Robust Video Surveillance for Fall Detection Based on Human Shape Deformation

Caroline Rougier, Jean Meunier, Alain St-Arnaud, and Jacqueline Rousseau

***Abstract*—Faced with the growing population of seniors, developed countries need to establish new healthcare systems to ensure the safety of elderly people at home. Computer vision provides a promising solution to analyze personal behavior and detect certain unusual events such as falls. In this paper, a new method is proposed to detect falls by analyzing human shape deformation during a video sequence. A shape matching technique is used to track the person's silhouette along the video sequence. The shape deformation is then quantified from these silhouettes based on shape analysis methods. Finally, falls are detected from normal activities using a Gaussian mixture model. This paper has been conducted on a realistic data set of daily activities and simulated falls, and gives very good results (as low as 0% error with a multi-camera setup) compared with other common image processing methods.**

***Index Terms*—Fall detection, Gaussian mixture model (GMM), novelty detection, Procrustes shape analysis, shape context, video surveillance.**

I. Introduction

ACCORDING to the Public Health Agency of Canada [1], one Canadian out of eight was older than 65 years old in 2001. In 2026, this proportion will be one out of five. Similar figures are observed in other industrialized countries. Faced with the growing population of seniors, developed countries need to establish new healthcare systems to ensure the safety of elderly people at home. Indeed, a majority of seniors, 93%, reside in private homes, and of these, 29% live alone [1]. Moreover, falls are one of the major risks for old people living alone, often causing severe injuries. The gravity of the situation can increase if the person cannot call for help, being unconscious or immobilized.

Most fall detection techniques are based on accelerometers [2]–[4] or help buttons [5]. But the major problem with these types of technology is that older people often forget to wear them, and in the case of a help button, it is useless if the person is unconscious after the fall. Moreover, batteries are needed for these devices and must be replaced or recharged regularly for adequate functioning. Floor vibration-based detectors could be a promising solution but they depend upon the floor dynamics and are still in their infancy [6].

Recently, the emergence of computer vision systems has allowed us to overcome these problems. The main advantage of computer vision systems is that the person does not need to wear any special device. Moreover, a camera provides a vast amount of information on the person and his/her environment. For example, we can extract information on the location, the motion, or the actions of the person. Thus, we can imagine a computer vision system providing information on falls, but also, checking other daily behaviors like medication intake, or meal/sleep time and duration. Typically, these systems would be powered by conventional electrical wall outlets with possibly a back-up power supply (battery pack). The reader can find a good study on fall detection techniques in a recent article by Noury *et al.* [7].

Manuscript received November 23, 2009; revised July 7, 2010; accepted November 5, 2010. Date of publication March 17, 2011; date of current version May 4, 2011. This work was supported by the Natural Sciences and Engineering Research Council of Canada. This paper was recommended by Associate Editor T. Fujii.

C. Rougier and J. Meunier are with the Department of Computer Science and Operations Research, Université de Montréal, Montréal, QC H3T 1J4, Canada (e-mail: rougierc@iro.umontreal.ca; meunier@iro.umontreal.ca).

A. St-Arnaud is with the CSSS Lucille-Teasdale (Health and Social Care System), Montréal, QC H1W 0A9, Canada (e-mail: alain.starnaud.lteas@ssss.gouv.qc.ca).

J. Rousseau is with the School of Rehabilitation, Université de Montréal, Montréal, QC H3W 1W4, Canada, and also with the Research Center, Institut Universitaire de Gériatrie de Montréal, Montréal, QC, Canada (e-mail: jacqueline.rousseau@umontreal.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2129370

II. Related Works in Computer Vision

A. Monocular Systems

Among fall detection methods, one of the simplest and commonly used techniques is to analyze the bounding box representing the person [8], [9] in the image. However, this method is efficient only if the camera is placed sideways, and can fail because of occluding objects. For more realistic situations, the camera has to be placed higher in the room to avoid occluding objects and to have a larger field of view.

Lee and Mihailidis [10] detected falls by analyzing the silhouette and the 2-D velocity of the person, with special thresholds for inactivity zones like the bed. Nait-Charif and McKenna [11] tracked the person using an ellipse, and analyzed the resulting trajectory to detect inactivity outside the normal zones of inactivity like chairs or sofas. However, they both used a camera mounted on the ceiling and therefore did not have access to the vertical motion of the body, which provides useful information for fall detection.

The 2-D (image) velocity of the person has also been used to detect falls [10], [12]. However, a problem with the 2-D velocity is that it is higher when the person is near the camera, so that the thresholds to discriminate falls from a person sitting down abruptly, for instance, can be difficult to define.

Recently, we have shown promising preliminary results on how a fall detector could be based on simple shape analysis or head tracking [13], [14]. More elaborate shape analysis will be considered in this paper based on the person's silhouette.

B. Multi-Camera Systems

Other work has been done using multi-camera systems. Thome and Miguet [15] proposed to use a layered hidden Markov model to distinguish falls from walking activities. The features used for motion analysis were extracted from a metric image rectification in each view. Anderson *et al.* [16] analyzed the states of a voxel person obtained from two cameras. Fall detection was achieved with a fuzzy hierarchy. Auvinet *et al.* [17] proposed to exploit the reconstructed 3-D silhouette of an elderly person for fall detection. An alarm was triggered when most of this volume was concentrated near the floor.

An important point about multi-camera systems is that they need to be calibrated to compute a reliable 3-D information. Another problem is that the video sequences of each camera need to be synchronized, which makes the system more difficult to implement than a monocular one.

Most fall detection systems are tested in a controlled environment. An attempt was made to use more realistic data sets by Hazelhoff *et al.* [18], with two uncalibrated cameras. Based on a principal component analysis, falls were detected using the direction of the principal component and the variance ratio of the human silhouette. A head tracking module helped to reject false alarms. These authors obtained a 100% detection rate when large occlusions were absent, but their recognition results decreased drastically to 55% for occluded activities. In this paper, we will show the performance of our method with a realistic data set containing large occlusions.

C. Our System

As seen previously, several methods exist to detect falls with good detection rates, but only a few of them take into account truly realistic data sets. The main difficulty of fall detection is not to detect falls versus walking, but rather falls versus lure activities like sitting down brutally or crouching down. Therefore, the first objective of this paper is to build a system capable of distinguishing normal and lure activities from real falls. Another problem which has to be taken into account is image processing difficulties. A realistic video data set should contain occlusions, object carrying, change of clothes and different viewpoints which are well-known sources of problems in computer vision. Therefore, the second objective of this paper is to make the system robust to most image processing difficulties encountered in practice.

The main characteristic of our method is the analysis of shape deformation through a video sequence assuming that falls will increase the shape deformation with time. Our method can work with only one uncalibrated camera, but we also tested an uncalibrated multi-camera system using an ensemble classifier to improve our results.

III. Data Set

In this section, we first introduce the data set to illustrate the main difficulties of realistic video sequences. To better test our system, each action was taken from several view points. Fig. 1 shows the configuration of the cameras in the room.

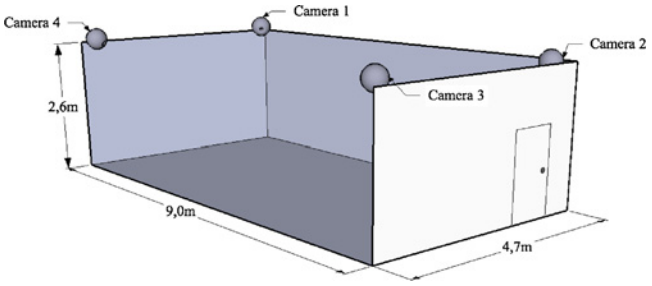

Fig. 1. Camera configuration.

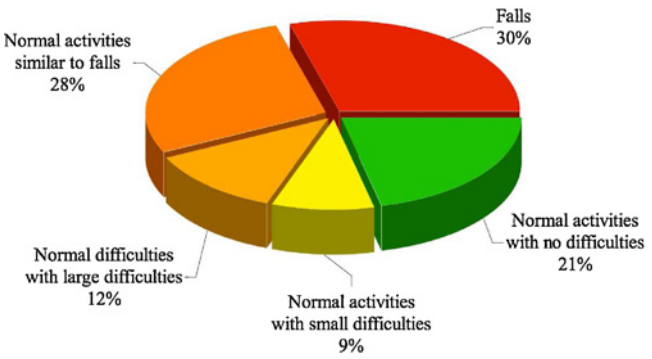

Fig. 2. Proportion of each type of event.

Our data set was composed of the following.

1) *Normal daily activities:*
   a) with no difficulties (walking in different directions);
   b) with some small difficulties (housekeeping, small occlusions);
   c) with some large difficulties (moving objects, large occlusions);
   d) with characteristics similar to falls (sitting down/standing up, crouching down).
2) *Simulated falls:*
   a) Forward falls, backward falls, falls when inappropriately sitting down, loss of balance. Falls were done in different directions with respect to the camera point of view. Note that a mattress was used to protect the person during the simulated falls.

Fig. 2 shows the proportion for each type of event in the dataset. We have a total of 75 different events for a total duration of more than 12 min for each camera. Some examples are shown in Fig. 3.

We wish our final system to be low-cost, so our video sequences were acquired using inexpensive IP cameras (Gadspot gs-4600 [19]) with a wide angle to cover all the room. The acquisition frame rate was 30 frames/s and the image size was 720×480 pixels.

Our video sequences contained typical difficulties which can lead to segmentation errors like:

1) *high video compression* (MPEG4) which can give artifacts in the image;

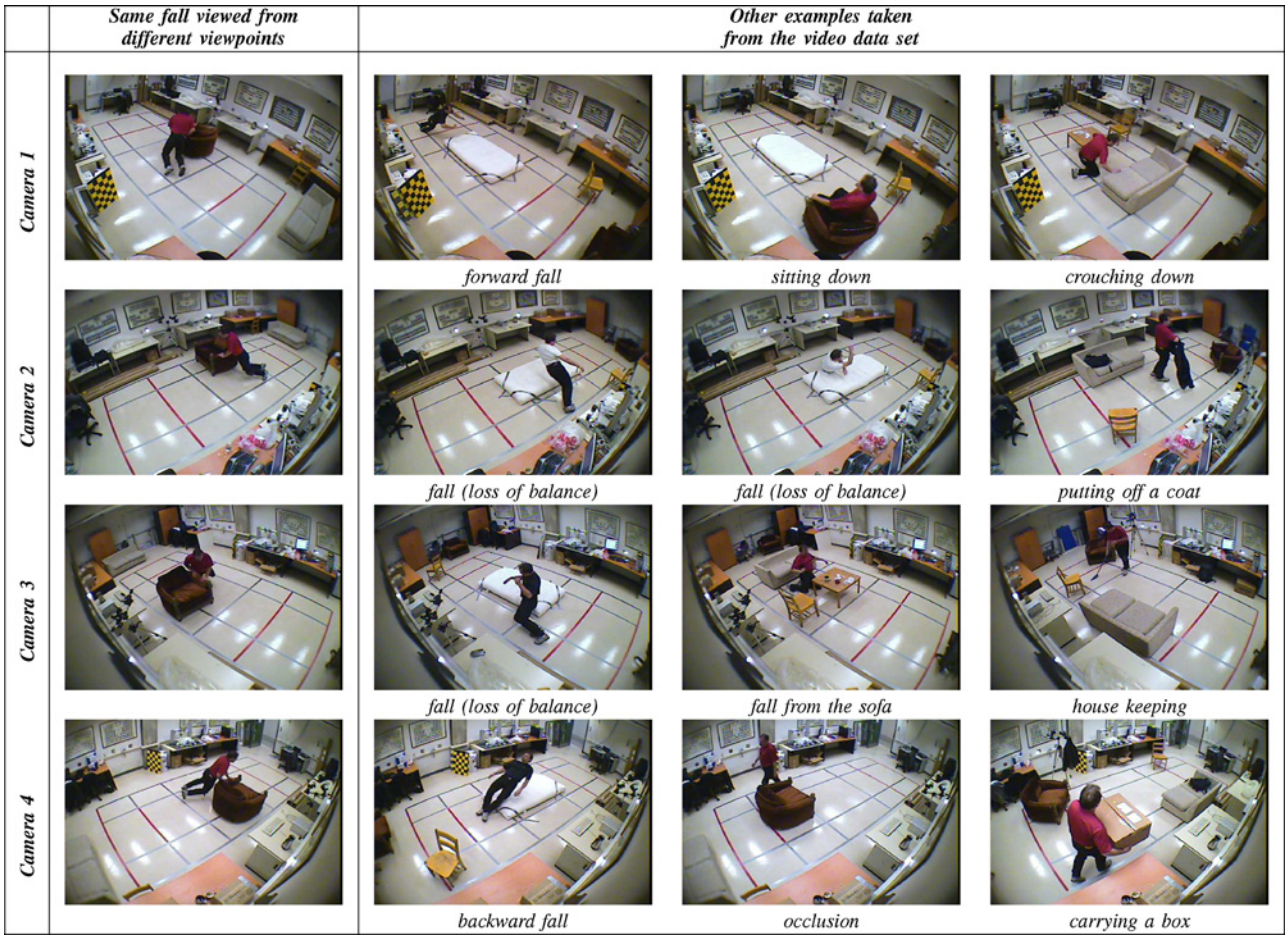

Fig. 3. First column shows a fall, slowed down by grabbing an armchair, from different viewpoints. The other columns show other examples of falls, lures, and data set difficulties.

2) *shadows* and *reflections* which can be detected as moving objects during a segmentation process;
3) *cluttered and textured background*;
4) *variable illumination* which must be taken into account during the background updating process;
5) *carried objects* (bags, clothes, and so on) which must also be taken into account during the background updating process;
6) *occlusions* (chairs, sofa, and so on);
7) *entering/leaving* the field of view;
8) *different clothes* with different color and texture, putting on and taking off a coat.

IV. Falls Characteristics

To better design our system, we must first understand how to detect a fall. According to Noury *et al.* [7], automatic methods for fall detection are based on the detection of:

1) *lack of significative movement:* usually, the person will remain immobile or will show little motion on the ground after a serious fall, at least for some time;
2) *a lying position:* according to the authors, this method is prone to many "false positives;" for example, if the person sleeps outside the bedroom at irregular hours;
3) *a person lying on the ground:* this method is better than the previous one unless the fall does not end on the ground;
4) *vertical speed:* an appropriate threshold could allow us to distinguish falls from normal activities (sitting down, crouching down, and so on);
5) *an impact shock:* easily detectable with an accelerometer or vibration detector, but more difficult with a computer vision system.

We can add another characteristic which, unlike other sensors, can be quantified using a camera.

1) *Body shape change:* indeed, the human shape will progressively and slowly change during usual activities, while during a fall, it will change drastically and rapidly.

Currently, two main technologies are used to detect falls: wearable devices [2]–[4], [7] and camera based devices [8]–[18]. Table I summarizes the sensor performance to detect fall characteristics with:

1) wearable devices: accelerometers and other sensors;
2) 2-D vision system: only one camera is required, the system does not need to be calibrated;
3) 3-D vision system: 3-D information can be recovered from one calibrated camera; but for better performance, a calibrated multi-camera system is preferable.

TABLE I
COMPARISON BETWEEN WEARABLE DEVICES AND VISION SYSTEMS

	Wearable Device	2-D Vision System	3-D Vision System
Lack of significative movement	++ ++	++ ++	++ ++
Lying position	+	+	++
Lying on the ground	–	+	++
Vertical speed	++	+	++
Impact shock	++	–	+
Body shape change	–	++	++

Lack of movement is easily detectable with all sensors. With a 3-D vision system, it is possible to localize precisely a person relatively to his/her environment. Therefore, a 3-D vision system can easily detect the characteristics of *lying position* and *lying position on the ground*. This is more difficult with a 2-D vision system since we do not have any accurate 3-D information without calibration. Wearable sensors do not give any information about the person's position relative to the ground. However, it is possible to have the body orientation using a 3-D gyroscope which could detect a *lying position*. For a more precise localization, such as *lying position on the ground*, it is necessary to couple these wearable sensors with floor sensors. The *vertical speed* depends on the camera point of view with 2-D vision system, but is easily measurable with wearable devices and 3-D vision systems. Detecting the *impact shock* with vision systems is not easy since the frame rate is usually not sufficiently high. It could be possible to detect some changes in the person's acceleration, but this change is generally not sufficiently accurate to distinguish a fall from a person sitting down. On the other hand, vision systems are outstanding for the analysis of the actions of the person and to quantify *body shape change*.

Based on these observations, we chose to combine two fall characteristics to increase the robustness of our system: human shape deformation during the fall, followed by a lack of significative movement just after the fall. These choices are justified because the body shape change includes somehow the information produced by the vertical speed and impact shock characteristics. Furthermore, the two *lying* characteristics are not sufficiently robust while the *lack of significative movement* feature is easy to compute and adds robustness to the *body shape change*.

While 3-D vision systems are better than 2-D systems, for some of the features listed in Table I, they are not significantly superior for *body shape change* and *lack of significative movement*. Furthermore, 3-D vision systems are more difficult to implement and they need to be calibrated. Thus, we develop here a 2-D vision system with only one or with a few uncalibrated cameras.

V. METHOD OVERVIEW

A fall is characterized by a large movement and some changes in the human shape. More precisely, during usual activities, the human shape will change progressively and (relatively) slowly, while during a fall, the human shape will change drastically and rapidly. Thus, we chose to detect falls during the video sequence by quantifying human shape deformation.

The main steps of our system are as follows.

1) *Silhouette edge points extraction.*
   For body shape change analysis, we need to compare two consecutive silhouettes of a person. As landmarks, we chose to extract some edge points from the silhouette of the person. The silhouette is obtained by a foreground segmentation method, and some edge points are extracted from the silhouette by a Canny edge detector. This step is described in Section VI.
2) *Matching with shape context.*
   The silhouette edge points are then matched through the video sequence. The shape matching is useful to track and to quantify the silhouette deformation. For this purpose, we use the shape context matching method [20] described in Section VII.
3) *Shape analysis.*
   For body shape change analysis, we chose to compare two deformation measures:
   a) the mean matching cost obtained from the shape context matching which has been used for shape recognition [20];
   b) the full Procrustes distance [21] which is a well-known tool for shape analysis, and which has been widely used to compare shapes in biology and medicine.

   The shape analysis step is described in Section VIII.
4) *Fall detection using GMM.*
   Finally, we use a Gaussian mixture model (GMM) to classify the different activities as a fall or not, based on shape deformation during the fall followed by a lack of significative movement after the fall. This step is described in Section IX.

VI. SILHOUETTE EDGE POINT EXTRACTION

Usually, the whole silhouette of the person is used for shape analysis [22], [23]. The silhouette is extracted by a background subtraction method which consists of comparing the current image with an updated background image. We chose the method described by Kim *et al.* [24] which takes into account shadows, highlights and high image compression. In addition to the foreground silhouette contour, we chose to extract edge points inside the silhouette using a Canny edge detector [25] to provide additional shape information. An example of edge point extraction is shown in Fig. 4.

The moving edge points will be used to match two consecutive human shapes. Since we do not need as many points, we select N landmarks regularly-spaced for each silhouette. We used $N = 250$ landmarks for our experiment. The increment to select regularly the landmarks is the same for the two shapes. If n_{i-1} and n_i are respectively the total number of edge points from the previous and the current silhouette, then the increment is set to

$$inc = \frac{\max(n_{i-1}, n_i)}{N}. \quad (1)$$

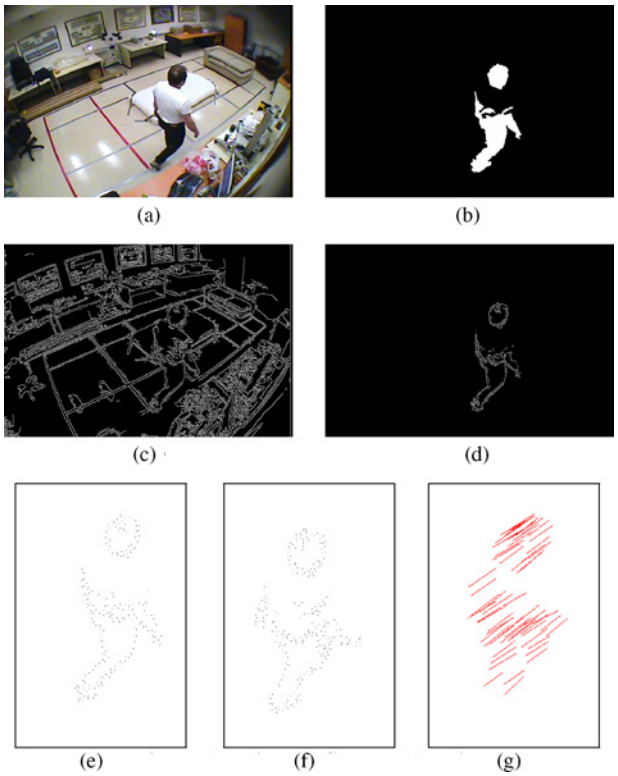

Fig. 4. From the (a) original image, the (b) foreground silhouette is extracted. Due to compression, occlusions, and segmentation problems, this silhouette is not clean enough to be used for shape analysis. By combining the (b) moving silhouette with the (c) Canny edge image, we obtain the (d) moving edge points which are used to choose a set of (e) selected edge points. By matching with the (f) previous selected edge points, we obtain the (g) matching points.

An example of moving edge points and selected edge points is also shown in Fig. 4.

VII. Matching Using Shape Context

The moving edge points extracted from two consecutive images are then matched using shape context [20]. shape context is a shape descriptor that encodes local information about each point relative to its neighbors. An algorithm for shape context matching with edge points has been proposed by Mori and Malik [26]. Unlike them, however, we discard unnecessary background edge points with the background subtraction segmentation. As shape context is sensitive to background edges, we improve it by considering only moving edge points for cluttered scenes.

The shape matching process consists of finding for each point p_i of the first shape, the best corresponding point q_j of the second shape.

For each point p_i on the shape, we compute a log-polar histogram h_i of the relative coordinates of the remaining $n-1$ points

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in bin(k)\}. \tag{2}$$

The log-polar histogram is obtained by positioning the polar coordinate system on each edge landmark p_i as shown in Fig. 5.

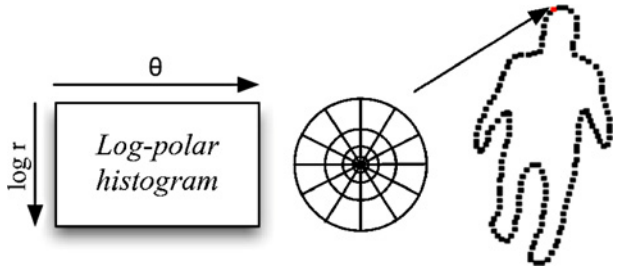

Fig. 5. Log-polar histogram computation for a point p_i. The log-polar histogram has 5 bins for $\log r$ and 12 bins for θ as proposed by the authors in [20].

Then, to find similar points on the two shapes, we compute a matching cost $C_{ij} = C\left(p_i, q_j\right)$ for each pair of points $\left(p_i, q_j\right)$. This matching cost is computed with the χ^2 statistic

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{K}\frac{\left[h_i(k) - h_j(k)\right]^2}{h_i(k) + h_j(k)} \tag{3}$$

where $h_i(k)$ and $h_j(k)$ denote the K-bin histograms respectively for p_i and q_j.

Given the set of costs C_{ij} between all pairs of points, the best corresponding points are obtained by minimizing the total matching cost $H(\pi)$ given a permutation $\pi(i)$

$$H(\pi) = \sum_i C\left(p_i, q_{\pi(i)}\right). \tag{4}$$

The authors in [20] proposed to use the Hungarian algorithm [27] for bipartite matching. The input of this algorithm is a square cost matrix with entries C_{ij}, and the result corresponds to the permutation $\pi(i)$ minimizing $H(\pi)$. As we can have some bad landmarks in our selected edge points (due to segmentation errors and/or partial occlusions), we need to add some dummy points or outliers in the matching process. However, the number of these dummy points is not easy to choose, especially in the case of severe occlusion where some bad landmarks can still remain.

In our implementation, we propose to match only the most reliable points by finding those that have their cost minimal for the row and the column of the matrix $\left(\min_i C_{ij} = \min_j C_{ij}\right)$.

To quantify the shape deformation, we need reliable landmarks, so we also clean the set of matching points based on the motion of the person. The mean motion vector $\bar{v}$ and the standard deviation σ_v are computed with the set of matching points. We keep the vectors within 1.28 standard deviations from the mean which corresponds to 80% of the motion vectors.

The *mean matching cost* $\bar{C}$ is then obtained by averaging all the best matching points costs

$$\bar{C} = \frac{1}{N^*}\sum_{n=1}^{N^*} C^*(n) \tag{5}$$

with $C^*(n)$ being the cost of the nth best matching points and N^*, the total number of best matching points.

An example of shape context matching is shown in Fig. 4. With the Hungarian algorithm, some bad matching points can appear in spite of the inclusion of dummy points. With our

method, only reliable landmarks are kept which is important in the quantification of the shape deformation. Another advantage is that the computational time is reduced with our method compared to the Hungarian algorithm.

VIII. Shape Analysis

A. Mean Matching Cost

The *mean matching cost* $\bar{C}$ of the best corresponding points is obtained during the shape matching step (see Section VII) and we propose here to use it to quantify abnormal shape deformation. In fact, the mean matching cost should be high during the fall and low just after the fall.

B. Full Procrustes Distance

Procrustes analysis [21] is a well-known tool for shape analysis, which has been widely used to compare shapes in biology or medicine. Some researchers have also used this method for gait recognition [22], [23], [28]. We propose to use it here to detect abnormal shape deformation for fall detection.

The main characteristic of Procrustes shape analysis [21] is that the shapes are compared once translational, rotational and scaling components are removed to normalize the shapes. Concretely, two sets of landmarks are obtained from the shapes to be compared. Then, the full Procrustes distance (defined below) can be computed to quantify the deformation between the two shapes. This distance will be high in case of a fall.

Each human shape is represented by k landmarks and can be described by a k dimensional complex vector Z

$$\begin{aligned} Z &= [z_1, z_2, \ldots, z_i, \ldots, z_k] \\ z_i &= x_i + jy_i \quad \text{where } j = \sqrt{-1}. \end{aligned} \tag{6}$$

The centered landmarks Z_C are obtained by multiplying the coordinates Z with the centering matrix C

$$Z_C = CZ \qquad \text{with} \quad C = I_k - \frac{1}{k} 1_k 1_k^T \tag{7}$$

where I_k is a $k \times k$ identity matrix and 1_k is a k dimensional vector of ones.

Consider now two centered configurations $v = (v_1, \cdots, v_k)$ and $w = (w_1, \cdots, w_k)$. A suitable distance between them can be obtained by choosing the best registration of v and w with a similarity transformation and then by computing the remaining distance between these complex vectors. For two centered complex configurations, this full Procrustes distance is simply [21]

$$D_f(v, w) = \left\{ 1 - \frac{|v^* w|^2}{\|v\|^2 \|w\|^2} \right\}^{1/2}. \tag{8}$$

The *full Procrustes distance* D_f is computed between the matching points of two consecutive images. D_f should increase in case of a fall, and should be low after the fall.

IX. Fall Detection Using GMM

The fall detection problem can be seen as an outlier detection problem. Indeed, the fall recognition system needs to be a one-class classifier, which is trained with normal activities with the aim of detecting abnormal events like falls. Detecting abnormality has been done by Nanri and Otsu [29] with an unsupervised method. Some cubic higher-order local auto-correlation features are used to create a subspace of normal movements. Then, any abnormal movement can be detected even in a scene with multiple persons. Xiang and Gong [30] demonstrated that an unlabeled dataset gives superior results than a labeled data set for abnormality detection from unseen video sequences. A survey of novelty detection methods can be found in the article [31]. For our experiment, we model our normal activity data with a GMM.

A. Gaussian Mixture Model (GMM)

A Gaussian mixture model [32] can be defined by a weighted sum of Gaussian distributions

$$p(x) = \sum_{j=1}^{M} P(j)\, p(x \mid j) \tag{9}$$

where M is the number of components in the mixture and $P(j)$ are the mixing coefficients. The jth Gaussian probability density function $p(x \mid j)$ has the form

$$p(x \mid j) = \frac{1}{\left(2\pi \prod_{i=1}^{d} \sigma_{j,i}^2\right)^{d/2}} \exp\left\{ -\sum_{i=1}^{d} \frac{\left(x_i - \mu_{j,i}\right)^2}{2\sigma_{j,i}^2} \right\} \tag{10}$$

where d is the dimensionality of the input space.

The parameters to be estimated are the mixing coefficients $P(j)$, the mean vector μ of dimension d and the diagonal covariance matrix $\sum_j = diag\left(\sigma_{j,1}^2, \ldots, \sigma_{j,d}^2\right)$. The parameters are determined using the expectation-maximization (EM) algorithm by maximizing the data likelihood. Specifically in our case, the parameters of the GMM are estimated from a training data set of normal daily activities (walking, sitting down, crouching down, housekeeping, and so on). Then, a new activity is tested with the GMM obtained by training, and a decision threshold (log-likelihood threshold) allows to classify this activity as normal or abnormal.

B. Leave-One-Out Cross-Validation

A leave-one-out cross-validation is used to train and test our dataset. The dataset is divided into N video sequences which contain some falls and/or normal activities (including lures). For testing, one sequence is removed from the dataset, and the training is done using the $N - 1$ remaining sequences (where falls are deleted because the training is only done with normal activities). This sequence is then classified with the resulting GMM. This is repeated N times by removing each sequence in turn. By counting the number of errors, classification error rate and other measurements can be computed (see Section IX-D).

C. GMM Features

A fall is characterized by a peak on the smoothed *full Procrustes distance* curve or *mean matching cost* curve followed by a lack of significative movement of the person just after the fall, as shown in Fig. 6.

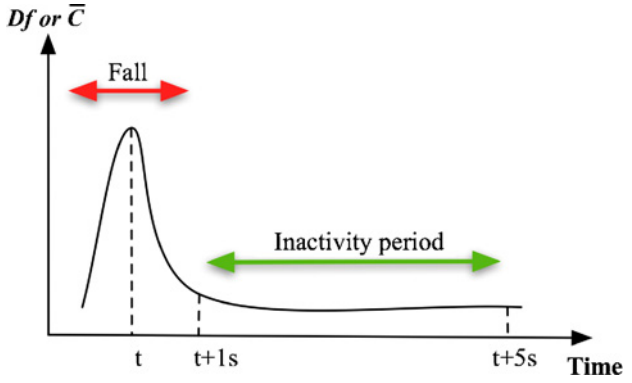

Fig. 6. Fall features based on the full Procrustes distance D_f or mean matching cost $\bar{C}$.

We therefore consider two features (F_1, F_2) for the GMM classification representing these characteristics.

1) F_1 *representing the fall.*
   This corresponds to $D_f^{(t)}$ or $\bar{C}^{(t)}$ at time t. F_1 will be high in case of a fall.
2) F_2 *representing the lack of significative movement after the fall.*
   This corresponds to the mean value of D_f or $\bar{C}$ between one and five seconds after the fall (i.e., $t+1$ s to $t+5$ s). This predefined period of time is not a sensitive parameter and could be longer if needed. We chose +1 s to avoid the peak and +5 s because after that period the actor in our videos stood up after a fall (we did not ask him to stay on the floor indefinitely).

D. GMM Analysis

To analyze our recognition results, we compute the sensitivity, the specificity, the accuracy and the error rate obtained with our GMM classifier as follows:

1) true positives (TP): number of falls correctly detected;
2) false negatives (FN): number of falls not detected;
3) false positives (FP): number of normal activities (including lures) detected as a fall;
4) true negatives (TN): number of normal activities (including lures) not detected as a fall;
5) sensitivity: $Se = TP/(TP + FN)$;
6) specificity: $Sp = TN/(TN + FP)$;
7) accuracy: $Ac = \frac{(TP+TN)}{(TP+TN+FP+FN)}$;
8) classification error rate: $Er = \frac{(FN+FP)}{(TP+TN+FP+FN)}$.

A good fall detection system must have a high sensitivity, which means that a majority of falls are detected, and a high specificity, which means that normal activities and lures are not detected as falls. Similarly, the accuracy must be high while the error rate must be low.

X. Experimental Results

Our method works with a single uncalibrated camera. The shape matching is implemented in C++ using the OpenCV library [33] and the fall detection step is done with MATLAB using the NETLAB toolbox [32] to perform the GMM classification.

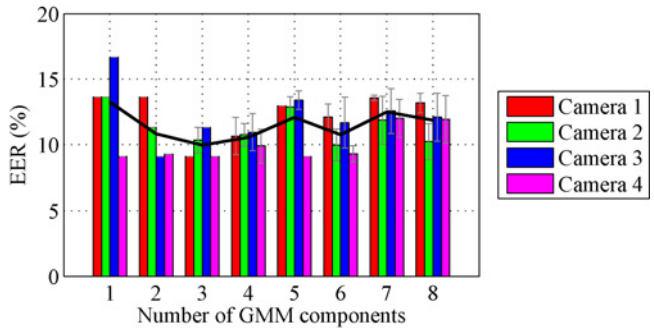

Fig. 7. EER as a function of number of GMM components for each camera for D_f features. The black curve represents the mean.

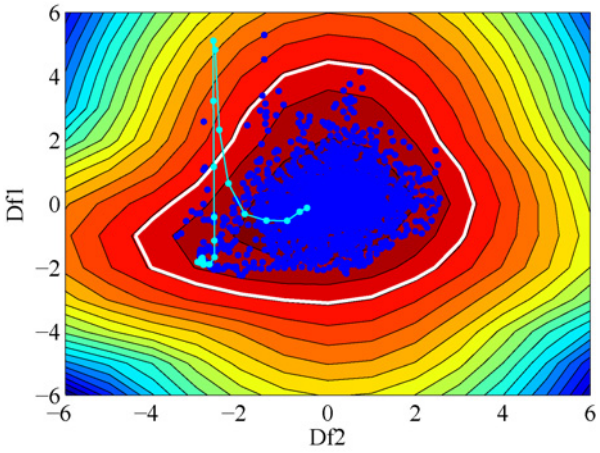

Fig. 8. Example of log-likelihood obtained with a 3-component GMM. The dark blue points represent the normalized training data. The light blue points correspond to a sequence where a fall occurs. The white line represents the boundary obtained for the log-likelihood threshold. The light blue points which are outside the boundary represent the detected fall.

The original video sequences were acquired with a frame rate of 30 frames/s, but 5 frames/s was sufficient to detect a fall. The computational time of the shape matching step is about 200 ms on an Intel Core 2 Duo processor (2.4 GHz), which is adequate for our application with a frame rate of 5 frames/s.

A. Number of GMM Components

We explored the relationship between the number of components of the GMM and the classification results.

The initialization of the EM algorithm [32] can influence the resulting GMM, so we repeated the cross-validation ten times for each number of components and each camera. A receiver operating characteristic (ROC) analysis is performed by varying the threshold, and the equal error rate (EER) is computed.

Fig. 7 shows the EER as a function of the number of GMM components. This demonstrates that one or two components for the GMM are not sufficient because they give poorer recognition results. If we take too many GMM components, the EER and its standard deviation increase, as well as the computation time. Thus, we chose to train a GMM with three components for our experiment, which is a good compromise between a low classification error rate, a good repeatability of the results and a reasonable computation time.

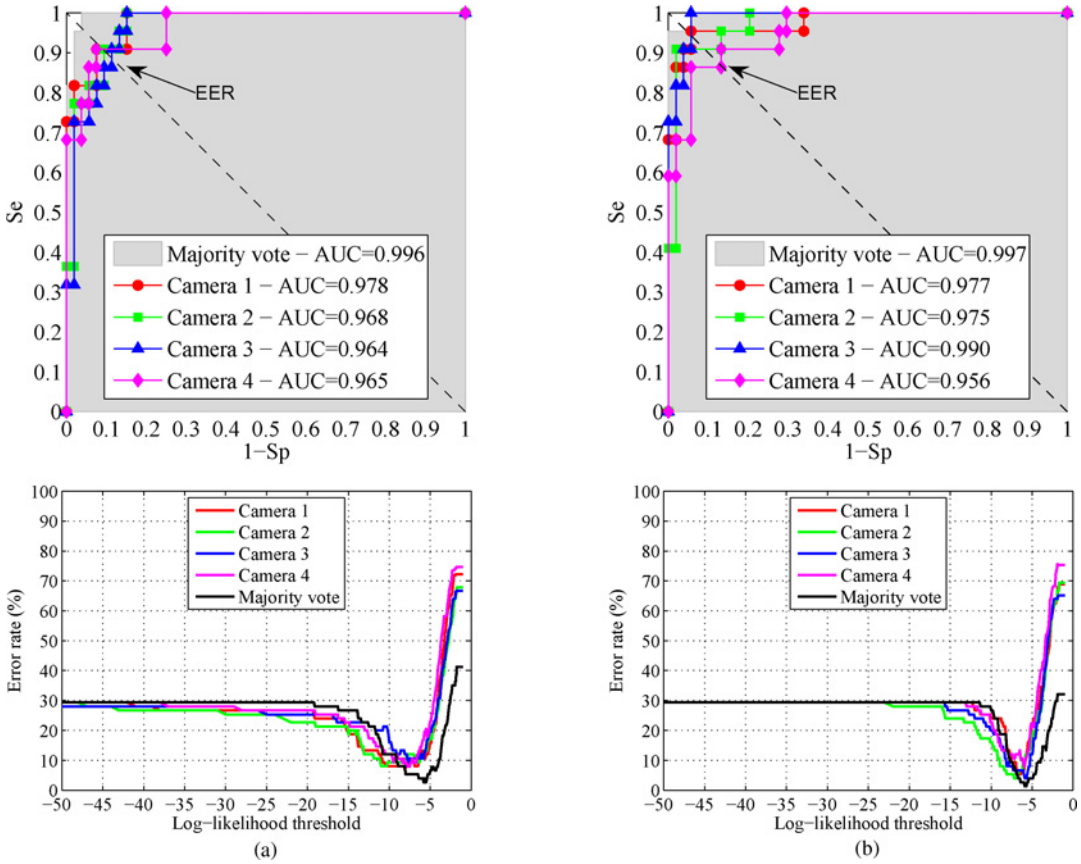

Fig. 9. On the top, the ROC curves obtained for the (a) full Procrustes distance and for the (b) mean matching cost. On the bottom, the classification error rate curves obtained by varying the GMM log-likelihood threshold. The results were computed for each camera independently and for a majority vote (at least three of four cameras). (a) Results using the full Procrustes distance features $\left(D_{f_1}, D_{f_2}\right)$. (b) Results using the mean matching cost features $\left(\bar{C}_1, \bar{C}_2\right)$.

B. Classification Results

Fig. 8 shows an example of the log-likelihood obtained with a 3-component GMM with full Procrustes distance features. The input features are normalized to unit standard deviations and zero means. An example of a trajectory generated from a video sequence where a fall occurs is also superimposed on the graphic.

The choice of a detection threshold depends on the sensitivity required for the system. We perform a ROC analysis to study the influence of the GMM log-likelihood threshold. Fig. 9 shows the ROC curves and the classification error rates obtained for each camera independently for the full Procrustes distance and mean matching cost features. They are obtained with a 3-component GMM and a log-likelihood threshold ranging from -50 to -1.

C. Ensemble Classifier

The ROC curves show that our recognition results are good for each camera independently. But we also tried to improve our results by combining the results of all cameras. This was done with an ensemble classifier as shown in Fig. 10.

The rule used is simply a majority vote on all cameras. Each camera GMM classifier has one vote, and if an abnormal event occurs in a majority of cameras, the event is considered as abnormal. Therefore, if at least three of the four cameras detect a fall, the event is considered as a fall. By combining all camera results, our system accuracy increased as shown in Fig. 9.

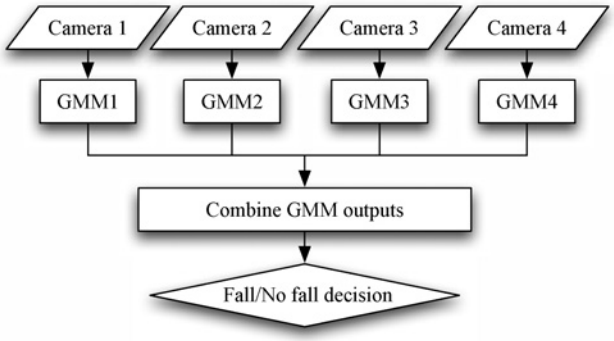

Fig. 10. Overview of the ensemble classifier.

By choosing the best threshold, two events still remain misclassified with the D_f majority vote. One lure is detected as a fall, because of a high D_f peak, when the person brutally sits down on the armchair. One fall is not detected when the person gets up from the sofa and falls on the small table. The reason is that this fall is rather smooth and the resulting D_f peak is not sufficiently high.

As expected, the *mean matching cost* gave good results similar to the *full Procrustes distance*. Fig. 9 showed that ROC curves with these features are similar, which provides evidence

Fig. 11. Example of the (red) bounding box and of the (green) approximated ellipse of the silhouette. We can see here that the bounding box is very sensitive to carried objects.

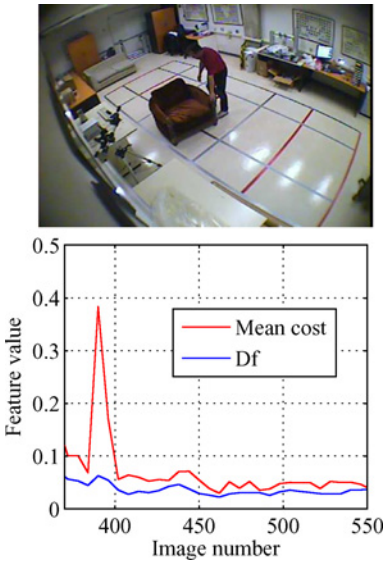

Fig. 12. Example of full Procrustes distance D_f and mean cost $\bar{C}$ curves for a video sequence where a person moved an armchair. The armchair is seen as a moving object which changes drastically the context of the object producing a false fall detection (peak) for $\bar{C}$.

that our method is view-independent for the two features. For the *full Procrustes distance*, the best classification error rate is less than 10% for each camera independently, and decreases to 2.7% using a majority vote. For the *mean matching cost*, the majority vote gave more than 98% accuracy.

D. Comparative Study with Other 2-D Features

In this section, we compare our shape feature with other commonly used 2-D features.

1) The *aspect ratio ρ of the bounding box*. Several works used this feature to detect falls [8], [9] because of its simplicity. The bounding box should change from a vertical to a horizontal orientation after the fall.
2) The *2-D vertical velocity* v_y. This feature has been used for fall detection in [12] with an infrared sensor. The 2-D vertical velocity is computed from the motion of the centroid of the person's silhouette. It should be high during the fall and low just after the fall.
3) The *normalized 2-D vertical velocity* v_{yn}. An object moving at a constant 3-D velocity will display a higher 2-D image velocity near the camera than far away from it due to perspective projection. One possible solution to this inconsistency problem could be to normalize the 2-D velocity by the person's 2-D size. To estimate this size, we compute the best fitting ellipse of the silhouette using moments [14] as shown in Fig. 11. The normalized 2-D vertical velocity is then obtained by dividing it by the size of the ellipse major axis. It should be high during the fall and low just after the fall.

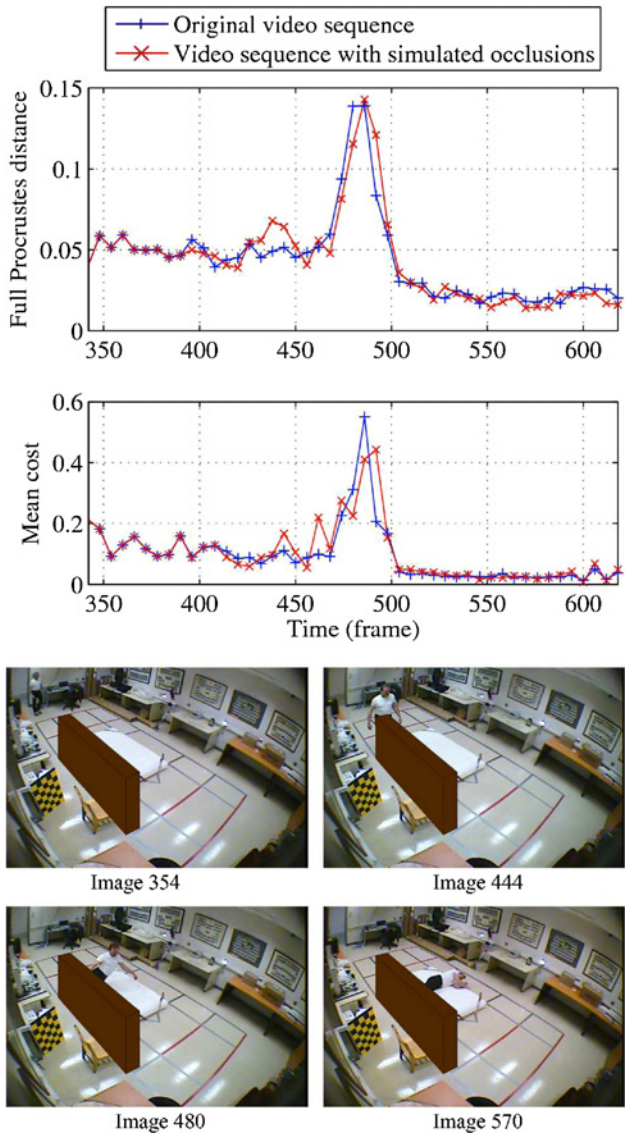

Fig. 13. Figure shows the influence of occluding objects. The blue curves correspond to the full Procrutes distance and the mean matching cost for an original video sequence of a fall. The red curves show the results obtained when an occluding object is simulated in the video sequence.

For a fair comparison, our 3-component GMM classifier is used with the new features representing the fall and the same inactivity period after the fall. Table II(a) summarizes the classification results obtained with the cross-validation for each feature. The computed values are the EER and the area under the curve (AUC) for each camera, and the EER and AUC corresponding to the majority vote for four cameras (events detected for at least three out of four cameras).

The *ratio of the bounding box* gave poor results. Indeed, segmentation errors can affect the bounding box, because

TABLE II
EER AND AUC VALUES FOR (A) DIFFERENT FEATURES, (B) USING HUNGARIAN SHAPE MATCHING AND (C) USING INACTIVITY ZONES

Features*	Shape Matching†	Inactivity Zones	Cam. 1 EER (AUC)	Cam. 2 EER (AUC)	Cam. 3 EER (AUC)	Cam. 4 EER (AUC)	Majority Vote EER (AUC)
$\left(D_{f_1}, D_{f_2}\right)$	Best	No	9.1% (0.978)	9.4% (0.968)	11.3% (0.964)	9.1% (0.966)	3.8% (0.996)
$\left(\bar{C}_1, \bar{C}_2\right)$	Best	No	5.7% (0.977)	9.1% (0.975)	5.7% (0.990)	13.2% (0.956)	4.6% (0.997)
(ρ_1, ρ_2)	Best	No	45.5% (0.631)	29.6% (0.729)	40.9% (0.578)	31.8% (0.719)	43.4% (0.488)‡
$\left(v_{y_1}, v_{y_2}\right)$	Best	No	36.4% (0.697)	22.7% (0.817)	22.7% (0.745)	40.7% (0.653)	11.3% (0.889)
$\left(v_{yn_1}, v_{yn_2}\right)$	Best	No	44.4% (0.622)	36.4% (0.701)	32.1% (0.675)	50.0% (0.538)	22.7% (0.776)‡
D_{f_1}	Best	No	27.3% (0.813)	11.3% (0.937)	18.2% (0.886)	24.1% (0.849)	13.6% (0.879)‡
$\bar{C}_1$	Best	No	27.3% (0.778)	17.0% (0.907)	22.7% (0.846)	36.4% (0.676)	17.0% (0.881)‡

(a) Features Comparison.

$\left(D_{f_1}, D_{f_2}\right)$	Hung	No	13.2% (0.963)	9.4% (0.965)	7.6% (0.988)	13.6% (0.930)	9.1% (0.907)
$\left(\bar{C}_1, \bar{C}_2\right)$	Hung	No	11.3% (0.953)	9.1% (0.979)	9.4% (0.979)	13.6% (0.935)	0% (1)

(b) Comparison with Hungarian Matching.

$\left(D_{f_1}, D_{f_2}\right)$	Best	Yes	5.7% (0.983)	9.1% (0.979)	7.6% (0.971)	9.1% (0.983)	0% (1)
$\left(\bar{C}_1, \bar{C}_2\right)$	Best	Yes	4.6% (0.984)	0% (1)	5.7% (0.988)	9.4% (0.972)	1.9% (0.999)

(c) Influence of Inactivity Zones.

*Full Procrustes distance D_f, mean matching cost $\bar{C}$, bounding box ratio ρ, 2-D vertical velocity v_y, normalized 2-D vertical velocity v_{yn}.
†Best = Best matching points, Hung = Hungarian matching.
‡Significantly different w.r.t. $\left(D_{f_1}, D_{f_2}\right)$ using a one-sided binomial test ($p < 0.05$).
ß Almost significantly different $p = 0.0898$.

of shadows, highlights, occlusions, object carrying or simply if the person extends his arms as shown in Fig. 11. The *2-D vertical velocity* is sensitive to the camera view point, the velocity is higher when the person is near the camera. Normalizing the 2-D vertical velocity does not improve the recognition results mainly because the size of the person is unreliable as explained for the bounding box aspect ratio. However, with a better assessment of the person's size, this approach could be more effective.

The *mean matching cost* and the *full Procrustes distance* gave the best recognition results with, respectively, an equal error rate of 4.6% and 3.8% with a majority vote.

Table II(a) shows also that the inactivity period is important to confirm the fall. If we only consider the first feature D_{f_1} or $\bar{C}_1$, the results deteriorate considerably as expected.

The results obtained using the Hungarian algorithm [27] for bipartite matching with 20% dummy points are also shown in Table II(b) for comparison with our method using only the best matching points. The results are not statistically different from those obtained with our methodology. However, Hungarian matching is more time consuming, requires choosing the percentage of dummy points (a parameter that affects considerably the quality of the results) and can leave bad matching points.

A solution to improve recognition results could be to define normal inactivity zones (for example, the bed or the sofa) as in the work of Lee and Mihailidis [10], where the detection thresholds were less sensitive. We defined manually normal inactivity zones in our sequences, and when the centroid of the person was inside one of these zones, the detection threshold was fixed at 1.5 times the normal threshold. The results using inactivity zones are better, as shown in Table II(c). An automatic method, which could be remotely activated, can be used to learn inactivity zones [11] when the system is installed or later if necessary (e.g., new furniture or object displacement).

To summarize, D_f or $\bar{C}$ features gave the best results compared to other features. A multi-camera system with four uncalibratred cameras increased the performance and so did the inclusion of known inactivity zones.

E. Occlusions and Other Difficulties

The *mean matching cost* $\bar{C}$ can be sensitive in case of moving object. Fig. 12 shows an example where $\bar{C}$ is incorrect when the person moved an armchair. Since $\bar{C}$ is based on shape context, and the context is not the same when an object moves, the matching cost increases similarly to a fall. The full Procrustes distance D_f is more robust in this case since it measures the shape deformation of *reliable* matched points.

We simulated a large object occlusion in a video sequence to analyze the effect of occlusions on the Full Procrutes distance and the mean matching cost curves. The curves, shown in Fig. 13, were slightly disturbed with this major occlusion but did not generate high peaks similar to falls. Notice that in the case of a severe occlusion (not enough edge points for silhouette matching), the algorithm stops and restarts when the silhouette reappears.

The combination of two characteristics explains the robustness of our method. Even if some error peaks appear, the *lack of significative movement* (which is not sensitive to occlusions) helps to discriminate real falls. This can be clearly observed in Table II(a), when we only consider the first feature D_{f_1} or $\bar{C}_1$, the results deteriorate considerably.

XI. Discussion and Conclusion

In this paper, we presented a new GMM classification method to detect falls by analyzing human shape deformation.

The edge point matching step (with shape context) is robust to occlusions and other segmentation difficulties. We also observed that the addition of edge points within the silhouette (Canny filter), while not absolutely necessary, generally helped to improve the results (e.g., reduction of the EER from 9.1% to 3.8% for the majority vote with D_f).

Human shape deformation is a useful tool to detect falls. We demonstrated that the full Procrustes distance and the mean matching cost are really discriminant features for classification. Because they can be sensitive to bad matching points, only reliable matching points were kept for shape deformation assessment. The peak representing the fall is an important feature to characterize a fall, but the *lack of significative movement* after the fall is also important for robustness when occlusions occur. A little motion after the fall will not significantly influence the human shape deformation.

This paper was done with a realistic data set, and in spite of the low-quality images (high compression artifacts, noise) and segmentation difficulties (occlusions, shadows, moving objects, different clothes, and so on), the recognition results are excellent. The system can run in real time at 5 frames/s which is fast and sufficient to detect a fall. Finally, compared with other 2-D features, the shape deformation features are significantly superior tools to detect falls.

When developing such systems, we must ensure the privacy of the person. This requirement is satisfied with our system as it is entirely automated, and nobody has access to the images except in case of emergency. The system will be activated to send an alarm signal toward an outside resource (e.g., via a cell phone or Internet) if and only if an abnormal event is detected (e.g., falling). Moreover, this is a technique that does not require the person to wear any device.

We hope that this research study will set the ground for the development of healthcare video surveillance systems to improve the quality of life and care for the elderly so that they can preserve their autonomy and enjoy a greater degree of comfort in their daily lives. This corresponds to the hopes of the elderly themselves, their families, caregivers and governments. Indeed, a recent study on the acceptability by older people of such vision systems has been conducted by our team [34], and has revealed an encouraging high rate of acceptance among elderly people and care givers. When the intelligent videomonitoring system is well explained, 83.3% of caregivers and 86.7% of seniors are in favor of such system. Various studies [35] have also shown the economic advantages of support in the home setting instead of placing the elderly in a specialized long-term care establishment.

Finally, we believe that such a video system will complement advantageously other types of sensors for healthcare surveillance by overcoming many of their limitations.

Acknowledgment

The authors gratefully acknowledge the helpful assistance of E. Auvinet for the video data set.

References

[1] *Canada's Aging Population*, Public Health Agency of Canada, Division of Aging and Seniors, 2002 [Online]. Available: http://dsp-psd.pwgsc.gc.ca/Collection/H39-608-2002E.pdf

[2] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait Posture*, vol. 28, no. 2, pp. 285–291, 2008.

[3] M. Nyan, F. E. Tay, and E. Murugasu, "A wearable system for pre-impact fall detection," *J. Biomech.*, vol. 41, no. 16, pp. 3475–3481, 2008.

[4] iLife. *Fall Detection Sensor* [Online]. Available: http://www.falldetection.com/iLifeFDS.asp

[5] Directalert. *Wireless Emergency Response System* [Online]. Available: http://www.directalert.ca/emergency/help-button.php

[6] M. Alwan, P. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, "A smart and passive floor-vibration based fall detector for elderly," in *Proc. 2nd Inform. Commun. Technol.*, vol. 1. 2006, pp. 1003–1007.

[7] N. Noury, A. Fleury, P. Rumeau, A. Bourke, G. Laighin, V. Rialle, and J. Lundy, "Fall detection: Principles and methods," in *Proc. 29th Annu. Int. Conf. IEEE EMBS*, Aug. 2007, pp. 1663–1666.

[8] B. Töreyin, Y. Dedeoglu, and A. Çetin, "HMM based falling person detection using both audio and video," in *Proc. IEEE Int. Workshop Hum.-Comput. Interaction*, 2005, pp. 1–4.

[9] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He, "Recognizing falls from silhouettes," in *Proc. Int. Conf. IEEE EMBS*, Aug. 2006, pp. 6388–6391.

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

[11] H. Nait-Charif and S. McKenna, "Activity summarization and fall detection in a supportive home environment," in *Proc. 17th ICPR*, vol. 4. 2004, pp. 323–326.

[12] A. Sixsmith and N. Johnson, "A smart sensor to detect the falls of the elderly," *IEEE Pervasive Comput.*, vol. 3, no. 2, pp. 42–47, Apr.–Jun. 2004.

[13] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Monocular 3-D head tracking to detect falls of elderly people," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2006, pp. 6384–6387.

[14] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," in *Proc. 21st Int. Conf. AINAW*, vol. 2. 2007, pp. 875–880.

[15] N. Thome, S. Miguet, and S. Ambellouis, "A real-time, multiview fall detection system: A LHMM-based approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1522–1532, Nov. 2008.

[16] D. Anderson, R. H. Luke, J. M. Keller, M. Skubic, M. Rantz, and M. Aud, "Linguistic summarization of video for fall detection using voxel person and fuzzy logic," *Comput. Vision Image Understanding*, vol. 113, no. 1, pp. 80–89, Jan. 2009.

[17] E. Auvinet, L. Reveret, A. St-Arnaud, J. Rousseau, and J. Meunier, "Fall detection using multiple cameras," in *Proc. 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2008, pp. 2554–2557.

[18] L. Hazelhoff, J. Han, and P. H. N. de With, "Video-based fall detection in the home using principal component analysis," in *Proc. Adv. Concepts Intell. Vision Syst.*, vol. 1. 2008, pp. 298–309.

[19] Gadspot, Inc. *Ip Camera Gadspot* [Online]. Available: http://gadspot.com

[20] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[21] I. Dryden and K. Mardia, *Statistical Shape Analysis*. Chichester, U.K.: Wiley, 1998.

[22] L. Wang, T. Tan, W. Hu, and H. Ning, "Automatic gait recognition based on statistical shape analysis," *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1120–1131, Sep. 2003.

[23] N. Jin and F. Mokhtarian, "Human motion recognition based on statistical shape analysis," in *Proc. IEEE Conf. Adv. Video Signal Based Surveillance*, Sep. 2005, pp. 4–9.

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

[25] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

[26] G. Mori and J. Malik, "Estimating human body configurations using shape context matching," in *Proc. Eur. Conf. Comput. Vision*, vol. 2352. 2002, pp. 150–180.

[27] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistic Quar.*, vol. 2, nos. 1–2, pp. 83–97, 1955.
[28] J. Boyd, "Video phase-locked loops in gait recognition," in *Proc. IEEE ICCV*, vol. 1. Jul. 2001, pp. 696–703.
[29] T. Nanri and N. Otsu, "Unsupervised abnormality detection in video surveillance," in *Proc. IAPR Conf. Mach. Vision Applicat.*, May 2005, pp. 574–577.
[30] T. Xiang and S. Gong, "Video behavior profiling and abnormality detection without manual labeling," in *Proc. 10th IEEE ICCV*, vol. 2. Oct. 2005, pp. 1238–1245.
[31] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004.
[32] I. T. Nabney, *NETLAB—Algorithms for Pattern Recognition*. Berlin, Germany: Springer, 2001.
[33] Opencv. *Intel Open Source Computer Vision Library* [Online]. Available: http://www.intel.com/technology/computing/opencv
[34] J. Rousseau, A. St-Arnaud, F. Ducharme, J. St-Arnaud, J. Meunier, S. Turgeon-Londei, and M. Jobidon, "Videomonitoring at home: Do you want it?" in *Proc. Can. Assoc. Occupational Therapy Annu. Conf: Exploring Frontiers Occupation*, Jun. 2008.
[35] N. L. Chappell, B. H. Dlitt, M. J. Hollander, J. A. Miller, and C. McWilliam, "Comparative costs of home care and residential care," *Gerontologist*, vol. 44, no. 3, pp. 389–400, Jun. 2004.

Caroline Rougier received the M.S. degree in image processing from the University of Paris 6, Paris, France, in 2003, and the Ph.D. degree in computer science from the Université de Montréal, Montréal, QC, Canada, in 2010.

She is currently a Post-Doctoral Fellow with the Department of Computer Science and Operations Research, Université de Montréal. Her current research interests include computer vision, image and video analysis, human activity recognition, and gait analysis.

Jean Meunier received the B.S. degree in physics from the Université de Montréal, Montréal, QC, Canada, in 1981, the M.Sc.A. degree in applied mathematics in 1983, and the Ph.D. degree in biomedical engineering from the Ecole Polytechnique de Montréal, Montréal, in 1989.

In 1989, after post-doctoral studies with the Montreal Heart Institute, Montréal, he joined the Department of Computer Science and Operations Research, Université de Montréal, where he is currently a Full Professor and Chair. He is a regular member of the Biomedical Engineering Institute at the same institution. His current research interests include computer vision and its applications to medical imaging and health care.

Alain St-Arnaud received the Masters degree in psychology from the Université de Trois-Rivières, Trois-Rivières, QC, Canada, in 1987.

He pursued his training in neuropsychology with the Université de Montréal, Montréal, QC, Canada. As a Clinician, he practices in the Psychogeriatric Team, CSSS Lucille-Teasdale (Health and Social Care System), Montréal. He is also the Research Coordinator of the Psychogeriatric Team. He is a member of the Quebec Rehabilitation Research Network. His clinical and current research interests include the elderly people affected by cognitive and mental health disorders living in the community.

Jacqueline Rousseau received the M.S. degree in 1992 and the Ph.D. degree in biomedical sciences (rehabilitation) from the Université de Montréal, Montréal, QC, Canada, in 1997.

She has been an Occupational Therapist since 1981. She practiced as a Clinician until 1989. She is currently an Associate Professor with the School of Rehabilitation, Université de Montréal and is a Researcher with the Research Center, Institut Universitaire de Gériatrie de Montréal, Montréal. She is a regular member of the Quebec Rehabilitation Research Network and the Quebec Network for Research on Aging. Her current research interests include home adaptation and community integration for people living with permanent disabilities (motor, cognitive, visual, and mental health) focusing on the development of assessment tools and technology to facilitate their social participation.