A ROBUST FALL DETECTION SYSTEM FOR THE ELDERLY IN A SMART ROOM

Miao Yu, Syed Mohsen Naqvi and Jonathon Chambers

Advanced Signal Processing Group, Electronic and Electrical Engineering Department
Loughborough University, Loughborough, Leicester, UK
{elmy, s.m.r.naqvi, eljac}@lboro.ac.uk

ABSTRACT

In the paper, we propose a robust fall detection method which combines head tracking and extraction of human shape within a smart home environment equipped with video cameras. A motion history image and an improved code-book background subtraction technique are combined to extract the human shape. An additional motion-based particle filtering head tracker is also used to ensure the robustness of the system. The extracted human shape information and the head tracking results are combined as criteria for judging the occurrence of a fall. The success of the method is confirmed on real video sequences.

Index Terms— motion history image, code-book background subtraction, motion-based particle filtering, head tracking, fall detection

1. INTRODUCTION

In recent years, there has been increasing public concern related to the topic of caring for old people. Falls are the leading cause of death due to injury and a large number of fractures are caused by falls among the unexpected events which happen in the elderly group[]. Traditional methods to detect falls use some wearable sensors. However, the problem arises with respect to such detectors because older people often forget to wear them and they are intrusive. In order to solve this problem, fall detection based on the use of digital video processing technology is developed.

There are many related works in the field of fall detection based on digital video processing techniques [1] [2] [3] and [4]. Lee and Mihailidis [1] put a camera in the ceiling to track the person's movements and a fall is detected by the observation that long time inactivity happens outside the normal zones of inactivity such as chairs or sofas. Hazelhoff and With [2] adopted a system with two uncalibrated cameras. A Gaussian multi-frame classifier helps to recognize fall events using the two features of the direction of the main axis of the body and the ratio of the variances in the motion in the x and y directions. In [3] and [4], Rougier detects falls by head tracking and human shape analysis respectively.

In this paper, we propose a fast reacting and robust fall detection method based on the combination of human shape analysis and head tracking. This method can detect a fall event when it is happening so the alarm signals can be sent immediately; moreover, compared to C.Rougier's methods, an improved background subtraction method and a more elegant head tracking approach are adopted and the combination makes the system more robust. The structure of this paper is as follows: In Section 2, we briefly give the structure of our fall detection system. In Section 3, we propose a motion-based particle filtering method for head tracking. In Section 4,we introduce how to extract human shape information based on a motion history image (MHI) and background subtraction method. Some experimental results are shown in Section 5 and in Section 6, conclusions are drawn and future work suggested.

2. SYSTEM OVERVIEW

A good fall detection system should detect a fall as quickly as possible for a person to get early treatment, so, it is necessary to detect the fall as it is happening, rather than detect it some time after its occurrence. We exploit the head's 3-D position and the human shape information as they provide the cues for detecting the ongoing fall. Our fall detection system aims at detecting the fall as it is happening and overcoming the problem that using human shape information or the head's 3-D information alone will fail under certain cases, so we utilize their combination for a robust detection. The system consists of two calibrated cameras which both cover the normal activity area. Two video streams are recorded and human shape analysis and 2-D head tracking are applied respectively on them to detect a fall.

The block-diagram in Figure 1 shows the structure of our fall detection system. The details of how every corresponding block works are discussed in the later sections.

3. HEAD TRACKING

In order to obtain the 3-D information of the head (head's 3-D position and velocity), head tracking is needed. Feature-points-based or model based 3-D head tracking can estimate the head's 3-D pose by using a single camera. However, in the application of fall detection, the camera's view covers nearly the whole room environment so the head only covers a small percentage of a whole cluttered background image so it is difficult to obtain features. Besides, sometimes the face is totally invisible and the model-based method will definitely fail. So, instead of 3-D head tracking, we use 2-D head tracking for two video streams obtained from two calibrated cameras and obtain the head's 3-D position from the tracking results by the method introduced in [5]. For 2-D head tracking, we use a more elegant motion based particle filtering method compared to [3] .

Motion-based particle filtering is superior when compared to the traditional generic particle filter based on the condensation algorithm. The underlying formulation of the motion-based particle filtering algorithm we use is the same as that of the generic one. But its proposal distribution is different and is related to the output of an

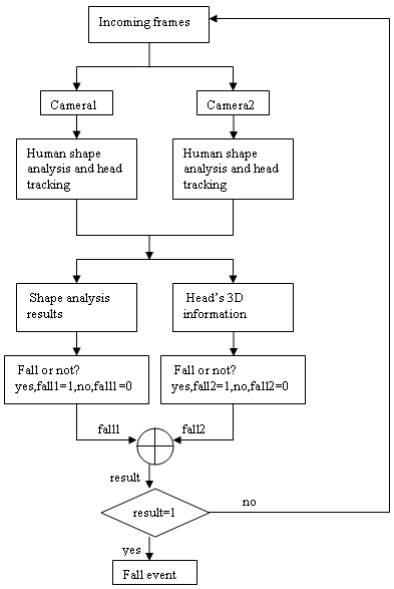

Fig. 1: The block-diagram describing the operation of our fall detection system

adaptive block matching (ABM) operation [6], which yields better results in cluttered environments.

3.1. Motion-based particle filter

The motion based particle filtering algorithm is based on the Bayesian sequential importance sampling scheme. We assume in 2-D tracking, the head is modeled as an ellipse and its state vector is $\mathbf{S} = [x, y, l, \theta]^T$, where (x,y) is the center of the ellipse representing for a head, l is the length of the minor semi-axis (we assume the ratio between the major and minor axis is fixed, 1.2) and θ is the ellipse's orientation. And at time instant t-1, if we have N particles $\{\mathbf{S}_{t-1}^n\}_{n=1:N}$ and their corresponding weights $\{w_{t-1}^n\}_{n=1:N}$ and they meet $\hat{\mathbf{S}}_{t-1} = \sum_{i=1}^N w_{t-1}^i \mathbf{S}_{t-1}^i$, where $\hat{\mathbf{S}}_{t-1}$ is the minimum mean square estimation (MMSE) for $\mathbf{S}_{t-1}$. We can then sample from an importance function $q(\mathbf{S}_t)$ to get N new samples for time t, and the corresponding weights are calculated as:

$$w_t^n = \frac{\sum_{j=1}^N w_{t-1}^j p(\mathbf{S}_t^n|\mathbf{S}_{t-1}^j)}{q(\mathbf{S}_t^n)} p(\mathbf{Z}_t|\mathbf{S}_t^n) \quad (1)$$

where $\mathbf{Z}_t$ is the measurement vector, $p(\mathbf{S}_t|\mathbf{S}_{t-1})$ is called the proposal distribution of $\mathbf{S}_t$ given $\mathbf{S}_{t-1}$ and $p(\mathbf{Z}_t|\mathbf{S}_t)$ is the distribution for the measurement given the state vector.

The MMSE at time t can be calculated according to the new $\{\mathbf{S}_t^n\}_{n=1:N}$ and $\{w_t^n\}_{n=1:N}$.

In a motion-based particle filtering algorithm, the proposal distribution $q(\mathbf{S}_t)$ has the following form:

$$q(\mathbf{S}_t) = \sum_{i=1}^N N_{\mathbf{S}_t}(\mathbf{S}_{t-1}^i + \Delta\mathbf{S}_t, \sum_G) \quad (2)$$

$N_{\mathbf{x}}(\mathbf{u}, \sum)$ is a multivariate Gaussian distribution with the following holds: $N_{\mathbf{x}}(\mathbf{u}, \sum) = \frac{1}{(2\pi)^{D/2}|\sum|^{1/2}} exp(-(\mathbf{x}-\mathbf{u})^T \sum^{-1}(\mathbf{x}-\mathbf{u}))$, $\mathbf{u}$ and $\sum$ are the mean and covariance matrix of the D-dimensional vector $\mathbf{x}$ and $|\cdot|$ denotes the determinant operator.

Straightforwardly, we can get a sampling scheme as follow:

$$\mathbf{S}_t^n = \mathbf{S}_{t-1}^n + \Delta\mathbf{S}_t + \mathbf{v}_t^n \quad (3)$$

where $\mathbf{v}_t \sim N(0, \sum_G)$.

The parameter $\Delta\mathbf{S}_t$ is calculated as: $\Delta\mathbf{S}_t = [x_c(t) - x_c(t-1), y_c(t) - y_c(t-1), 0, 0]$, in which $[x_c(t), y_c(t)]$ is the center of ellipse which best fits the ABM output at frame t. The details of how to calculate the ABM output are shown in [7].

In order to complete the algorithm description, we should also know $p(\mathbf{Z}_t|\mathbf{S}_t^i)$, in [8], it is shown that $p(\mathbf{Z}_t|\mathbf{S}_t^i) = p(\mathbf{Z}_{t,gradient}|\mathbf{S}_t^i)p(\mathbf{Z}_{t,color}|\mathbf{S}_t^i)$.

The calculations of $p(\mathbf{Z}_{t,gradient}|\mathbf{S}_t)$ and $p(\mathbf{Z}_{t,color}|\mathbf{S}_t^i)$ are found in details in [8]. The complete algorithm is as follow [6]:

1.From the previous estimated $\hat{\mathbf{S}}_{t-1}$, run the adaptive block matching module. Compute $\Delta\mathbf{S}_t$.

2.Re-sampling from the old sample set $\{\mathbf{S}_t^n, w_t^n\}_{n=1:N}$ to obtain a new set $\{\mathbf{S}_t^n, 1/N\}_{n=1:N}$.

3.Use the sampling scheme in equation (2) to draw N samples.

4.Weight the samples according to equation (1) and normalize them.

5.Calculate the weighted average of all particles at time t to get the new estimate.

This concludes the motion-based particle filtering algorithm. We track the head in two video streams, and the 3-D positions and velocities of the head are then calculated from the 2-D tracking result obtained from the two video streams. If both the horizontal and vertical velocities exceed corresponding thresholds and the head's height is below a certain threshold. A fall is detected.

4. HUMAN SHAPE ANALYSIS

In order to be successful to detect a fall when head tracking fails in a very cluttered home environment. Human shape analysis is needed to add robustness to the system and it is applied in both two cameras. For human shape analysis, we analyze the change of the human shape when a large movement is detected, the determination of the movement's magnitude is based on a parameter C_{motion}, which is calculated from the motion history image and background subtraction result. For the human shape analysis, we fit an ellipse to the region of human and the variations of its orientation θ and the ratio between its major and minor semi-axes in an interval are calculated to determine a fall.

4.1. Movement Classification

First, we use the method introduced in [9] to obtain the motion history image(MHI), and a code-book background subtraction method in [10] to get the binary result of background subtraction. Then we determine whether the movement is large or small by calculating the parameter C_{motion}. The formula to calculate C_{motion} is shown as follow [4]

$$C_{motion} = \frac{\sum_{pixel(x,y)\in blob} H_\tau(x, y, t)}{\sharp pixels \in blob} \quad (4)$$

where *blob* represents the region of the person extracted using the code-book background subtraction, and $H_\tau(x, y, t)$ represents the value of pixel(x,y) in the t-th motion history image $H_\tau(x, y, t)$, where τ is a parameter called duration time defined in [9]. The denominator $\sharp pixels \in blob$ can be obtained by multiplying the number of pixels in the extracted person's region with 255. $C_{motion} = 0$ represents no motion and $C_{motion} = 1$ represents full motion. We calculate the $C_{motion,camera1}$ and $C_{motion,camera2}$ of the two cameras and use a max operator to obtain the final $C_{motion} = max(C_{motion,camera1}, C_{motion,camera2})$. We assume that if $C_{motion} > 0.65$, then large movement occurs.

4.2. Human Shape Analysis

After a large movement is detected, we can make a decision on whether a fall occurs or not by further analyzing the changing of the shapes for both camera recordings in an interval. For every video recording, an approximated ellipse is fitted to the human blob. The fitting of the blob can be achieved by using moments [4].

For an image f(x,y), the moments are given by: $m_{pq} = \sum_{(x,y)\in Pixels} x^p y^q f(x, y)$, where p, q = 0, 1, 2 ..... We can obtain the centroid $(\bar{x}, \bar{y})$of the ellipse by: $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{0\cdot}/m_{00}$.

After obtaining the centroid, we can compute the central moment in the following way: $u_{pq} = \sum_{(x,y)\in Pixels}(x-\bar{x})^p(y-\bar{y})^q f(x, y)$.

With the aid of central moments, the orientation of the ellipse can be calculated as:

$$\theta = \frac{1}{2}\arctan(\frac{2u_{11}}{u_{20} - u_{02}}) \tag{5}$$

and the major semi-axis a and the minor semi-axis b of the fitted ellipse can be calculated as: $a = (4/\pi)^{1/4}[\frac{(I_{max})^3}{I_{min}}]^{1/8}$ and $b = (4/\pi)^{1/4}[\frac{(I_{min})^3}{I_{max}}]^{1/8}$ where I_{max} and I_{min} are the large and small eigenvalues of J, where

$$J = \begin{pmatrix} u_{20} & u_{11} \\ u_{11} & u_{02} \end{pmatrix} \tag{6}$$

After obtaining the parameters, we calculate the variations of the orientation and the ratio of a and b, namely σ_θ and σ_ρ in a time interval (here we set the interval as 1s) for both camera recordings. Similar to the calculation of C_{motion}, a max operator is applied to obtain the final σ_θ and σ_ρ from the results of both camera recordings. If either σ_θ or σ_ρ exceeds certain threshold, a fall is detected.

5. EXPERIMENTS AND EVALUATIONS

Figures 2 shows the MHI and background subtraction results for four situations (fast walking, slow walking, sitting down and a fall), an ellipse is fitted to the output of the background subtraction.

Figure 3 shows some tracking results of 2-D head tracking.

We analyze three cases of falling. The first one denoted Case 1 is a recording for people doing a series of activities—walking fast, walking slow, sitting down, standing up then falling over. The second one denoted Case 2 is a slow fall event in which a person slips slowly from the chair to the ground. The last one denoted Case 3 is where a fall happens when the head tracking fails due to the cluttered environment. Figures 4-6 show the variations of the parameters which are used for determining the fall.

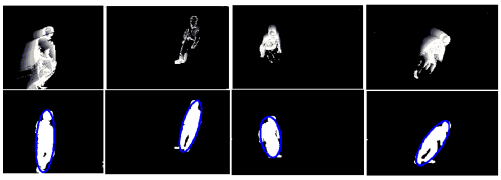

Fig. 2: MHI and background subtraction results for fast walking, slow walking, sitting down and falling (from left to right)

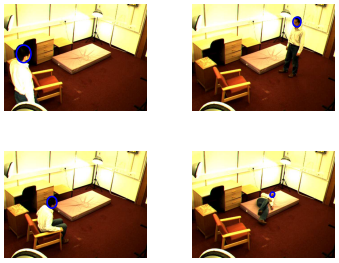

Fig. 3: Some head tracking results

Figure 4 shows the variations of horizontal and vertical velocities of the head for the three cases. In Case 1, we can see initially, the person walks very fast so the horizontal velocity is high but the vertical velocity is low, for the slow walking and sitting and standing period, neither of the velocities exceeds the corresponding threshold. At 20s after the activity starts, a fall happens and both the horizontal and vertical velocities exceed the thresholds. In Case 2, because it is a slow fall so the head's movement is very slow and the velocities are below the thresholds. In Case 3, head tracker just locks at a fixed position so both velocities are approximate to zeros.

Figure 5 shows the variation of the head's height for the three cases. We can see in the first case, the head's height varies with time according to different activities and at the end of Case 1 when a fall happens, the head's height decreases drastically to the below of the threshold (0.5m). So does it at the end of the Case 2. In Case 3, the estimated head's height is fixed due to the failure of the head tracker.

Figure 6 shows the C_{motion}, σ_θ and σ_ρ's changes for the three cases. We can see at the end of Case 1, the σ_θ exceeds the threshold (30 degree). In Case 2, because C_{motion} is under the threshold (65%) during the whole period, the corresponding σ_θ and σ_ρ are not calculated and set to zeros. In Case 3, a fall happens at the end of it (at 40s) and the corresponding σ_θ exceeds the threshold.

We can see in the first case. When a fall happens at 20s after start, the σ_θ exceeds the threshold. And the head's 3-D velocities and its height also exceed the set threshold. A fall is detected. In the second case, although C_{motion} does not exceed the threshold (65%) and neither do the velocities. But from the head's height information, we know something abnormal occurs. In the third case, although the head tracking fails due to head trackers lock at the environment part which is similar to the head, fall is still detected because the σ_θ exceeds the threshold when a fall happens.

We test our fall detection system on several datasets which are composed of some normal activities and compare with other fall de-

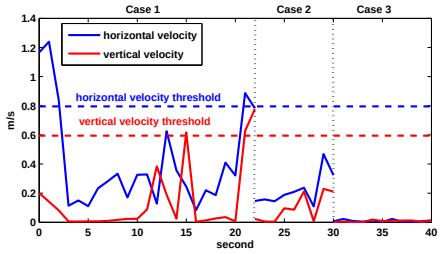

Fig. 4: Variations of velocities and of head in 25s

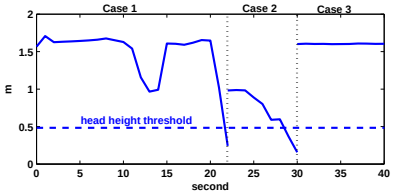

Fig. 5: Variations of head's height for three cases

tection methods. The comparison results are given in Table 1.

From Table 1, we can see that our method achieve the highest fall detection rate quickly. Although its false detection rate (the rate of mistaking other activities as fall) is a bit higher than that of C.Rougier's method in [3] and Hazelhoff and With's mehtod in [2]. But considering the risk posed by no detection of a true fall is much larger than mistaking other activity as a fall. So, a high fall detection rate should be taken as priority and our method is the best one.

6. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a new fall detection system based on head tracking and human shape analysis. This system is composed of two calibrated cameras, and 2-D head tracking and human shape analysis are operated on both video recordings recorded by the two cameras. Finally, the head's 3-D velocities and shape information and the extracted human shape information σ_θ and σ_ρ are used as criteria for determining a falling event. Experimental results show it is superior than other fall detection methods.

A more robust fall detection system can potentially be achieved by the combination of audio and video information, which is well known as multimodal processing. Blind source separation technique can be applied to extract the person's voice information, such as the words such as 'help', from the noisy environment. And a speech recognition system could then be used to analyze the extracted voice

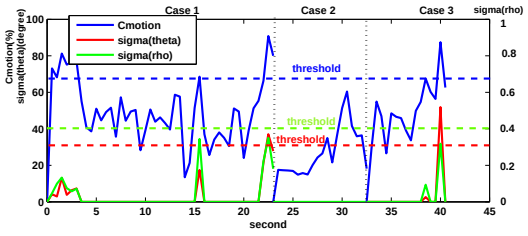

Fig. 6: Variations of C_{motion}, σ_θ and σ_ρ for three cases

Table 1: The comparison between different fall detection methods.

Method	Detection time	Fall detection (%)	False detection (%)
Our method	As fall happens	92.6	12.5
Caroline et al. [3]	As fall happens	66.7	10
Caroline et al. [4]	As fall happens	88.2	12.5
Lee and Mihailidis [1]	Certain time after falling over	77	15
Hazelhoff and With [2]	Certain time after falling over	85	7

to make a decision on whether a fall may have occurred.

7. REFERENCES

[1] T. Lee and A. Mihailidis, "An intelligent emergency response system: preliminary development and testing of automated fall detection," *Journal of Telemedicine and Telecare*, vol. 11, pp. 194–198, 2005.

[2] L. Hazelhoff, J. Han, and P. H. With, "Video-based fall detection in the home using principal component analysis," *In Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems*, vol. 11, pp. 298–309, 2008.

[3] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Monocular 3d head tracking to detect falls of elderly people," *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pp. 6384–6387, 2006.

[4] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," *In Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, vol. 2, pp. 875–880, 2007.

[5] D. Gatica-Perez, G. Lathoud, J. Odobez, and I. McCowan, "Audiovisual probabilistic tracking of multiple speakers in meetings," *IEEE transaction on audio,speech and language processing*, vol. 15, pp. 601–616, February 2007.

[6] N. Bouaynaya, Q. Wei, and D. Schonfeld, "An online motion-based particle filter for head tracking applications," *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings*, vol. 2, pp. 225–228, 2005.

[7] K. Hariharakrishnan and D. Schonfeld, "Fast object tracking using adaptive block matching," *IEEE Transactions on Multimedia*, vol. 7, pp. 853–859, 2005.

[8] X. Xu and B. Li, "Head tracking using particle filter with intensity gradient and colour histogram," *In IEEE International Conference on Multimedia and Exploration*, pp. 888–891, 2005.

[9] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 257–267, 2001.

[10] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using code-book model," *Real-Time Imaging*, vol. 11, pp. 172–185, June 2005.