 + \bar{b}_i^2} \tag{4}$$

In (2), $I_{\text{low}} = \alpha\hat{L}_i$, $I_{hi} = \min\{\beta\hat{L}, \frac{\check{L}}{\alpha}\}$ and $I = \bar{L}$.

After the extraction of foreground pixels, we group them into foreground region. The grouping the foreground pixels in a polygon to reduce the data. Indeed when we group the foreground by polygon, we only use the vertices of this polygon during our process. The polygon is obtained by searching the convex hull of all contours detected in threshold image.

Let us consider a set X of point. The convex hull of X is the convex set that contains all points of X. To obtain the convex hull, we search the point of X that belongs to the minimal convex hull. Most of the time, this point is the point of X that has the least x-coordinate. After that we create a list P. In P, we store the numbers of the points and their positions in X. We sort X in increasing order (except for $P[0]$ which is the first point) using a pair-wise comparison of the elements. For the points, the comparison takes into account their left position with regard to the starting $R = XP[0]$ point. If C point is on the left from RB vector then $B < C$. This allows us to have a starting point for our polygon and a potential order in the succession of the vertices of the polygon (confers Fig. 4a). Finally, you cut the angles. Then, we create a list S and place

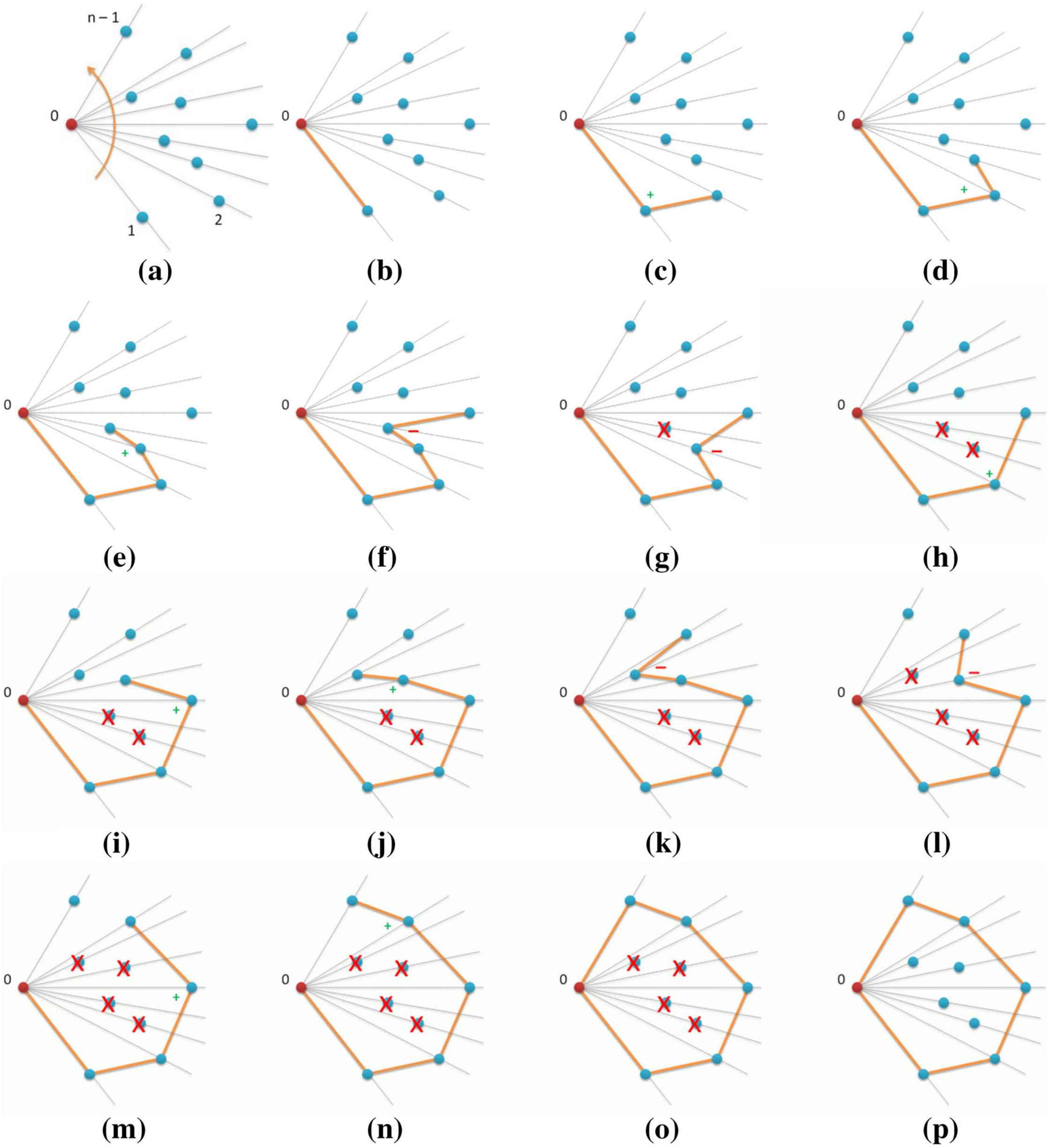

Fig. 4 Example of detection of the convex hull of a set of points

$P[0]$, $P[1]$ into it. We look through all other vertices of P by keeping track of recent three points and find the angle formed by them. If the orientation angle formed with these points is not counterclockwise, we cut it and remove the last vertex from S. Otherwise, if the orientation of the angle is clockwise, we place the current vertex into S. This last step is illustrated by Fig. 4. Then, all regions can be approximated by a polygon and each polygon is convex. Figure 5 presents an example of the approximation of foreground maps by a polygon.

3.2 Foreground maps fusion

After the extraction of the foreground maps, they need to be fused to obtain a more global information. The foreground maps fusion is based on homography. Homographies are usually estimated between a pair of images by finding feature correspondence in these images. To perform the homography mapping, the most commonly used feature is corresponded points in different images, though other features such as lines or conics in the individual images may be used. These fea-

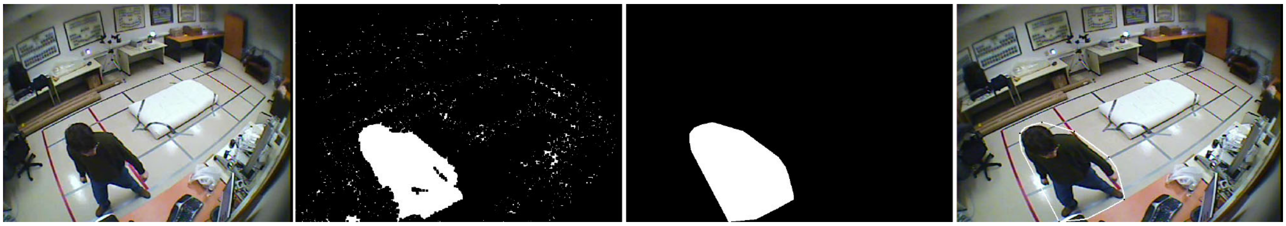

Fig. 5 Result of foreground map extraction. The *first column* shows the input frame, the *second column* shows the foreground pixels detected, the *third column* presents the foreground maps and the *fourth column* presents the polygons

tures are selected and matched manually (or automatically) from 2D images to compute the homography between two camera views or the homography between one camera view and the top view. Then, different views of the same scene are related by a homography that consists of a 3 × 3 matrix which maps points on a plane in one view [29]. In the case of calibrated cameras, homographies can be constructed from calibration parameters [30]. If, instead, such information is not available, the homographies can be estimated. In the latter case, given a set of pixels x_i^j in the jth view I_j^t and a corresponding set of pixels x_i^k in the kth view I_t^k, the task is to compute the projective transformation, a 3 × 3 matrix H, that maps each x_i^k to x_i^j, i.e., $x_i^j = Hx_i^k$ for each i. To obtain H, we use a feature-based approach using scale invariant feature transform keypoints [31] that are extracted from one view and then matched to those extracted from a different view. The homography H is then estimated by means of direct linear transformation [29] and random sample consensus [32] algorithms, that yield an initial guess for H and a list of inlier matches. The initial estimated homography H can be further refined using Levenberg–Marquardt optimization minimizing the re-projection error. Using the principle of the homography, we choose the reference view and we project the polygons from the second view to the reference view. A polygon projection is performed by projecting only the vertices of the polygon. After the projection, the intersection of the polygons represents the moving object which will be tracked.

3.3 Features extraction

After the projection of foreground maps, we extract some surfaces to characterize the posture of the people. For each individual, we compute the surfaces $\omega_i, i \in \{1, 2\}$ of the polygons obtained by projecting the polygons associated with this individual in each camera into the reference view. After that, we also evaluate the surface σ of the polygon which is the intersection of the two projected polygons in the reference plane. With these features we compute $\varrho_1 = \frac{\sigma}{\omega_1}$ and $\varrho_2 = \frac{\sigma}{\omega_2}$. ϱ_1 and ϱ_2, respectively, represent the percentage of surface in contact with the ground detected by the first camera and the second camera. These last two values allow us to assess the posture of an individual. Indeed, when the individual approaches the ground (falling), the values of ϱ_1 and ϱ_2 will be greater (close to 1). Some examples of using ϱ_1 and ϱ_2 for fall detection (based on some scenario taken from the dataset which is used in this paper and described in the Sect. 4.1) are shown by Figs. 6, 7 and 8.

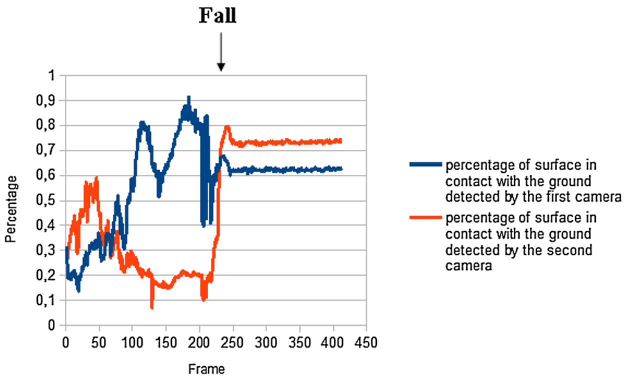

Fig. 6 Example of using of the proposed features for fall detection (scenario 1)

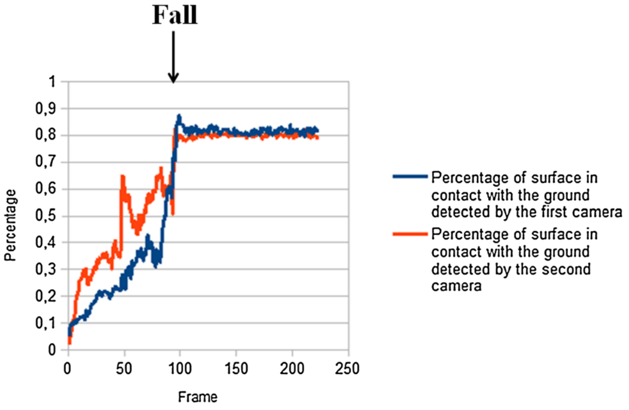

Fig. 7 Example of using of the proposed features for fall detection (scenario 2)

3.4 Tracking

The object tracking is performed by the fusion center. It is based on spatial location of objects. In each frame, the object takes the id of the object from the previous frame that has the

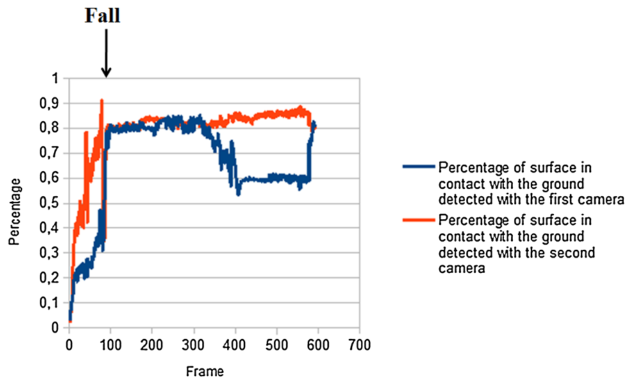

Fig. 8 Example of using of the proposed features for fall detection (scenario 3)

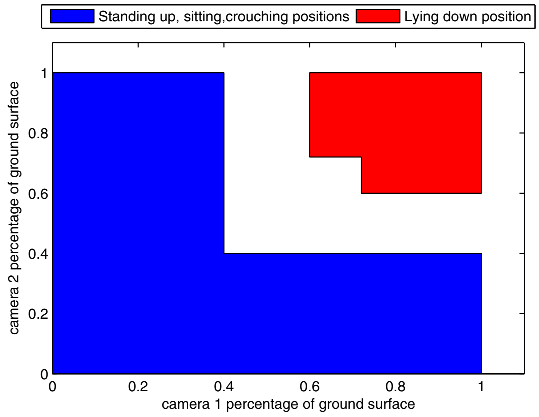

Fig. 9 Position classification

lowest spatial distance. When a new object appears on the scene, the number of objects for an image will be higher than the one in the previous image. Then, we associate a new id to the object which does not correspond to previous objects. When an object leaves the scene, the number of objects will be lower than for the previous one. Then, the previous object which does not correspond to an object in the current image is not considered anymore.

3.5 Decision-making

Using the feature extracted in Sect. 3.3, we classify the posture. This classification is into two groups: lying down position and other positions (standing up, sitting, crouching positions). The values of the thresholds are fixed according to our experiments. During these experiments, we use one video sequence (sequence of the ninth scenario) in the dataset described in Sect. 4.1 to finding the threshold. The choice of this sequence was done according to two major reasons. First, it includes the three most important positions: sitting, standing up and lying on the ground. Second, this sequence is also used for the training step of other research works which exploit this dataset. Using the result of this training step, the posture of the person can be classified according to the values present in Fig. 9. This figure, respectively, shows ϱ_1 values on the x-axis and ϱ_2 values on the y-axis. Thus, for one person, if $\varrho_1 < 0.4$ or $\varrho_2 < 0.4$ then we conclude that this person is in a posture other than lying on ground position. Whereas if the condition (($\varrho_1 \geq 0.6$ and $\varrho_2 \geq 0.72$) or ($\varrho_2 \geq 0.6$ and $\varrho_1 \geq 0.72$)) is true then the person is lying on the ground plane. When a person changes rapidly from the first group of posture to lying on the ground posture, a warning state is attached to him and when he did not come quickly, an alarm is emitted to indicate the fall. In this work, we assume that the person changes rapidly posture when the delay between this change is less or equal to 1.5 s. After the warning state, we propose that the person stays down for 3 s before the alarm is triggered. This delay is important to manage the people who fall and get up afterward. In addition, if after the triggering of the alarm, the posture of the person is other than ”lying down”, we expect 3 s before disabling the alarm. By proceeding like that we limit the false negatives.

The system attempts also to solve the problem of occlusion by objects. Indeed, when the person is occluded by an object, the estimation of ω_1 and/or ω_2 will be biased. But, in an indoor environment, the person is rarely occluded in both two views of the orthogonal cameras. So, one of the values between ω_1 and ω_2 is correct. Then, the features ϱ_1 and ϱ_2 are more reliable because they are obtained after the fusion of ω_1 and ω_2.

4 Experimental environment and performance evaluation

4.1 Experimental environment

In this paper for our experimental results, we use the “Multiview fall dataset” proposed by Auvinet et al. [33], which was adopted in the experiments of many research works. Then, it is possible to compare the performance of our proposed algorithm to some past research works’ performance. Eight inexpensive IP cameras with a wide angle were set up to cover the whole room. The experimental environment is presented in Fig. 10. This dataset contains 24 scenarios. These scenarios show 24 fall incidents and 24 confounding events (11 crouching, 9 sitting, and 4 lying on a sofa). All events are viewed by all the cameras and are performed by one subject. The normal daily activities include walking in different directions, housekeeping, activities with characteristics similar to falls (sitting down/standing up, crouching down) whereas the simulated falls include forward falls, backward falls, falls when inappropriately sitting down, and loss of balance. Falls were performed in different directions with respect to the camera point of view. The sequences also contain confounding events that include crouching, kneeling, carrying objects, doing housework, etc. Camera settings, spatial arrangement, information of multi-cameras synchronization, calibration parameters, and event annotation for all scenarios are provided in their study [33]. The sequences contain also some difficulties which can lead to segmentation errors such as shadows, reflections, variable illumination, and occlusions. Figure 11 presents some examples of typical simulated fall incidents and normal daily activities are shown. For single-view object detection, we use parameters defined in [28]. The experimental environment is Intel® Core™i7 CPU L 640 @ 2.13 GHz × 4 processor with 4 GB memory and the programming language is C++. In this paper, we use video sequences from five pairs of cameras. The first pair is composed by the cameras 1 and 3, the second pair is composed by the cameras 2 and 5, the third pair is composed by the cameras 4 and 7, the fourth pair is composed by the cameras 6 and 3 and the fifth pair is composed by the cameras 5 and 8. Figures 12 and 13 represent some examples of finding polygon intersection in the ground plane using the first pair of cameras (cameras 1 and 3).

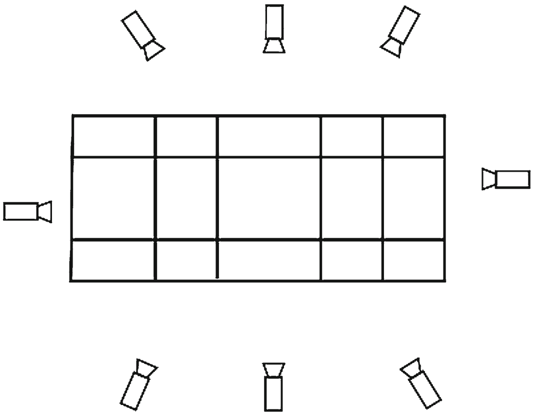

Fig. 10 Experimental environment

4.2 Performance evaluation

In this subsection, we evaluate the performance of our proposed people fall detection and compare it with other research works [8,20,26]. We compute some metrics for testing the efficiency and the accuracy of our algorithm and compare it with the cited papers which are also tested on the same dataset. These metrics are sensitivity Se and specificity Sp.

$$\mathrm{Se} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{6}$$

In (5) and (6), TP is the number of falls correctly detected, FN is the number of falls not detected, TN is the number of normal activities not detected as a fall and FP is the number of normal activities detected as a fall. High sensitivity means that most fall incidents are correctly detected. Similarly, high specificity implies that most normal activities are not detected as fall events. A good fall detection method must achieve high

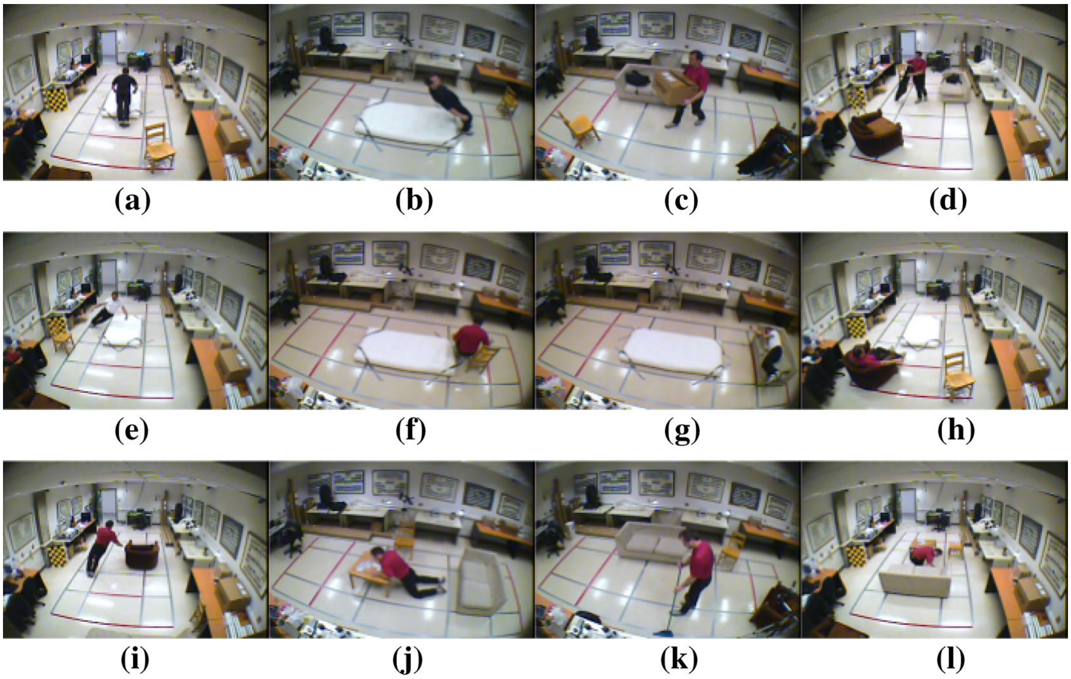

Fig. 11 Examples of falls and normal daily activities

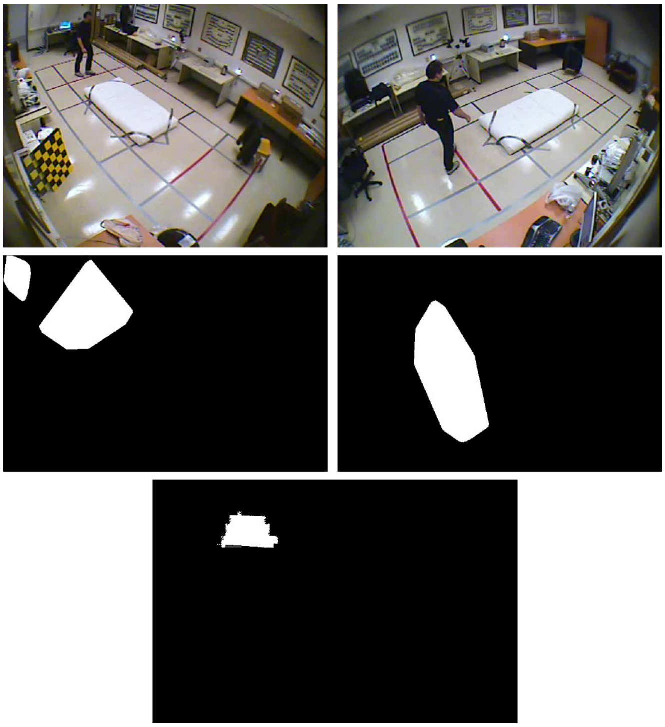

Fig. 12 The *first row* shows the camera views, the *second row* shows the foreground maps detected, the *third row* presents the surface in contact with the ground

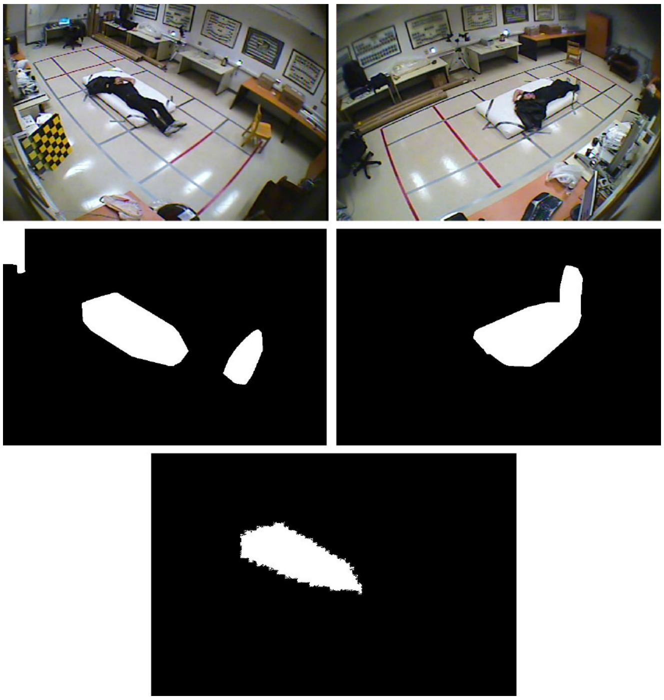

Fig. 13 The *first row* shows the camera views, the *second row* shows the foreground maps detected, the *third row* presents the surface in contact with the ground

values of sensitivity and specificity. The results are reported in Table 1. In this table,

– the results of our method are the same for all pairs of cameras we used in our experiments. This is explained by the fact that the falls are observed by all the cameras;
– the results of the method proposed by Hung and Saito [26] are obtained using cameras 2 and 5;

Table 1 Performance comparison between our method and three state-of-the-art methods, tested on the same dataset

	Sensitivity (Se, %)	Specificity (Sp, %)
Our method	95.8	100
Auvinet et al. [20]	80.6	100
Rougier et al. [8]	95.4	95.8
Hung and Saito [26]	95.8	100

Table 2 Speed comparison

	Hung and Saito [26]	Proposed method
Speed (fps)	10.95	**15.25**

The bold value is the best value

- the results of the method proposed by Auvinet et al. [20] are reported using a network of three cameras. Because of the 3D reconstruction method used by Auvinet et al. it is impossible to have acceptable results with less than three cameras. Auvinet et al. prove that the sensitivity can be boosted to 100 % if a network of more than four cameras is employed.

By comparing algorithms we conclude that our algorithm has similar performance to recent algorithms. Such as Hung and Saito [26] algorithm, our method only fails in the 22nd scenario in which the person is sitting on a chair and suddenly slips to the floor. However, both Auvinet et al. and Rougier et al. methods are of high computational costs because they are based on 3D reconstruction algorithm. 3D reconstruction algorithm also requires a lot of camera view to obtain a good result. Rougier et al. [8] report the implementation of 5 fps and argues that this frame rate is sufficient for detecting fall events. Auvinet et al. [20] present the GPU implementation to realize their method in real time. But our method is implemented in a common desktop which is described in Sect. 4.1. We compare our processing time to the processing time of the method proposed by Hung and Saito [26] which is also implemented in real time on a desktop. The speed comparison is presented in Table 2 and the value is expressed in frames per second. Results of this table are obtained using the same environment (we use the same pair of cameras (camera 2 and camera 5) and the same scenario (scenario 9 of the dataset) during the learning step to extract the thresholds). According to this table, we conclude that our algorithm is of lower computational cost than the algorithm proposed by Hung and Saito [26]. This is due to several aspects:

- the use of superpixels clustering algorithm in foreground pixels extraction module reduces the computational cost of moving object detection;
- the approximation of the foreground pixels using polygons and the fusion of these polygons using homography mapping is less complex than the estimation of width and height proposed by Hung and Saito [26].

Therefore, we conclude that our multi-cameras system uses the smallest number (two) of cameras and has the smallest computational complexity compared with existing methods.

5 Conclusion

We have presented a novel video-based method of fall detection. The proposed approach contains two main components, object detection and the use of a falling model. For object detection, we use a codebook-based method and we approximate the foreground pixel using polygons. This approximation reduces the number of information which will be processed by the fusion process. This fusion is done using homography mapping. For the fall model, we extract a set of features such as the surface of the polygon and the percentage of the surface which is in contact with the ground. Our experimental results using public dataset show that the proposed method can accurately detect a single falling person. The limitations of our method are not unexpected. First, such as all automatic video surveillance systems, our fall detection method is highly dependent on each camera foreground pixel detection. Then the presence of false positive (false detection) and/or false negative (misdetection) can influence the fall detection results. Some errors will also occur if non-human objects appear in the scene.

Acknowledgements This work is partially financially supported by the Association AS2V and Fondation Jacques De Rette, France. Mikaël A. Mousse is grateful to the “Service de Coopération et d’Action Culturelle de l’Ambassade de France au Bénin”. The authors also appreciate the comments provided by the reviewers as these have improved the manuscript.

References

1. Vishwakarma, S., Agrawal, A.: A survey on activity recognition and behavior understanding in video surveillance. Vis. Comput. **29**(10), 983–1009 (2013)
2. Gillespie, L., Gillespie, W., Robertson, M., Lamb, S., Cumming, R., Rowe, B.: Interventions for preventing falls in elderly people. Cochrane Database Syst. Rev. **3** (2003)
3. Friedman, S.M., Munoz, B., West, S.K., Rubin, G.S., Fried, L.P.: Falls and fear of falling: which come first? A longitudinal prediction model suggests strategies for primary and secondary prevention. J. Am. Geriatr. Soc. **50**, 1329–1335 (2002)
4. Hindmarsh, J.J., Estes Jr., E.H.: Falls in older persons: causes and interventions. Arch. Intern. Med. **149**(10), 2217–2222 (1989)
5. Yu, X.G.: Approaches and principles of fall detection for elderly and patient. IEEE Int. Conference on e-Health Networking, Applications and Services, pp. 42–47 (2008)
6. Mubashir, M., Shao, L., Seed, L.: A survey on fall detection: principles and approaches. Neurocomputing (2012)
7. Mirmahboub, B., Samavi, S., Karimi, N., Shirani, S.: Automatic monocular system for human fall detection based on variations in silhouette area. IEEE Trans. Biomed. Eng. **60**, 427–436 (2013)
8. Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Robust video surveillance for fall detection based on human shape deformation. IEEE Trans. Circuits Syst. Video Technol. **21**, 611–622 (2011)
9. Feng, W., Liu, R., Zhu, M.: Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera. Signal Image Video Process. **8**(6), 1129–1138 (2014)

10. Liao, Y.T., Huang, C.L., Hsu, S.C.: Slip and fall event detection using bayesian belief network. Pattern Recognit. **45**(1), 24–32 (2012)
11. Charfi, I., Miteran, J., Dubois, J., Atri, M., Tourki, R.: Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and adaboost-based classification. J. Electron. Imaging **22**(4), 041106 (2013)
12. Rougier, C., Auvinet E., Rousseau J., Mignotte M., Meunier J.: Fall detection from depth map video sequences. International Conference on Smart Homes and Health Telematics, pp. 121–128 (2011)
13. Kepski, M., Kwolek, B.: Fall detection using ceiling-mounted 3d depth camera. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pp. 640–647 (2014)
14. Stone, E., Skubic, M.: Fall detection in homes of older adults using the microsoft kinect. IEEE J. Biomed. Health Inf. **19**, 290–301 (2014)
15. Zhang, Z., Liu, W., Metsis, V., Athitsos, V.A.: Viewpoint-independent statistical method for fall detection. IEEE Conference on Pattern Recognition, pp. 3626–3630 (2012)
16. Mastorakis, G., Makris, D.: Fall detection system using Kinect's infrared sensor. J. Real-Time Image Process. **9**(4), 635–646 (2014)
17. Bian, Z., Hou, J., Chau, L., Magnenat-Thalmann, N.: Fall detection based on body part tracking using a depth camera. IEEE J. Biomed. Health Inf. (2014)
18. Auvinet, E., Reveret, L., St-Arnaud, A., Rousseau, J., Meunier, J.: Fall detection using multiple cameras. Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE, pp. 2554–2557 (2008)
19. Auvinet, E., Multon, F., St-Arnaud, A., Rousseau, J., Meunier, J.: Fall detection using body volume reconstruction and vertical repartition analysis. International Conference on Image and Signal Processing, pp. 376–383 (2010)
20. Auvinet, E., Multon, F., St-Arnaud, A., Rousseau, J., Meunier, J.: Fall detection with multiple cameras: an occlusion-resistant method based on 3-d silhouette vertical distribution. IEEE Trans. Inf. Technol. Biomed. **15**, 290–300 (2011)
21. Anderson, D., Luke, R., Keller, J., Skubic, M., Rantz, M., Aud, M.: Linguistic summarization of video for fall detection using voxel person and fuzzy logic. Comput. Vis. Image Understand. **113**(1), 80–89 (2009)
22. Anderson, D., Luke, R.H., Keller, J.M., Skubic, M., Rantz, M.J., Aud, M.A.: Modeling human activity from voxel person using fuzzy logic. IEEE Trans. Fuzzy Syst. **17**(1), 39–49 (2009)
23. Thome, N., Miguet, S., Ambellouis, S.: A real-time, multiview fall detection system: a lhmm-based approach. IEEE Trans. Circuits Syst. Video Technol. **18**(11), 1522–1532 (2008)
24. Cucchiara, R., Prati, A., Vezzani, R.A.: multi-camera vision system for fall detection and alarm generation. Expert Syst. **24**(5), 334–345 (2007)
25. Zweng, A., Zambanini, S., Kampel, M.: Introducing a statistical behavior model into camera-based fall detection. International Conference on Advances in Visual Computing, pp. 163–172 (2010)
26. Hung, D.H., Saito, H.: The estimation of heights and occupied areas of humans from two orthogonal views for fall detection. IEEJ Trans. Electron. Inf. Syst. **133** (2013)
27. Hung, D.H., Saito, H., Hsu, G.S.: Detecting fall incidents of the elderly based on human-ground contact areas. 2nd IAPR Asian Conference on Pattern Recognition, pp. 516–521 (2013)
28. Mousse, M.A., Motamed, C., Ezin, E.C.: Fast Moving Object Detection from Overlapping Cameras. International Conference on Informatics in Control, Automation and Robotics, pp. 296–303 (2015)
29. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
30. Tsai, R.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE J. Robotics Autom. **3**, 323–344 (1987)
31. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**, 91–110 (2004)
32. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**, 381–395 (1981)
33. Auvinet, E., Rougier, C., Meunier, J., St-Arnaud, A., Rousseau, J.: Multiple cameras fall data set Technical Report 1350. DIRO Université de Montréal. (2010)

Mikaël A. Mousse received his Bachelor Engineering degree in computer science applied to management in 2008 from Ecole Nationale d'Economie Appliquée et de Management (Benin) and his Master degree in Computer Engineering and Applied Sciences in 2012 from Institut de Mathématiques et de Sciences Physiques (Bénin). He is currently completing his Ph.D degree in Artificial Intelligence and Computer Vision conjointly at Institut de Mathématiques et de Sciences Physiques (Université d'Abomey-Calavi, Bénin) and at Université du Littoral Côte d'Opale (France). He is a Computer Vision Foundation member, he is both affiliated with Unité de Recherche en Informatique et Sciences Appliquées (Benin) and with Laboratoire d'Informatique Signal et Image de la Côte d'Opale (France). His research interests include signal processing, image processing, video processing, computer vision and human behaviors analysis and recognition.

Cina Motamed is an associate professor in Computer Science in the University of Littoral Cote d'Opale, Calais, France. He received his B.Sc. in mathematics, and M.Sc in Electrical Engineering and Computer Science from the University of Caen, France and the Ph.D. degree in Computer Science from the University of Compiegne, France, in 1987, 1989, and 1992, respectively. His current research concerns about the automatic visual surveillance of wide area scenes using computational vision. His research interests focus on the design of multi-camera system for real-time multi-object tracking and human action recognition. He is recently focusing on the uncertainty management over the vision system using graphical models, and beliefs propagation. He is also interested in unsupervised learning approaches for human activity recognition.

Eugène C. Ezin IEEE member in computer society, received his Ph.D. degree in 2001 with highest level of distinction after research works carried out on neural networks and fuzzy systems for speech applications at the International Institute for Advanced Scientific Studies in Italy. Since July 2012, he is an associate professor in computer science in the field of artificial intelligence. He supervised many master theses and some works are ongoing for Ph.D. theses. He is a reviewer for Mexican International Conference on Artificial Intelligence and other journals. His research interests include machine learning, neural networks and fuzzy systems, signal and image processing, cryptography, information system and network security. He is also interested in human activities recognition through multi-sensor systems.