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 this paper, we propose a novel and robust fall detection system by using a density method for modeling a fall event as a function of certain video feature.3-D head velocity and human shape information are extracted as feature and three types of density model, single Gaussian, mixture of Gaussians and Parzen window method, are constructed for modeling the density of fall with respect to the extracted video feature. Falls are then detected according to the corresponding obtained density model and the success of the method is confirmed on real video sequences.

Index Terms— code-book background subtraction, motion-based particle filtering, head tracking, density method, 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 among the unexpected events which happen in the elderly group. Such falls can also result in fractures. Traditional methods to detect falls use some wearable sensors. However, a 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, 4, 5]. 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 et al. [2] adopted a system with two uncalibrated cameras. A Gaussian multi-frame classifier helps to recognize fall events using the two feature 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], Yu et al. detect falls according to the 3-D horizontal and vertical velocity information of the head obtained by head tracking with a variable state model particle filter. In [4, 5], Rougier et al. detect falls by head tracking and human shape analysis respectively.

In this paper, we propose a more robust fall detection system based on estimating the density of a fall with respect to corresponding video feature, and falls are then detected according to the obtained density information. The structure of this paper is as follows: In Section 2, we describe how to extract the video feature—the head velocity and human shape variation information. In Section 3, we show the construction of the density model by using a single Gaussian, a mixture of Gaussians and a Parzen window method. Some experimental results are shown in Section 4 and in Section 5, conclusions are drawn and future work suggested.

2. VIDEO FEATURE EXTRACTION

For video feature, we use the head's 3-D velocity and human shape's variation information. To extract them, head tracking and human shape analysis techniques are applied. Two cameras whose optical axis are orthogonal to each other are used for video feature acquisition.

2.1. Head tracking

In order to obtain the 3-D information of the head (heads 3-D position and velocity), head tracking is needed. We apply 2-D head tracking for two video streams obtained from two calibrated cameras and obtain the heads 3-D position from the 2-D tracking results. For 2-D head tracking, we use a more elegant motion-based particle filtering method. 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 adaptive block matching (ABM) operation [6], which yields better results in cluttered environments, because the processing focuses on image areas where the head is very likely to be present.

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 at 1.2) and θ is the ellipse's orientation angle. And at time instant $t-1$, if we have N particles $\{\mathbf{S}_{t-1}^n\}_{n=1:N}$ and their corresponding weights $\{w_{t-1}^n\}_{n=1:N}$, the following holds: $\hat{\mathbf{S}}_{t-1} = \sum_{i=1}^{N} w_{t-1}^i \mathbf{S}_{t-1}^i$, where $\hat{\mathbf{S}}_{t-1}$ is the minimum mean square estimate (MMSE) for $\mathbf{S}_{t-1}$. We can then sample from an importance function $q(\mathbf{S}_t)$ to obtain N new samples for time t, and the corresponding weights are calculated as:

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

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:

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

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

Straightforwardly, we can get a sampling scheme as follow:

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

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 the 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 detail 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 by using the camera calibration technique proposed in [9]. The horizontal and vertical head velocities will be taken as feature.

2.2. Human shape analysis

Human shape analysis is applied to both camera outputs. For human shape analysis, an ellipse is fitted to the region of the human and the variation of its orientation θ in an interval is calculated.

For every video recording, an approximate ellipse is fitted to the human blob. The fitting of the blob can be achieved by using moments [5].

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 $\bar{y} = m_{01}/m_{00}$.

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:

$$\theta = \frac{1}{2}\arctan(\frac{2u_{11}}{u_{20} - u_{02}}) \tag{4}$$

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 larger and smaller eigenvalues of J, where

$$J = \begin{pmatrix} u_{20} & u_{11} \\ u_{11} & u_{02} \end{pmatrix} \tag{5}$$

After obtaining the parameters, we calculate the variation of the orientation, namely σ_θ in a time interval (here we set the interval as 1s) for both camera recordings. A max operator is applied to obtain the final σ_θ from the results of both camera recordings.

Finally, the obtained head velocities and σ_θ are combined as video feature and those correspond to the fall case are used to train the density model. The construction of three different kinds of density models is shown in the next section.

3. DENSITY MODEL ESTIMATION

Density method, which is one kind of one classification method, can be used for fall detection by firstly constructing the fall density $p(\mathbf{x}|fall)$, then selecting the proper threshold th. If for an incoming feature point $\mathbf{z}$, $p(\mathbf{z}|fall) > th$, then a fall is determined; otherwise, it is regarded as non-fall. $p(\mathbf{x}|fall)$ can be assumed to be single Gaussian, mixture of Gaussians or estimated by the Parzen window method. The receiver operating characteristic (ROC) curve [10] is analyzed to obtain the optimal value of th for achieving the best performance.

3.1. Single Gaussian model

The density $p(\mathbf{x}|fall)$ can be modeled by a single Gaussian with the form:

$$p(\mathbf{x}|fall) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp(-(\mathbf{x}-\mathbf{u})^T\Sigma^{-1}(\mathbf{x}-\mathbf{u})) \tag{6}$$

where $\mathbf{u}$ is the mean and Σ is the covariance matrix.

By the maximum likelihood (ML) method, we can get the estimated $\mathbf{u}_{ML}$ and Σ_{ML} as:

$$\mathbf{u}_{ML} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n \tag{7}$$

$$\Sigma_{ML} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}-\mathbf{u}_{ML})(\mathbf{x}-\mathbf{u}_{ML})^T \tag{8}$$

3.2. Mixture of Gaussians model

However, for a non-Gaussian density distribution, a single Gaussian model can not always accurately model the target distribution. We can then refer to a mixture of Gaussians model, which is a linear combination of Gaussian distributions. If we model $p(\mathbf{x}|fall)$ by a mixture of Gaussians, we have:

$$p(\mathbf{x}|fall) = \sum_{k=1}^{K}\pi_k N(\mathbf{x}|\mathbf{u}_k, \Sigma_k) \tag{9}$$

Each Gaussian density $N(\mathbf{x}|\mathbf{u}_k, \Sigma_k)$ is called a component of the mixture and has its own mean $\mathbf{u}_k$ and covariance Σ_k. And π_k, k

= 1,...,N are the mixing the mixing coefficients which satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

The corresponding parameters $\mathbf{u}_k$, Σ_k and π_k can be estimated by the expectation-maximization algorithm which is shown in [11], the steps are shown as follows:

1. Initialize the means $\mathbf{u}_k$, covariances Σ_k and mixing coefficients π_k, and evaluate the initial value of the log likelihood.

2. E step. Evaluate the responsibilities, $\gamma(z_{nk})$ using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k N(\mathbf{x}_n|\mathbf{u}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(\mathbf{x}_n|\mathbf{u}_j, \Sigma_j)}$$

3. M step. Re-estimate the parameters using the current respons ities

$$\mathbf{u}_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \mathbf{u}_k^{new})(\mathbf{x}_n - \mathbf{u}_k^{new})^T \quad (12)$$

$$\pi_k^{new} = \frac{N_k}{N} \quad (13)$$

where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$.

4. Evaluate the log likelihood

$$\text{In } p(X|\mathbf{u}, \Sigma, \pi) = \sum_{n=1}^{N} \text{In}\{\sum_{k=1}^{K} \pi_k N(\mathbf{x}|\mathbf{u}_k, \Sigma_k)\} \quad (14)$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

Typically, the number of components K is chosen between 3-5. In this paper, we choose K=5.

3.3. Parzen window method

Parzen window method is a non-parametric density estimation method, there is no model by which the density distribution is assumed to follow and the density is estimated by:

$$p(\mathbf{x}|fall) = \sum_{n=1}^{N} k(\mathbf{x}; \mathbf{x}_n) \quad (15)$$

where $\mathbf{x}_n$ is a training data point and $k(\cdot)$ is called the Parzen window, which can be any kind of kernel function meeting $k(\mathbf{u}) \geq 0$ and $\int k(\mathbf{u}) d\mathbf{u} = 1$.

In order to obtain a smooth density estimation, we choose the Gaussian kernel $k(\mathbf{x}; \mathbf{x}_n) = \frac{1}{(2\pi h^2)^{1/2}} exp\{-\frac{\|\mathbf{x}-\mathbf{x}_n\|^2}{2h^2}\}$and the estimated density is:

$$p(\mathbf{x}|fall) = \sum_{n=1}^{N} \frac{1}{(2\pi h^2)^{1/2}} exp\{-\frac{\|\mathbf{x} - \mathbf{x}_n\|^2}{2h^2}\} \quad (16)$$

The parameter h can be estimated from the ML method.

4. EXPERIMENTS AND EVALUATIONS

The experiments are carried out in Loughborough University's Smart Room. Two calibrated cameras both covering the area where a person performs activities are used. Corresponding feature are extracted from the video stream for fall detection.

Figures 1 show the 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.

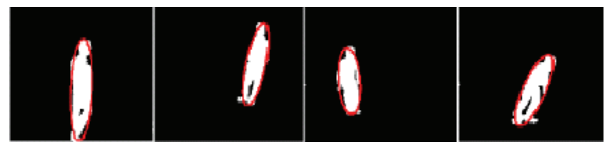

Fig. 1: Background subtraction results for fast walking, slow walking, sitting down and falling (from left to right), each frame includes a fitted red ellipse

Figure 2 show some tracking results of the 2-D head tracking. The head position is highlighted by a red ellipse.

Fig. 2: Some head tracking results, the current tracked head position is highlighted by a red ellipse

Figures 3 and 4 show the variations of head velocities, σ_θ during a video clip lasting approximate 22s, which contains a series of activities: fast walking, sitting down, standing up, slow walking and falling.

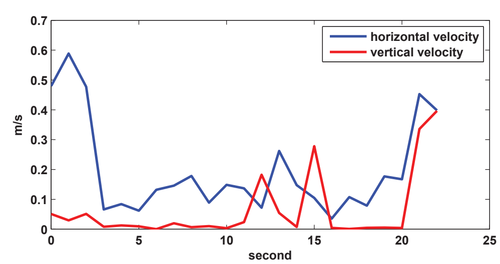

Fig. 3: Variations of velocities of head

Initially, we use a training dataset which contains 30 video feature corresponding to fall events (including fast falls and slow falls). And they are used to construct the fall density models. Here we use the data description toolbox [10], provided by D. Tax for density estimation. After the construction of density models, we test the performance of each model by using a test dataset which contains

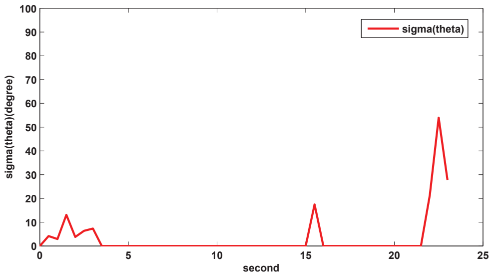

Fig. 4: Variations of σ_θ

30 fall activities and 30 non-fall activities (including walk, bend, crouch, lie, sit). The receiver operating characteristic (ROC) curves for different density models are plotted in 5.

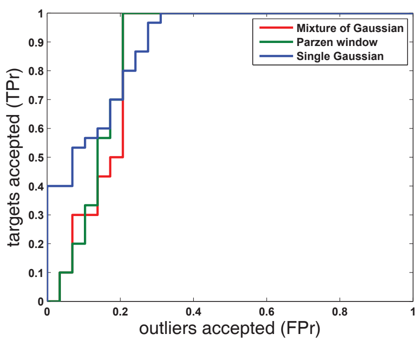

Fig. 5: The ROC curves for single Gaussian model, mixture of Gaussians model and Parzen window method

Figure 5 shows the ROC curves for single Gaussian model, mixture of Gaussians model and Parzen window method. From this figure, we can see that when true positive rate (TPr) is 100 %, the minimum false positive rate (FPr) of mixture of Gaussians and Parzen window is lower than that of single Gaussian model. For the comparison between mixture of Gaussians and Parzen window method, we can see that the the Area-under-the-ROC-curve (AUC) of Parzen window method is larger than that of mixture of Gaussian, according to [10], higher values indicate a better separation between target and outlier objects, so that Parzen window method is superior. However, Parzen window method is time consuming because every training data point is used for calculation.

Table 1 shows the performance of each density model for this test dataset under the corresponding optimal threshold (for which the geometric means, $\sqrt{TPr * (1 - FPr)}$ is maximized). From the table, we can see that the performance of mixture of Gaussians and Parzen window method is superior than that of single Gaussian model.

Table 1: The comparison between different fall detection methods.

Method	TPr (%)	FPr (%)	Geometric means
Single Gaussian model	93.3	27.6	0.67
Mixture of Gaussians	100	20	0.8
Parzen window	100	20	0.8

5. 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 applied to both video recordings recorded by the two cameras. Finally, the head's 3-D velocity and position information and the extracted human shape information σ_θ are used as feature for construct the corresponding fall density model to determine a falling event.

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, and another subset of one class classification technique, boundary method, will be used in future work for fall detection to cope with the high-dimensional feature situation.

6. ACKNOWLEDGEMENT

Here the author acknowledges the help of C. Rougier et al. and their papers [4, 5] which provide useful video feature for density estimation and D. Tax for the data description toolbox.

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 Proc. ACIVS 2008, Juan-les-Pins, France*.

[3] M. Yu, S. Naqvi, and J. Chambers, "Fall detection in the elderly by head tracking," *In Proc. IEEE SSP 2009, Cardiff, UK*.

[4] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Monocular 3d head tracking to detect falls of elderly people," *In Proc. IEEE EMBC 2006, New York USA*.

[5] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," *In Proc. AINAW 2007, Niagara Falls, Canada*.

[6] N. Bouaynaya, Q. Wei, and D. Schonfeld, "An online motion-based particle filter for head tracking applications," *In Proc. IEEE ICASSP 2005, Philadelphia, USA*.

[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 Proc. IEEE ICME 2005, Amsterdam, The Netherlands*.

[9] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, pp. 323–344, 1987.

[10] D. Tax, "The manual of data description toolbox 1.6.3," 2008.

[11] C. Bishop, "Pattern recognition and machine learning," *Springer*, 2006.