Procrustes Shape Analysis for Fall Detection

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

Procrustes Shape Analysis for Fall Detection

Caroline Rougier and Jean Meunier
Département d'Informatique
et de Recherche Opérationnelle
Université de Montréal
Montréal, Canada
{rougierc, meunier}@iro.umontreal.ca

Alain St-Arnaud
Centre de santé
et de services sociaux
Lucille-Teasdale
Montréal, Canada
astarnaud@ssss.gouv.qc.ca

Jacqueline Rousseau
Centre de Recherche de
l'Institut Universitaire
de Gériatrie de Montréal
Montréal, Canada
jacqueline.rousseau@umontreal.ca

Abstract

In Western countries, the growing population of seniors brings us to think about new healthcare systems to ensure the safety of elderly people at home. Falls are one of the greatest risk for seniors living alone. Computer vision provides a promising solution to analyze people behavior and detect some unusual events like falls. In this paper, we propose to detect falls by analyzing the human shape deformation during the video sequence. The elderly silhouette is tracked along the video sequence using shape context matching. These silhouettes are then used to quantify the deformation based on Procrustes shape analysis. Our algorithm gives very promising results on video sequences of daily activities and simulated falls.

1. Introduction

1.1. Context

As others Western countries, Canada faces the growing population of seniors. According to the Public Health Agency of Canada [5], one Canadian out of eight was older than 65 years old in 2001. In 2026, this proportion will be one out of five. Moreover, a majority of seniors, 93%, reside in private house, and among them, 29% live alone [5]. Falls are one of the major risk for old people living alone, causing severe injuries. The gravity of the situation can increase if the person can not call for help, being unconscious or immobilized. Nowadays, the usual solution to detect falls is to use some wearable sensors like accelerometers or help buttons. However, 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.

1.2. Vision System

Computer vision system provides a better solution because such a system doesn't require that the person wears anything. Moreover, a camera gives more information on the motion of a person and his/her actions than an accelerometer. Thus, we can imagine a computer vision system providing information on falls, but also, checking some daily behaviors like medication intake, or meal/sleep time and duration. Typically these systems would be powered by regular house electrical wall outlet with a back-up power supply (battery pack).

When we talk about vision system, it is important to notice that the system is entirely automated to ensure the privacy (nobody has access to the images unless authorized or there is an emergency) of the person. Another study, on the acceptability by older people of such vision systems, is currently under way by our team, and has revealed a surprising high rate of acceptance among elderly people, their family and care givers when such system is well explained.

2 Related Work

Most of fall detection techniques are based on accelerometers, but since a few years, we can find some computer vision systems to detect falls. The reader can find a good study on fall detection techniques in a recent article by Noury et *al.* [13].

2.1. Fall Detection using Computer Vision Systems

The use of computer vision to detect falls is recent. A basic method consists of analyzing the bounding box representing the person in a single image [15]. This can only be efficient if the camera is placed sideways, and can fail because of occluding objects. For more realistic situation,

the camera will be placed higher in the room to not suffer of occluding objects and to have a larger field of view. In this case, depending on the relative position of the person and on the field of view of the camera, a bounding box could not be sufficient to discriminate a fall from a person sitting down.

To overcome this problem, some researchers [10] [12] have mounted the camera on the ceiling. Lee and Mihailidis [10] detect a fall by analyzing the shape and the 2D velocity of the person, and define inactivity zones like the bed. Nait-Charif and McKenna [12] track the person using an ellipse, and analyze the resulting trajectory to detect inactivity outside the normal zones of inactivity like chairs or sofas.

Infrared sensor can also be used to detect falls [14]. In their work, Sixsmith and Johnson classify falls with a neural network using the 2D vertical velocity of the person. But, as they use a wall-mounted sensor, the 2D vertical velocity could not be sufficient to discriminate a real fall from a person sitting down abruptly. Their camera resolution was also limited.

In our case, we use wall-mounted cameras to cover large areas. In this current work, we present the results of our method based on the analysis of the shape deformation during the video sequences. Shape deformation gives crucial information on the motion and on the action of the person, and can be discriminant to detect a fall from a normal activity.

2.2. Procrustes Shape Analysis

Procrustes Shape Analysis [3] has been widely used to compare shapes, for example in biology or medicine. Some researchers have also used this method for gait recognition [16][7]. Indeed, Procrustes Shape Analysis is an interesting tool to quantify the human shape deformation. However, in the previous works, the camera is placed sideways and the person is always entirely visible in the image. In our case, we use wall-mounted cameras and the silhouette of the person can be occluded by an object (chair or sofa for example) or when the person enters or leaves the field of view. Moreover, as we use low-cost camera, the silhouette is disturbed by artifacts due to video compression. For this purpose, we chose to track the moving edges of the person which are used as landmarks to compute the full Procrustes distance [3] between two consecutive images. An overview of the system is shown in the next section.

3 System 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 progressively change, while during a fall, the human shape will change drastically. Based on this observation, our fall detection system analyzes the human shape deformation through the image sequence. The main steps of our system are :

Silhouette Edge Points Extraction We first need to find some landmarks from two consecutive images. 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 4.

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 proposed by Belongie et *al.* [1] and described in section 5.

Procrustes Shape Analysis for Fall Detection The human shape deformation can then be quantify using the Procrustes shape analysis method. This step is described in section 6.

4 Silhouette Edge Points Extraction

Procrustes shape analysis has been widely used for gait recognition [2][16]. We propose here to use it for fall detection. Usually, the shape analysis is done using the whole silhouette of the person. To extract this silhouette, we use a background subtraction method which consists to compare the current image with an updated background image. We chose the method described by Kim et *al.* [8] which takes into account shadows, highlights and high image compression. An example of background subtraction is shown in Fig. 1.

However, since we use a low-cost image acquisition system composed of IP cameras, we have to deal with noisy and compressed images. The silhouette segmented from the background is usually not clean enough to be used for shape analysis. Moreover, a shape extracted with a background subtraction method can suffer from occlusions or bad segmentation. To overcome this problem and to find more reliable landmarks, we chose to extract some edge points inside this rough human silhouette. A Canny edge detector is performed to extract all the edges in the image. Combining this image with the crude foreground silhouette, gives the moving edge points of the silhouette. To match two consecutive human shapes, we don't need as many points, so we select N landmarks regularly on each silhouette. We need approximately $N = 250$ landmarks for our experiment. The increment to select regularly the landmarks on the shape is computed as follows. If $nbEdgePts1$ and $nbEdgePts2$

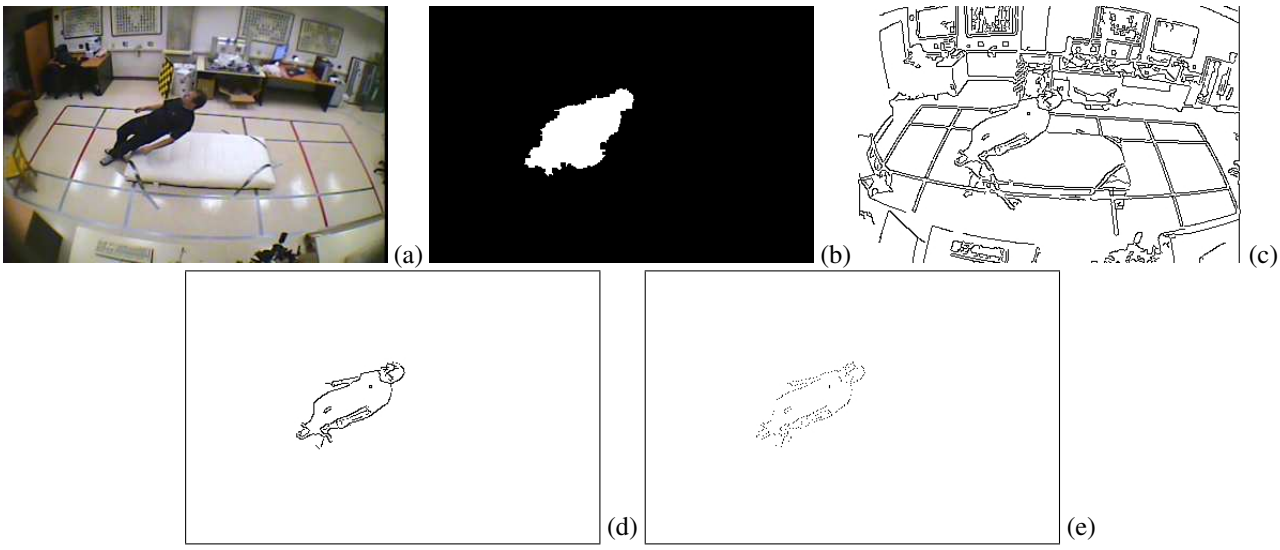

Figure 1. (a) Original image, (b) human shape extracted with the segmentation process, (c) Canny edge detection, (d) moving edge points and (e) selected edge points.

are respectively the number of edge points from the previous and the current silhouette, then the increment is equal to:

$$inc = \frac{\max\left(nbEdgePts1, nbEdgePts2\right)}{N} \tag{1}$$

An example of selected edge points is shown in Fig. 1.

5 Matching using Shape Context

Once we have some silhouette edge points in two consecutive images, we use Shape Context [1] to find the best matching points between the two human shapes. An example of Shape Context matching with edge points has been done by Mori and Malik [11], but unlike them, we discarded unnecessary background edge points with the background subtraction segmentation.

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. In our case, we want to match each point p_i of the previous shape to a point q_j of the current shape.

Shape Context is a shape descriptor that encodes local information about each point relative to its neighbours. 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 :

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

The log-polar histogram is obtained by positionning the polar coordinate system on each edge landmark p_i as shown in Fig. 2.

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 couple of points $\left(p_i, q_j\right)$. This matching cost is computed with the χ^2 test statistic:

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

where $h_i\left(k\right)$ and $h_j\left(k\right)$ 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 $H\left(\pi\right)$ given a permutation $\pi\left(i\right)$:

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

The solution can be found with the Hungarian algorithm [9] for bipartite matching. The input of this algorithm is a square cost matrix with entries C_{ij}, and the result corresponds to the permutation $\pi\left(i\right)$ minimizing $H\left(\pi\right)$.

As we can have some bad landmarks in our selected edge points (due to segmentation errors or partial occlusions), we add some dummy points or outliers in the matching process. We chose to accept 20% of these dummy points with a cost equal to zero, these points will be eliminated at the end of the matching for the next step computation.

The silhouette edge points extracted from two consecutive images are enough reliable to give a good matching

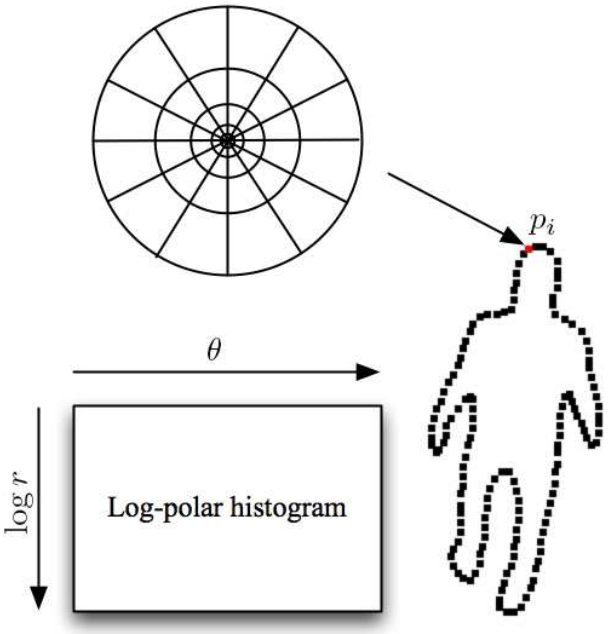

Figure 2. Log-polar histogram computation for a point p_i. The log-polar histogram has 5 bins for $\log r$ and 12 bins for θ as proposed by the authors in [1].

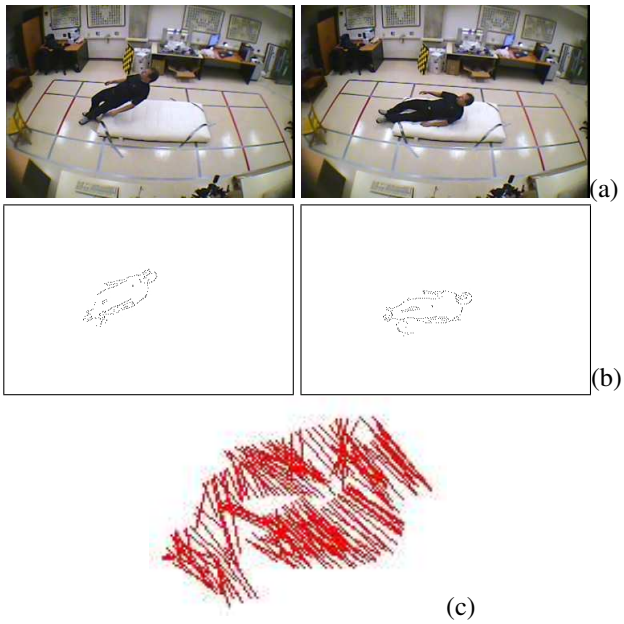

Figure 3. Shape Context matching example: (a) two consecutive images, (b) the corresponding selected edge points and (c) the matching result.

points set for shape analysis. You can see an example of Shape Context matching for two consecutive images in Fig. 3.

6 Fall Detection

Falls can be recognized from normal activities by analyzing the shape deformation through the video sequence. In our work, this deformation is quantified by the full Procrustes distance computed between two consecutive human shapes.

6.1. Kendall Shape Space

To compute the full Procrustes distance between two shapes, we fist need to compute the centred pre-shape Z_C of the original configuration, which is invariant under the translation and scaling [3].

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

$$X = [z_1, z_2, \ldots, z_i, \ldots, z_k], \quad z_i = x_i + jy_i \qquad (5)$$

The centred landmarks X_C are obtained by multiplying the coordinates X with the centring matrix C:

$$X_C = CX \qquad with \;\; C = I_k - \frac{1}{k} 1_k 1_k^T \qquad (6)$$

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

The final centred pre-shape Z_C is obtained by normalizing the centred landmarks:

$$Z_C = CX / \parallel CX \parallel \qquad (7)$$

6.2. Procrustes Analysis

Consider two centred configurations $y = (y_1, \cdots, y_k)$ and $w = (w_1, \cdots, w_k)$, with $y^* 1_k = 0$ and $w^* 1_k = 0$ (y^* denotes the transpose of the complex conjugate of y). A suitable procedure to establish a measure of distance between the two shapes is to match y and w using the similarity transformations. The difference between the fitted and the observed y indicate the magnitude of the difference in shape between w and y [3].

The full Procrustes fit (superimposition) of w onto y is chosen to minimize:

$$D^2(y, w) = \parallel y - w\beta e^{i\theta} - (a + ib)\, 1_k \parallel^2 \qquad (8)$$

with translation $a+ib$, scale $\beta > 0$ and rotation $0 \leq \theta < 2\pi$.

Then, the full Procrustes distance between two centered complex configurations w and y is defined as the minimum full Procrustes fit [3]:

$$\begin{aligned} d_F(w,y) &= inf_{\beta,\theta,a,b} \parallel \frac{y}{\|y\|} - \frac{w}{\|w\|}\beta e^{i\theta} - a - ib \parallel \\ &= \left\{1 - \frac{y^* w w^* y}{w^* w y^* y}\right\}^{1/2} \end{aligned} \tag{9}$$

The denominator $w^* w y^* y = 1$ for the centred pre-shape configuration.

The full Procrustes distance is computed between two consecutive silhouettes during the video sequence. Fig. 4. shows an example of the curve corresponding to the full Procrustes distance along the video sequence of a fall. This curve is smoothed with a Gaussian filter before the analysis step.

Notice that a peak appears on the curve when a fall occurs. This peak is detected by a threshold T_{Df}. The fall is then confirmed by an inactivity period after the peak, corresponding to the person lying on the ground. This assumption is usual in fall detection techniques [13]. This inactivity period is detected by analyzing the mean and the standard deviation of the full Procrustes distance on a short period of time. The fall is confirmed if the mean and the standard deviation are below two thresholds $T_{\bar{Df}}$ and $T_{\sigma_{Df}}$ respectively.

7 Experiments

Our method works with a single uncalibrated camera, the shape matching is implemented in C++ using the OpenCV library [6] and the fall detection step is done with Matlab®.

7.1. Dataset

We wish the final fall detection system to be a low-cost system. So, our video sequences were acquired using an IP camera (Gadspot gs-4600 [4]) with a wide angle to cover all the room. The image size is 360x240 pixels and the frame rate is 30 fps. Our dataset is composed of video sequences representing daily normal activities (walking, sitting down, standing up, crouching down) and simulated falls (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. Some examples are shown in Fig. 5.

Our dataset includes 24 video sequences representing 22 falls and 28 lures, for a total time of 12min51 s. All sequences get the entry of the person in the room, and some sequences show also some moving objects (chairs, boxes) and extra entering/leaving . The falling person wears different clothes in the video sequences (black clothes, white clothes, other color clothes) and put on/off a coat . A mattress was used to protect the person during the simulated falls. The full description of the dataset is presented in the table 1.

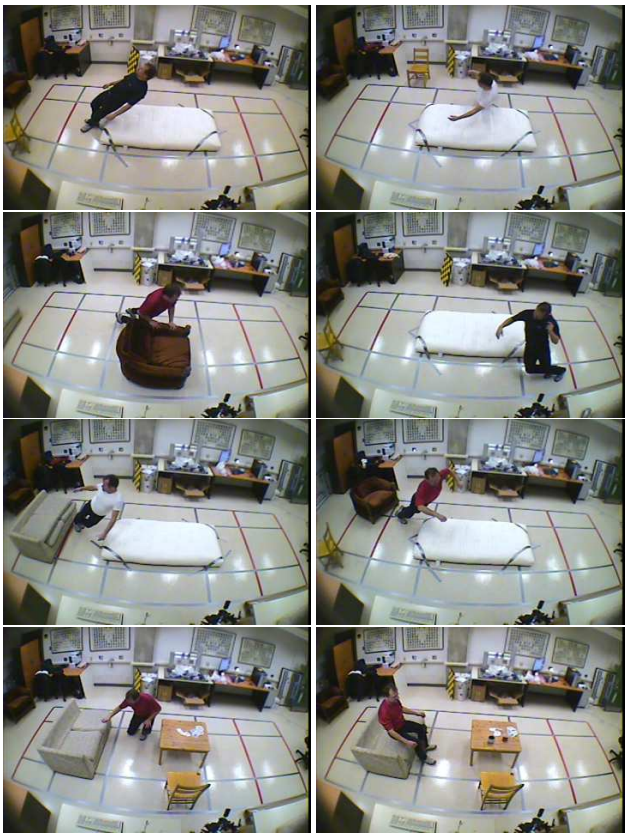

Figure 5. From top to bottom, left to right: backward fall, sideway fall, fall partly slowed down by grabbing the armchair, sideway fall, forward fall from the sofa, forward fall from the armchair, crouching down, siting down

7.2. Frame rate

We have acquired the original video sequences with a frame rate of 30 fps, but 6 fps was found to be sufficient to detect a fall. The original frame rate of 30 fps allows to detect the fall but also the cyclic characteristics of a walking person viewed sideways. If we analyze the deformation with a lower frame rate, the fall is more visible than the cyclic walking characteristics. We thus chose to detect the falls with a frame rate of 6 fps. Currently, the shape matching speed, not optimized, is about 250 ms on an Intel Core 2 Duo processor (2.4 GHz). For future work, it could be improved to run at 6 fps.

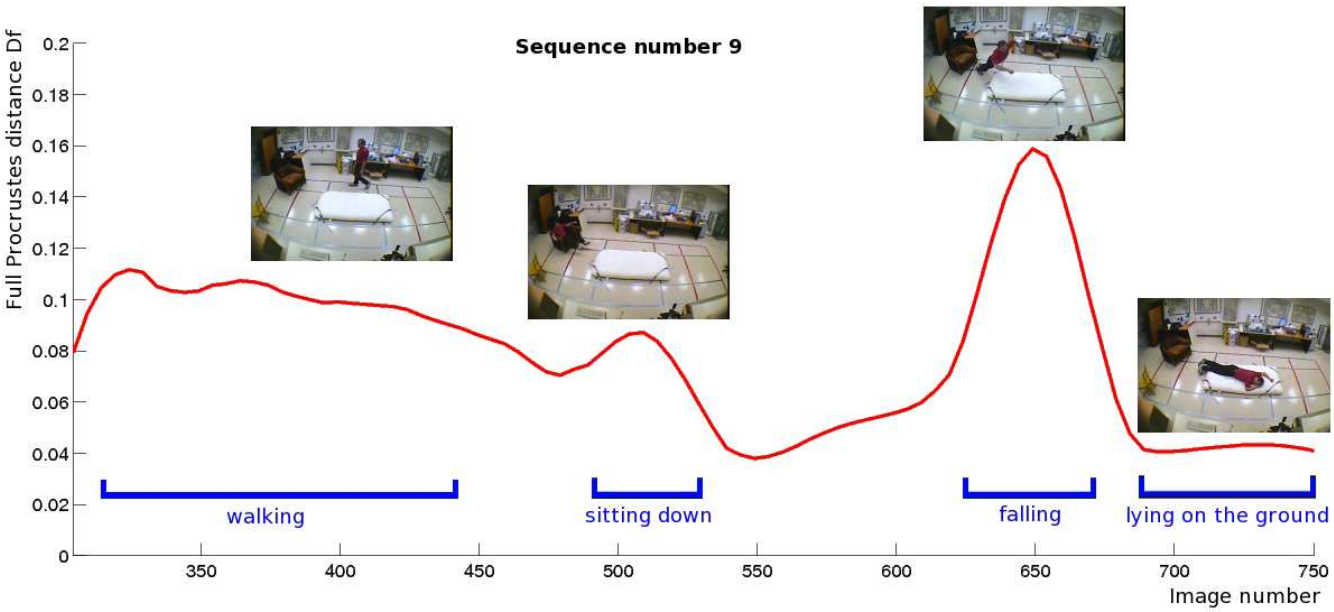

Figure 4. Example of full Procrustes distance D_f curve for a video sequence of a forward fall. We can see the peak corresponding to the fall and the inactivity period after the fall on the curve. This example also shows a lure when the person sits down in the armchair.

7.3. Number of landmarks

Changing the number of landmarks used to track the silhouette produces only a few change on the full Procrustes distance curve, provided that we have a sufficient number of landmarks to estimate D_f ($N \geq 250$ for our experiments). If the number of landmarks increases, the full Procrustes distance will be better estimated, but the computational time will increase due to the matching step. In our experiment, we chose a number of landmarks close to 250 which is a good compromise between accuracy and complexity. Figure 6 shows the D_f curve obtained for $N \simeq 250$ matching points versus the D_f curve computed for all moving edge points. Notice that the full Procrustes distance will be finally estimated with about 200 matching points since we have 20% dummy points in the matching process.

7.4. Results

Our system gave impressive good results in spite of the low-quality images (high compression artifacts, noise) and segmentation difficulties (occlusions like entering/leaving in the field of view, moving objects like chairs, etc).

Table 2 itemizes the fall recognition results obtained with our dataset. We obtained a sensitivity of 95.5% and a specificity of 96.4% with the following thresholds : $T_{Df} = 0.1$, $T_{\bar{D}f} = 0.05$ and $T_{\sigma_{Df}} = 0.01$.

	DETECTED	NOT DETECTED
FALLS	True Positive : 21	False Negative : 1
LURES	False Positive : 1	True Negative : 27

Table 2. Recognition results

One fall was not detected (Sequence number 20) because the fall occurs in two step: first, the person begins to fall but leans against the chair, and finishes the fall later. We can see the two peaks of this fall on the curve in Fig. 7. The first peak is detected but the fall is not confirmed later because of the fall continuation. The second peak is not detected with a threshold of $T_{Df} = 0.1$, since the end of the fall is slower. The other mistake corresponds to a lure detected as a fall (Sequence number 24) because the person brutally sits down in the sofa.

We perform a ROC analysis to study the influence of the three thresholds on the results. Figure 8 shows the ROC convex hull obtained with the threshold variations described in table 3.

The ROC convex hull shows that our method is efficient in spite of the bad video quality with an accuracy (measured by the area under the ROC curve (AUC)) of 0.9889.

Sequence number	Duration (s)	Number of falls	Number of lures	Description
1	20	1	0	fall because of an imbalance when the person puts on his coat
2	17	1	0	fall due to a leg deficiency
3	19	1	0	forward fall
4	25	1	0	fall due to a leg deficiency
5	13	1	0	forward fall due to a carpet
6	24	1	0	backward fall
7	18	1	0	sideway fall
8	15	1	0	sideway fall
9	23	1	1	the person sits down in the armchair, forward fall
10	18	1	0	fall when the person sits down
11	15	1	0	sideway fall
12	21	1	0	sideway fall
13	28	1	1	the person sits down, fall because of an imbalance when the person gets up from the sofa
14	33	1	1	the person sits down, fall because of an imbalance when the person gets up from the sofa
15	23	1	1	the person sits down, fall because of an imbalance when the person gets up from the sofa
16	29	1	3	the person crouches down, crouches down again, sits down in the sofa, fall because of an imbalance when the person gets up from the sofa
17	27	1	3	the person crouches down, crouches down again, sits down in the sofa, fall because of an imbalance when the person gets up from the sofa
18	22	1	1	the person sits down in the armchair, fall due to simulated Parkinson deficiency
19	23	1	0	fall slowed down by grabbing the armchair
20	24	1	0	fall slowed down by grabbing the armchair
21	27	1	1	the person sits down in the armchair, forward fall from the armchair
22	27	1	1	the person sits down in the armchair, fall from the armchair with the buttock hitting the floor
23	170	0	9	the person crouches down, crouches down again, fall without gravity, crouches down again, sits down brutally on the sofa, sits down brutally on the sofa, fall without gravity from the sofa, crouches down again, sits down
24	110	0	6	the person crouches down during housekeeping, crouches down to pick up the broom, crouches down during housekeeping, crouches down during housekeeping, coat falling on the floor and the person crouches down to pick up his coat
TOTAL:	771	22	28	

Table 1. Description of the dataset

THRESHOLD	RANGE	INCREMENT
T_{Df}	$[0.05 \dots 0.5]$	0.01
$T_{\bar{D}f}$	$[0.02 \dots 0.1]$	0.002
$T_{\sigma_{Df}}$	$[0.002 \dots 0.04]$	0.001

Table 3. Threshold variations used for ROC analysis

8 Conclusion and Discussion

In this work, we propose a new method to detect falls of elderly people. Falls can be recognized from normal activities by analyzing the shape deformation. The Procrustes shape analysis is a useful tool to detect some changes in the silhouette through the video sequence. Our fall detection system has proven its robustness on realistic image sequences of simulated falls and daily activities. For future work, we will use a training dataset and learning algorithms to automate the thresholds setting.

9 Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[2] J. Boyd. Video phase-locked loops in gait recognition. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 696–703, 2001.

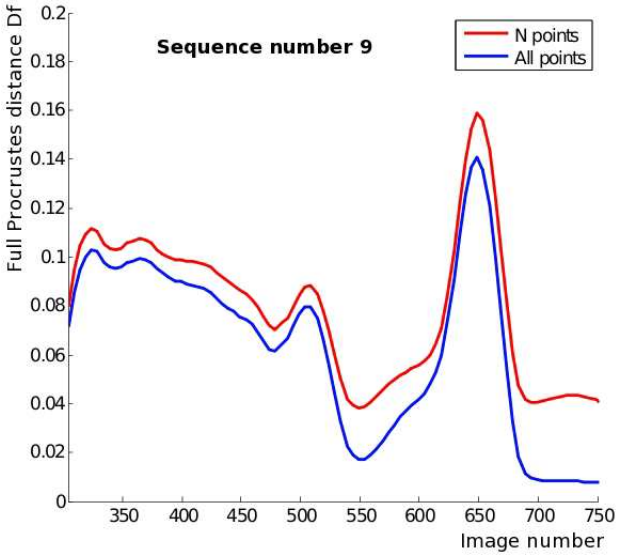

Figure 6. D_f versus number of landmarks

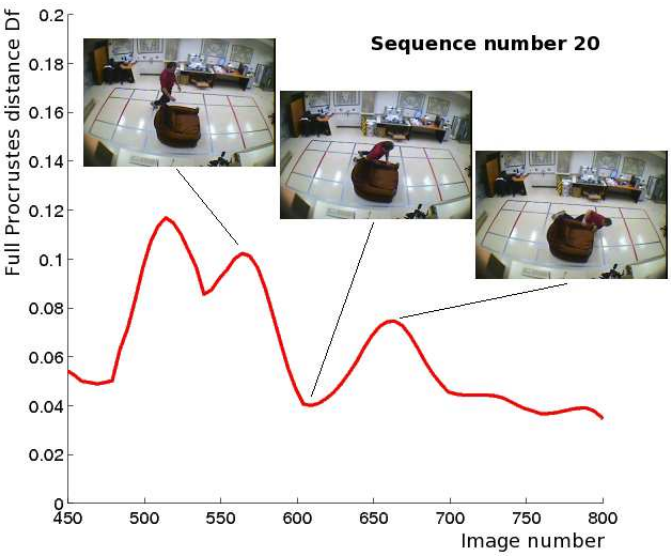

Figure 7. Fall slowed down by grabbing the armchair

[3] I. Dryden and K. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, Chichester, 1998.

[4] Gadspot inc. *IP camera Gadspot*. http://gadspot.com.

[5] Health Canada. *Canada's Aging Population*, 2002. Public Health Agency of Canada, Division of Aging and Seniors.

[6] Intel Open Source Computer Vision Library. *OpenCV*. http://www.intel.com/technology/computing/opencv.

[7] N. Jin and F. Mokhtarian. Human motion recognition based on statistical shape analysis. In *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2005)*, pages 4–9, 2005.

[8] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.

[9] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[10] T. Lee and A. Mihailidis. An intelligent emergency response system: preliminary development and testing of automated fall detection. *Journal of telemedicine and telecare*, 11(4):194–198, 2005.

[11] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision LNCS 2352*, volume 3, pages 666–680, 2002.

[12] H. Nait-Charif and S. McKenna. Activity summarisation and fall detection in a supportive home environment. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 323–326, 2004.

[13] N. Noury, A. Fleury, P. Rumeau, A. Bourke, G. Laighin, V. Rialle, and J. Lundy. Fall detection - principles and methods. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2007)*, pages 1663–1666, 2007.

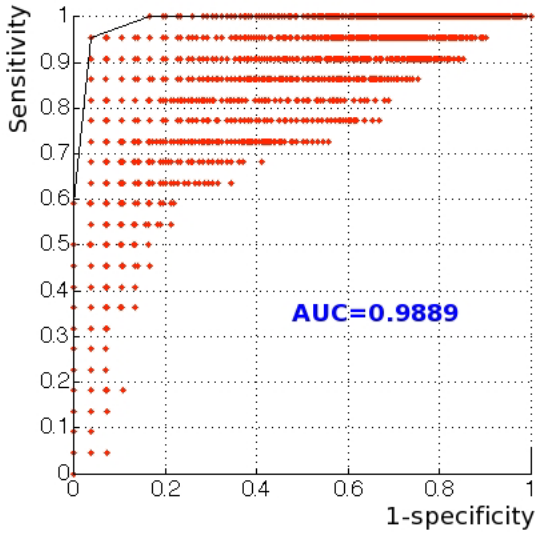

Figure 8. ROC convex hull

[14] A. Sixsmith and N. Johnson. A smart sensor to detect the falls of the elderly. *IEEE Pervasive Computing*, 3(2):42–47, 2004.

[15] B. Töreyin, Y. Dedeoglu, and A. Çetin. Hmm based falling person detection using both audio and video. In *IEEE International Workshop on Human-Computer Interaction*, 2005.

[16] L. Wang, H. Ning, W. Hu, and T. Tan. Gait recognition based on procrustes shape analysis. In *IEEE International Conference on Image Processing (ICIP)*, volume 3, pages 433–436, 2002.