

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



CC creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

 **Attribution.** You must attribute the work in the manner specified by the author or licensor.

 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

Posture Recognition Based Fall Detection System For Monitoring An Elderly Person In A Smart Home Environment

Miao Yu, Adel Rhuma, Syed Mohsen Naqvi, Liang Wang and Jonathon Chambers

Abstract—We propose a novel computer vision based fall detection system for monitoring an elderly person in a home care application. Background subtraction is applied to extract the foreground human body and the result is improved by using certain post-processing. Information from ellipse fitting and a projection histogram along the axes of the ellipse are used as the features for distinguishing different postures of the human. These features are then fed into a directed acyclic graph support vector machine (DAGSVM) for posture classification, the result of which is then combined with derived floor information to detect a fall. From a dataset of 15 people, we show that our fall detection system can achieve a high fall detection rate (97.08%) and a very low false detection rate (0.8%) in a simulated home environment.

Index Terms— Health care, assistive living, fall detection, multi-class classification, DAGSVM, system integration

I. INTRODUCTION

In this section, we will briefly review the existing fall detection systems and describe our new computer vision based fall detection system.

A. Current fall detection techniques

Nowadays, the trend in western countries is for populations to contain an increasing number of elderly people. As shown in [1], the old-age dependency ratio (the number of people 65 and over relative to those between 15 and 64) in the European Union (EU) is projected to double to 54 percent by 2050, which means that the EU will move from having four persons of working age for every elderly citizen to only two. So, the topic of home care for elderly people is receiving more and more attention. Among such care, one important issue is to detect whether an elderly person has fallen or not [2]. According to [2], falls are the leading cause of death due to injury among the elderly population and 87% of all fractures in this group are caused by falls. Although many falls do not result in injuries, 47% of non-injured fallers can not get up without assistance and this period of time spent immobile also affects their health. An efficient fall detection system is essential for monitoring an elderly person and can even save his life in some cases. When an elderly person falls, a fall detection system will detect the anomalous behavior and an alarm signal will be sent to certain caregivers (such as hospitals or health centers) or the elderly person's family members by a modern communication method. Fig. 1 shows such a fall detection system.

Different methods have been proposed for detecting falls and are mainly divided into two categories: non-computer vision based methods and computer vision based methods.

Copyright (c) 2007 IEEE. Personal use of this material is permitted.

Miao Yu, Adel Rhuma, Syed Mohsen Naqvi and Jonathon Chambers are with the Advanced Signal Processing Group, School of Electronic, Electrical and Systems Engineering, Loughborough University, UK, e-mails: (m.yu, a.rhuma, s.m.r.naqvi, j.a.chambers)@lboro.ac.uk.

Liang Wang is with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China, e-mail: wangliang@nlpr.ia.ac.cn.

1) *Non-computer vision based methods:* There are many non-computer vision based methods for fall detection [3], [4], [5] and [6]. For these methods, different sensors (including acceleration sensors, acoustic sensors and floor vibration sensors) are used to capture the sound, vibration and human body movement information and such information is applied to determine a fall.

Veltink et al. [3] were the first to utilize a single axis acceleration sensor to distinguish dynamic and static activities. In their work, acceleration sensors were placed over the chest and at the feet to observe the changes, and a threshold based algorithm was applied on the measured signals for fall detection. Kangas et al. [4] proposed an improved scheme, they used a single three axis acceleration sensor to attach to the subject's body in different positions and the dynamic and static acceleration components measured from these acceleration sensors were compared with appropriate thresholds to determine a fall. Experimental results confirmed that a simple threshold based algorithm was appropriate for certain falls. Some researchers have also used acoustic sensors for fall detection. In [5], an acoustic fall detection system (FADE) that would automatically signal a fall to the monitoring care giver was designed. A circular microphone array was applied to capture and enhance sounds in a room for the classification of 'fall' or 'non-fall', and the height information of the sound source was used to reduce the false alarm rate. The authors evaluated the performance of FADE using simulated fall and nonfall sounds performed by three stunt actors trained to behave like elderly people under different environmental conditions and good performance was obtained (100% fall detection rate and 3% false detection rate using a dataset consisting of 120 falls and 120 nonfalls). Y. Zigel et al. in [6] proposed a fall detection system based on floor vibration and sound sensing. Temporal and spectral features were extracted from signals and a Bayes' classifier was applied to classify fall and nonfall activities. In their work, a doll which mimicked a human was used to simulate falls and their system detected such falls with a fall detection rate of 97.5% and a false detection rate of 1.4%.

Although non-computer vision based methods may appear to be suitable for wide application in the fall detection field, several problems do exist; they are either inconvenient (elderly people have to wear acceleration sensors) or easily affected by noise in the environment (acoustic sensors and floor vibration sensors). In order to overcome these problems, computer vision based fall detection techniques are adopted. Infringement of personal privacy is a concerning issue for computer vision based fall detection systems and elderly people may worry that they are being 'watched' by cameras. However, in most computer vision based fall detection systems, only the alarm signal (sometimes with a short video clip as further confirmation of whether an elderly person has fallen or not) will be sent to the caregivers or family members when a fall is detected; additionally, the original video recordings of an elderly person's normal activities will not be stored, nor transmitted.

2) *Computer vision based methods:* In the last 10 years, there have been many advances in computer vision and camera/video and image processing techniques that use real time movement of the subject, which opens up a new branch of methods for fall detection.

For computer vision based fall detection methods, some re-

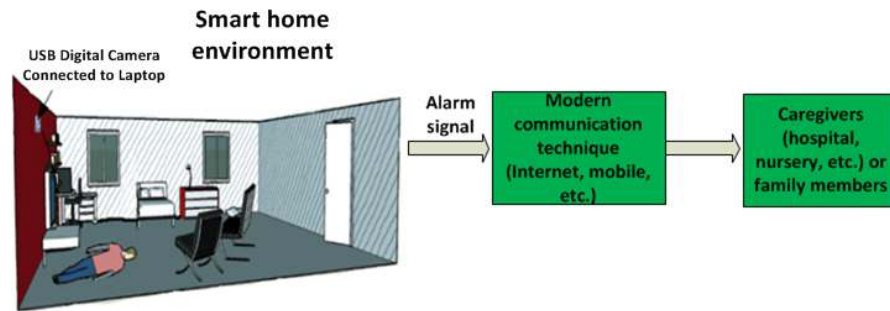


Fig. 1. Schematic representation of a fall detection system.

searchers have extracted information from the captured video and a simple threshold method has been applied to determine whether there is a fall or not; representative ones due to Rougier et al. are [7] and [8]. In these two papers, the head's velocity information and the shape change information were extracted and appropriate thresholds were set manually to differentiate fall and non-fall activities. However these two methods produce high false detection rates (such as when a fast sitting activity was misclassified as a fall activity [7]) and the performance was strongly related to the set threshold. Another threshold based method was proposed in [9] in which calibrated cameras were used to reconstruct the three-dimensional shape of people. Fall events were detected by analyzing the volume distribution along the vertical axis, and an alarm was triggered when the major part of this distribution was abnormally near the floor over a predefined period of time. The experimental results showed good performance of this system (achieving 99.7% fall detection rate or better with four cameras or more) and a graphic processing unit (GPU) was applied for efficient computation.

With the recent rapid development of pattern recognition techniques many researchers have exploited such methods in fall detection. Posture recognition based fall detection methods are proposed in [10], [11], [12] and [13]; in [10], the researchers used a neural fuzzy network for posture classification, and when the detected posture changed from 'stand' to 'lie' in a short time, a fall activity was detected. A similar idea was proposed in [11] except that the classifier was replaced with a more common k-nearest neighbour classifier; moreover, statistical hypothesis testing was applied to obtain the critical time difference to differentiate a fall incident event from a lying down event, and a correct detection rate of 84.44% was obtained according to their experimental results. In [12] and [13], Mihailidis et al. used a single camera to classify fall and non-fall activities. Carefully engineered features, such as silhouette features, lighting features and flow features were extracted to achieve robustness in the system to lighting, environment and the presence of multiple moving objects. In [13] three pattern recognition methods were compared (logistic regression, neural network and support vector machine) and the neural network achieved the best performance with a fall detection rate of 92% and a false detection rate of 5%.

Some other researchers classified fall and non-fall activities based on the features extracted from short video clips. The representative papers are [14] and [15]. For [14], a bounding box and motion information were extracted from consecutive silhouettes as features. These were then used to train a hidden Markov model (HMM) for classifying fall and non-fall activities. In [15], a person's three-dimensional orientation information was extracted from multiple uncalibrated cameras, and an improved version of HMM-layered hidden Markov model (LHMM) was used for fall detection. Although theoretically elegant, insufficient experimental results were provided in this paper (it only concerned two kinds of activities – walking and falling).

There are also some other computer vision based methods for

fall detection. Nait-Charif and McKenna [16] proposed a method for automatically extracting motion trajectory and providing a human readable summary of activity and detection of unusual inactivity in a smart home. A fall was detected as a deviation from usual activity according to the particle filter-based tracking results. This method exploited an unsupervised approach to detect abnormal events (mainly falls) and as it is common with unsupervised methods has the disadvantage that a long training period was required. In [17], D. Anderson proposed a fuzzy logic based linguistic summarization of video for fall detection. A hierarchy of fuzzy logic was used, where the output from each level was summarized and fed into the next level for inference. Corresponding fuzzy rules were designed under the supervision of nurses to ensure that they reflect the manner in which elders perform their activities. This system was tested on a dataset which contained 14 fall activities and 32 non-fall activities; all the fall activities were correctly detected and only two non-fall activities were mistaken as fall activities, which shows an acceptable level of performance.

In this paper, we propose a new computer vision fall detection system which is based on posture recognition using a single camera to monitor an elderly person who lives alone at home. An efficient codebook background subtraction algorithm is applied to extract the human body foreground and some post-processing is applied to improve the results. From the extracted foreground silhouette, we extract features from the fitted ellipse and projection histogram, which are used for classification purposes. These features are fed into the DAGSVM (which is trained from a dataset containing features extracted from different postures in different orientations) and the extracted foreground silhouette is classified as one of four different postures (bend, lie, sit and stand). The classification results, together with the detected floor information, are then used to determine fall or non-fall activities. The flow chart of the proposed fall detection system is shown in Fig. 2. In the next sections, we will describe different blocks of this flow chart in detail.

II. METHODS

A. Human body extraction

1) *Background subtraction*: In visual surveillance, a common approach for discriminating moving objects from the background is detection by background subtraction. Currently, there are many background subtraction algorithms, these include the single-mode model background subtraction method [18] and [19], the mixture of Gaussians (MoG) background subtraction method [20], the non-parametric density estimation based method [21] and the codebook background subtraction method [22]. In this fall detection system, we use the codebook method because of its advantages. There is no parametric assumption on the codebook model and it shows the following merits as proposed in [22]: (1) resistance to artifacts of acquisition, digitization and compression, (2) capability of coping with illumination changes, (3) adaptive and compressed background

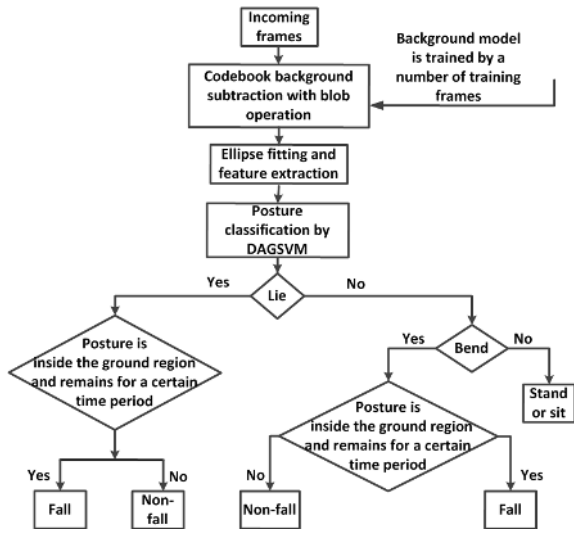


Fig. 2. The flow chart of the proposed fall detection system

models that can capture structural background motion over a long period of time under limited memory, (4) unconstrained training that allows moving foreground objects in the scene during the initial training period.

The codebook method is available for both colour and grayscale images, it is a pixel-based approach and initially a codebook is constructed for each pixel during a training phase. Assuming the training dataset \mathbf{I} contains a number of N images: $\mathbf{I} = \{imag_1, \dots, imag_N\}$, then, for a single pixel (x, y) , it has N training samples $imag(x, y)_1, \dots, imag(x, y)_N$. From these N training samples, a codebook is constructed for this pixel, which includes a certain number of codewords. Each codeword, denoted by \mathbf{c} , consists of an RGB vector $\mathbf{v} = (\overline{R}, \overline{G}, \overline{B})$ and a 6-tuple $\mathbf{aux} = (\hat{I}, \check{I}, f, \lambda, p, q)$. The meanings of the six parameters in \mathbf{aux} are described as follows:

\hat{I}	Maximum intensity that has been represented by the codeword.
\check{I}	Minimum intensity that has been represented by the codeword.
f	Number of times that the codeword has been used.
λ	Maximum negative runtime length (MNRL) in number of frames.
p	The first frame in which this codeword was used.
q	The last frame in which this codeword was used.

The details of the training procedure are given in [22] and the trained codebooks of pixels are then used for background subtraction purpose. For an incoming colour frame \mathbf{f} , its pixel $\mathbf{f}(x, y) = (R(x, y), G(x, y), B(x, y))$ (a 3-dimensional vector) is determined as a foreground or background pixel by comparing $\mathbf{f}(x, y)$ with codewords in the codebook of this pixel. If $\mathbf{f}(x, y)$ is not matched with any codeword, then it is a foreground pixel. For a particular codeword \mathbf{c} , we say the codeword \mathbf{c} matches $\mathbf{f}(x, y)$ if the following two conditions are met.

$$\begin{aligned} \text{colordist}(\mathbf{f}(x, y), \mathbf{c}) &\leq \varepsilon \\ \text{brightness}(I, \langle \hat{I}, \check{I} \rangle) &= \text{true} \end{aligned} \quad (1)$$

where ε is a preset threshold value for comparison, I represents the norm of $\mathbf{f}(x, y)$, \hat{I} and \check{I} are the first two parameters of the 6-tuple \mathbf{aux} vector of the codeword \mathbf{c} .

The $\text{colordist}(\mathbf{f}(x, y), \mathbf{c})$ measures the chromatic difference between two colour vectors, which can be calculated by:

$$\text{colordist}(\mathbf{f}(x, y), \mathbf{c}) = \sqrt{\|\mathbf{f}(x, y)\|^2 - \frac{\langle \mathbf{f}(x, y), \mathbf{v} \rangle^2}{\|\mathbf{v}\|^2}} \quad (2)$$

where \mathbf{v} represents the RGB vector $\mathbf{v} = (\overline{R}, \overline{G}, \overline{B})$ of codeword \mathbf{c} , and $\|\cdot\|$ and $\langle \cdot \rangle$ denote respectively the Euclidean norm and dot product operations.

The $\text{brightness}(I, \langle \hat{I}, \check{I} \rangle)$ is defined as:

$$\text{brightness}(I, \langle \hat{I}, \check{I} \rangle) = \begin{cases} \text{true} & \text{if } I_{low} \leq \|\mathbf{f}(x, y)\| \leq I_{hi} \\ \text{false} & \text{otherwise} \end{cases} \quad (3)$$

where $I_{low} = \alpha \hat{I}$ and $I_{hi} = \min\{\beta \hat{I}, \frac{\check{I}}{\alpha}\}$. In our experiment, α and β are fixed to be 0.5 and 2 respectively for background subtraction.

An important problem in background subtraction is background model updating, because the background will not be kept constant (such as with gradual light change, or movement of the furniture). The codebook background subtraction method therefore provides a background model updating scheme. The matched codeword according to (1) is updated as shown in [22]. Moreover, an additional cache model is introduced. If one codeword in this model is matched with the incoming pixel values for a period longer than a time threshold (which means this codeword is a new background codeword), it is added to the original codebook. And for a codeword which is not matched with incoming pixels longer than a time threshold (which means this codeword is no longer a background codeword), it is deleted from the codebook. Through the background model updating scheme, we can cope with change of the background in an indoor environment.

2) *Post-processing*: The result of the codebook background subtraction is definitely not perfect and needs to be improved to obtain a more accurate result to define the human's silhouette. As shown in one example in Fig. 3 (d) (the original background subtraction result), we can see two types of problems: 1) There are many noise-like pixel regions (very small areas which have sizes of less than 50 pixels, marked in blue); 2) Occasional movement of furniture, may produce ghost foreground regions (marked in yellow in Fig. 3 (d)) and the furniture at a new position can also be taken as the foreground (marked green in Fig. 3 (d)). These two problems will definitely deteriorate the result of the human body extraction. In order to solve these problems, certain post-processing is applied.

As proposed in [23], the connected foreground pixels form a region termed as a blob. By using the OpenCV blob library [24], we obtain blobs in a binary image format and small blobs with a size smaller than 50 pixels are removed. In this way, noise can be removed.

The background updating scheme can cope to some extent with the large ghosting errors caused by movement of furniture, and furniture appearing at a new position through absorption into the background model [22]. However, there are two problems if we rely solely on the background updating scheme: 1) It will take a time for ghosting and furniture to be absorbed into the background model by background updating; 2) The background updating scheme will wrongly absorb a foreground human body into the background model if he/she is static for a long time. In order to solve these two problems, we use a novel three step blob operation strategy as follows:

Step1. Blob merging: If the distance between two blobs is less than a threshold, these two blobs will be merged (as shown in Fig. 3 (d), the blobs B2 and B3 contain several separate blobs which are near to each other). The distance between two blobs is defined as the minimum 4-distance [23] between two rectangles which enclose the blobs as given by:

$$\text{Distance}(B1, B2) = \min_{p1 \in R1, p2 \in R2} d_4(p1, p2) \quad (4)$$

where $B1$ and $B2$ are two blobs, $R1$ and $R2$ are two rectangles which enclose them, and $p1$ and $p2$ are points belonging to $R1$ and $R2$. Fig. 4 shows examples of the distance between two blobs with respect to their positions.

Step2. Active blob determination: If the number of blobs after blob merging is more than one, it suggests some furniture has been moved (and we assume that the elderly person lives alone so that normally there should be only one human moving object). In this case, we determine which blob is the moving blob by using the frame

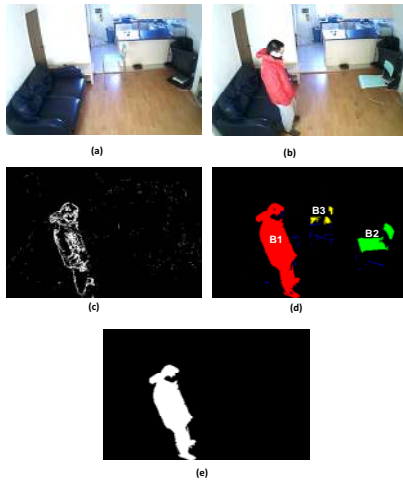


Fig. 3. The background subtraction and the human body blob determination. a) Background image; b) Image with human object; c) Frame difference result obtained from two consecutive frames; d) Original background subtraction result, there are three large blobs (B1, B2 and B3) after the blob merging operation and they are marked red, green and yellow, and the blue colour represents the small noise-like blobs; e) The final obtained human body blob.

difference technique [23]. Frame differencing is applied between consecutive frames to obtain the moving pixels (shown in Fig.3 (c)), and the blob with the greatest number of moving pixels is taken as the moving blob (human body blob). From Fig.3, we can see that the blob B1 contains the most moving pixels and so B1 is finally taken as the human body blob.

Step3. Selective updating: The non-active blobs are removed (as shown in Fig.3 (e), B2 and B3 are removed from the final background subtraction result) and their pixel values form new codewords to be added to the background codebook immediately for background model updating. And no updating is performed for pixels in the active blob.

In this way, ghosting and furniture at a new position are absorbed into the background model immediately; while the foreground human body object is not absorbed into the background model even though he/she is static for a long time.

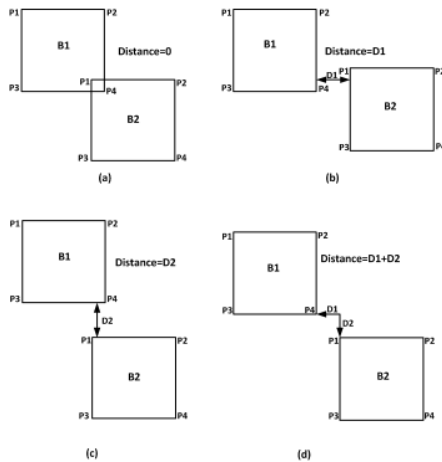


Fig. 4. Four cases of the distance between two blobs with respect to their relative positions

3) *Background model retraining*: The trained background codebook model can be affected in various ways, such as dramatic global illumination change due to suddenly turning on the light. In

this situation, the codebook needs to be re-trained because the previous codebook is no longer available. The dramatic global illumination change can be detected by frame differencing results, if the percent of the active pixels in an image is larger than a threshold (we set 50%), then we assume that dramatic global illumination change has occurred and the background model is retrained.

Next, having extracted a silhouette representation of the human, we consider feature extraction to describe the posture of the person.

B. Feature extraction

After human body region extraction, the next step is to extract useful features from the human body region. For feature extraction, we extract two kinds of features: global features (which roughly describe the shape of the human body) and local features (which encapsulate the detail information of the posture of the human body).

To obtain global features, we use ellipse fitting [25] for a binary image. The moments for a binary image $f(x, y)$ are given as:

$$m_{pq} = \sum_{x,y} x^p y^q f(x, y) \quad \text{with } p, q = 0, 1, 2, 3, \dots \quad (5)$$

By using the first and zero order spatial moments, we can compute the center of the ellipse as: $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$. The angle between the major axis of the person and the horizontal axis x gives the orientation of the ellipse, and it is computed as:

$$\Theta = \frac{1}{2} \arctan\left(\frac{2u_{11}}{u_{20} - u_{02}}\right) \quad (6)$$

where the central moment can be calculated as:

$$u_{pq} = \sum_{x,y} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad \text{with } p, q = 0, 1, 2, 3, \dots \quad (7)$$

The major semi-axis a and the minor semi-axis b can be obtained by calculating the greatest and least moments of inertia, here we denote them as I_{max} and I_{min} . They can be calculated by evaluating the eigenvalues of the covariance matrix:

$$J = \begin{pmatrix} u_{20} & u_{11} \\ u_{11} & u_{02} \end{pmatrix} \quad (8)$$

These are calculated as:

$$I_{max} = \frac{u_{20} + u_{02} + \sqrt{(u_{20} - u_{02})^2 + 4u_{11}^2}}{2} \quad (9)$$

$$I_{min} = \frac{u_{20} + u_{02} - \sqrt{(u_{20} - u_{02})^2 + 4u_{11}^2}}{2} \quad (10)$$

Finally, according to [8], we can calculate the major semi-axis a and minor semi-axis b as:

$$a = (4/\pi)^{1/4} \left[\frac{I_{max}}{I_{min}} \right]^{1/8} \quad (11)$$

$$b = (4/\pi)^{1/4} \left[\frac{I_{min}}{I_{max}} \right]^{1/8} \quad (12)$$

An ellipse fitting result is depicted in Fig.5, and we compare the ellipse fitting result and the rectangle fitting result used in [10]. The ellipse fitting is clearly better in describing the human posture in the presence of noise (such as the line underneath a person's feet due to the poor segmentation, as shown in Fig.5). After ellipse fitting, the orientation of the ellipse and the ratio between a and b are taken as global features, which have been found experimentally to be sufficient to describe the posture of a human body.

Such global features are, however, insufficient to describe the postures in detail, and sometimes it is hard to differentiate two postures by using only the global information (such as a sit posture and a sit-like bend posture). We need to use more information (local



Fig. 5. A comparison of rectangular fitting (a) and ellipse fitting (b). A line underneath the human's feet is obtained due to the poor segmentation.

features) to describe the postures for better differentiating purpose. A widely used feature to describe such detail information is the projection histogram, as mentioned in [10], [11] and [26]. This feature is computationally efficient so that it can be applied in a real time application, while achieving a good performance for posture classification. In this work, we also adopt the projection histogram; and because we use ellipse fitting, the projection directions are along the long and short axes of the ellipse. One example is as in Fig. 6, the projection histograms of the long and short axes of the fitted ellipses are obtained. From the results, we can observe the differences in the patterns within the histograms between different postures, which are helpful for posture classification.

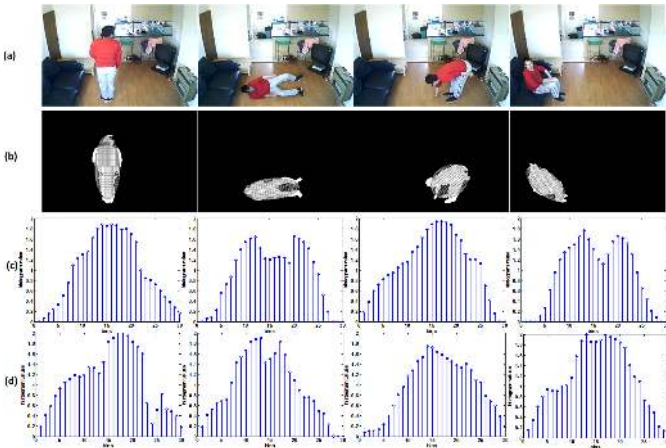


Fig. 6. Projection histograms of four different types of postures. (a) original frames (b) Background subtraction results with fitted ellipses and projection lines (c) Projection histograms along the long axis of the ellipse (d). Projection histograms along the short axis of the ellipse. The horizontal axis of the projection histogram represents the index of bins and the vertical axis represents the value of the projection histogram.

In this work, the numbers of bins of the long axis projection and short axis projection histograms are all 30, a value found empirically to provide suitable detail whilst not introducing undue complexity. For particular bins of the projection histograms along the ellipse's major and minor axes, their values are calculated as in (13) and (14).

$$bin_{major}(i) = \frac{NoFP_{major}(i)}{L_{major}} \quad (13)$$

$$bin_{minor}(i) = \frac{NoFP_{minor}(i)}{L_{minor}} \quad (14)$$

where i is the index of bins, $NoFP_{major}(i)$ and $NoFP_{minor}(i)$ denote the number of foreground pixels along the i th-projection line in the directions of major and minor axes respectively. The results are normalized by L_{major} and L_{minor} , which represent the lengths of the major and minor axes. The purpose of the normalization is to make sure this feature is invariant to the size of the foreground human body region (which will be different according to a person's

distance to the camera).

In this way, we obtain the 'local features' which describe the detail information of the postures. The global features and the local features are combined as a whole feature vector for classification. We calculate the two values as the global feature and 60 values for detail description, so the final feature has a dimension of 62. The combination of these two types of features will be shown experimentally in Section III to yield better classification results than using global or local features alone.

C. Multi-class support vector machine classification by DAGSVM

After obtaining features, an appropriate classifier is needed to classify features into one of four posture classes and in this work we propose the use of a DAGSVM for classification purpose. DAGSVM is a scheme to combine multiple support vector machines (SVMs) [27], which are designed to solve two-class classification problems, to achieve multi-class classification. As pointed out in [28], the DAGSVM scheme has been proven to have a theoretically defined generalization error bound and it is more efficient than other multi-class SVM schemes with respect to the training and computation time.

The structure of the DAGSVM is shown in Fig. 7 for a four class classification problem. We can see that the DAGSVM has a tree-like structure and each node in this tree-like structure corresponds to a two-class SVM. The decision process just follows the structure and is based on a sequence of two-class operations, a decision is made when a bottom node is reached.

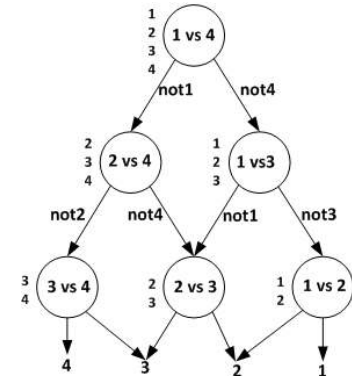


Fig. 7. The decision process for the traditional DAGSVM for a four class problem [28]

As for all other classifiers, an indispensable step for training the DAGSVM classifier is to determine optimal parameters so that the classifier can achieve the optimal performance. There are two sets of parameters which need to be tuned for the DAGSVM classifier: 1) The structure for the DAGSVM; an optimal structure is needed for the DAGSVM to guarantee good performance. 2) The parameters of each two-class SVM node, including the Gaussian kernel parameters (Gaussian kernel is used here because it can map the original data into an infinite dimensional feature space thus can increase the separability of the two classes) and the penalty parameter in the two-class SVM scheme [27]. In this work, we use the method in [29] to tune these parameters, which is based on the distance between two classes (DBTC) as proposed in [30] for efficient parameter tuning of a two-class support vector machine.

D. Rules for fall detection

After obtaining the posture class, we determine a fall or non-fall activity if the following three conditions are met:

- 1) The posture is classified as 'lie' or 'bend'.

- 2) The posture is inside the ground region.
- 3) The above two conditions are kept for a certain time, which exceeds a preset time threshold (we use 30s).

The three conditions are set according to the characteristic of fall activity, in most cases, fall activities end up with a ‘lie’ posture and this ‘lie’ posture usually remains for a certain time due to the period of immobility of an elderly person after the fall. Besides, different from lying on the bed/sofa, the posture should be inside the ground region (or at least a large part is inside the ground region). And considering that an elderly person rarely ‘bends’ for a long time in the ground region (here we define the ‘bend’ class as postures of bending to fasten a shoe lace or bending to pick up something, which is very common to occur in an elderly person’s daily life), if a ‘bend’ posture is detected in the ground region for a certain time, we also regard it as an abnormal activity (this can happen when an elderly person falls while ending up with a bend-like posture, an example will be given in Section III).

From the above three conditions, we can see that in order to detect falls, the ground region needs to be determined initially. Before the fall detection phase, floor detection is carried out. In this phase, when the posture is classified as stand or sit, the region nearby the lower extreme point of the ellipse (here we use an 8×8 block) is marked as the ground region. Fig. 8 shows the result of floor detection.

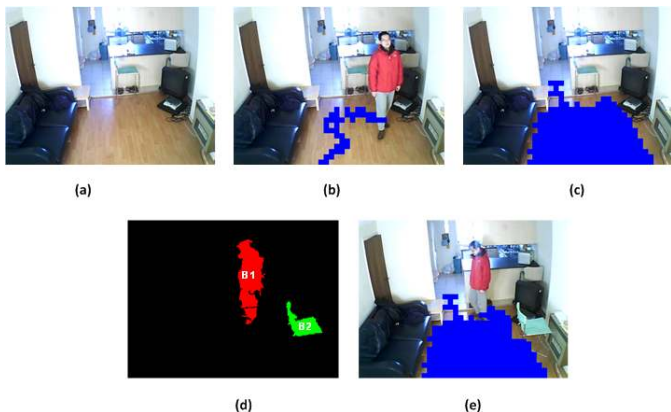


Fig. 8. The floor detection results. a) Original image; b) Detected floor region while a person is walking; c) Floor detection result after some time; d) More than one blob after the furniture is moved, the moving blob (human body) is marked in red, the static blob (furniture) is marked green; e) The updated floor region result after moving furniture. The region nearby the new position of the furniture is unmarked and that nearby the person’s feet is marked as the floor region.

Note, as shown in (d) and (e), sometimes the furniture is moved and the floor region has to be updated accordingly. The detected floor region is extremely helpful to distinguish a fall on the floor from lying on a sofa, which will be shown in the experimental part.

III. RESULTS

In this part, we show the performance of our fall detection system. The experiments were carried out in a simulated home environment. A USB camera was used for recording the real video sequence with the image size of 320×240 , the recorded video sequence is processed by using VC++ 6.0 (with OpenCv library 1.0) and Matlab on an Intel(R) Core(TM)2 Duo CPU laptop with 1.00GB Memory. Fig. 9 shows the camera we used in the experiment and the background image it records. 15 people (11 males and 4 females) were invited to attend the experiments for simulating different postures and activities. The characteristics of the 15 people are summarized in Table I. We have to note that real elderly people were not invited to participate in the experiments because it is unsafe for an elderly person to simulate fall activities; instead, younger people were invited to mimic elderly people, and they all gave their consent before commencing the study.



Fig. 9. The USB camera used in the experimental room environment (a) and the experimental environment (b)

TABLE I
THE CHARACTERISTICS OF 15 PARTICIPATORS IN OUR EXPERIMENTS

No. of people	15
Male/Female	11/4
Age	25-49
Weight	57-94 (kg)
Height	158-187 (cm)

A. Background subtraction results

Some background subtraction results are shown in Fig. 10, Fig. 11 and Fig. 12, in which we analyze three challenging scenarios which occur commonly in a home environment. Initially, a short video clip which contains 100 frames is applied for training the original background model, which will be updated with the evolution of time.

Fig. 10 shows the background subtraction results at different times of day with gradual light change. Images (a) and (c) show a frame captured at noon time and the corresponding background subtraction result. The background model is updated to cope with the gradual light change and the results are shown in (b) and (d), where (b) is a frame captured in the afternoon and (d) is the background subtraction result with the updated background model.

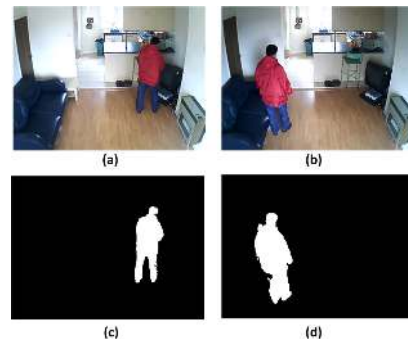


Fig. 10. Background subtraction results at different times of a day; (a) and (c) show an original frame captured at noon time and the corresponding background subtraction result; (b) is a frame captured in the afternoon with the light condition changed and (d) is the background subtraction result of (b) with the updated background model.

Fig. 11 shows the background subtraction results of moving objects. Line (a) shows four frames sampled in a short video sequence, which shows that a person moves the table and fruit plate. Line (b) shows the background subtraction results by directly applying the codebook background subtraction method. We can see there are many segmentation errors due to the movement of the table and fruit plate. Frame differencing results are shown in line (c), which indicate active pixels and help to find the active blob representing the human body. By the post-processing technique as discussed in Section II.A, improved background subtraction results are obtained in line (d).

Fig. 12 shows a case of sudden illumination change. At frame (c), the light is turned on and we can observe a large illumination



Fig. 11. Background subtraction results for moving furniture. Line (a) shows original frames of a person moving the table and fruit plate. Codebook background subtraction results are shown in line (b). Line (c) shows the frame differencing results which indicate active pixels. From the frame differencing results and blob operations, improved background subtraction results are obtained in line (d).

change. This sudden change of illumination can be detected by the frame differencing result as shown in (g), with more than 50% of pixels being marked as active ones (white). The background model is then retrained to cope with this sudden illumination change. As shown in (d) and (h), good segmentation is obtained by the retrained background model.

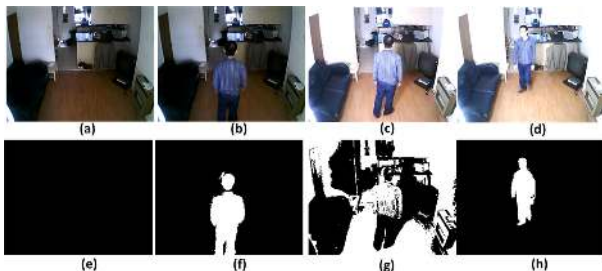


Fig. 12. Background subtraction for sudden illumination change. Frames (a) and (b) are captured with the light off, at frame (c) the light is turned on and we can observe drastic illumination change. And frame (d) is captured after the light is turned on for a certain time. Frames (e) and (f) are the background subtraction results of (a) and (b). Image (g) is the frame difference result for (c), sudden illumination is detected because more than 50% of the pixels are marked as active ones and the background model is retrained. Frame (h) is the subtraction result of (d) by the retrained background model.

B. Posture classification results

To form our posture dataset, 3200 postures (including 800 stands, 800 sits, 800 lies and 800 bends) from 15 people were recorded. As in [18], each person was asked to simulate postures in different directions so that the constructed classifier is robust to view angles. Some samples are shown in Fig. 13.

Based on this recorded dataset, a commonly used 10-fold validation [31] was applied to evaluate the performance of the posture classification and three types of comparisons were made:

The first evaluation is the feature comparison, which is shown in Table II; the classification results are compared when using the global feature or local feature alone and using a combination of these two features. DAGSVM is applied for classification and from this table we can see that the classification result by the combined feature is clearly better than using either feature alone.

TABLE II
CLASSIFICATION RESULT BY DIFFERENT TYPES OF FEATURES.

	Global features	Local features	Combined features
Classification rate	89.72%	76.70%	96.09%

In the second assessment, the performance of DAGSVM is compared with other traditional classifiers (k-nearest neighbor (KNN), multi-layer perceptron neural network (MLPNN), naive Bayes classifier (NBC) and binary decision tree (BDT) implemented in PRtools—a well-known software package for pattern recognition [32]), the results of which are shown in Table III. For a fair comparison, the parameters of the comparison classifiers are tuned to be optimal by using the cross validation function in PRtools.

TABLE III
COMPARISON OF DIFFERENT CLASSIFIERS.

	DAGSVM	KNN	MLPNN	NBC	BDT
Classification rate	96.09%	92.64%	92.53%	75.27%	83.72%

We can see that DAGSVM achieves the best performance with a slightly better classification rate performance than k-nearest neighbor and MLP neural network.

However, in the real situation, the training dataset is not perfect and it is common that some outliers exist in the training dataset. In posture classification, outliers are mainly caused by extremely bad background subtraction results and wrong labeling (the feature of one class is mislabeled as another class). In Table IV, the comparison results of the classifiers are presented on a dataset which includes 10% outliers.

TABLE IV
COMPARISON OF DIFFERENT CLASSIFIERS ON A DATASET WHICH IS CORRUPTED BY 10% OUTLIERS.

	DAGSVM	KNN	MLPNN	NBC	BDT
Classification rate	95.51%	84.07%	85.93%	72.42%	58.72%

From Table IV, we can see a more obvious advantage of the classification rate of the DAGSVM over other classifiers on this noisy dataset. And compared with other classifiers, DAGSVM is the most robust to such noise (with only a 0.58% drop in classification rate compared with the result of Table III). So, according to Table III and Table IV, DAGSVM is better for posture classification than other traditional classifiers.

C. Fall detection

Posture classification results along with the detected floor information are used to detect falls according to the three conditions in Section II.D. To illustrate, we show five cases presented in Fig. 14, (a) shows a person who has fallen on the floor, and a ‘lie’ posture is detected and most parts of the human body region are in the detected ground region; and this posture is kept for a certain time (longer than 30s), so a fall is detected. For (b), although a ‘lie’ posture is detected, the human body blob is not in the floor region, so the lying on the sofa case is correctly classified as non-fall. For (c), (d) and (e), the postures are all classified as ‘bend’; however, for (c), we can see that a large portion of the human body is in the ground region and the ‘bend’ posture remains longer than 30s. It is generally impossible for an elderly person to bend for a long time in the ground region, so this is abnormal activity and it is detected as a fall. For (d) and (e), either the detected ‘bend’ posture does not hold for a long time (for case (d), a person ties his shoe lace and the ‘bend’ posture recovers to ‘stand’ posture in a short time), or the posture is not in the ground region (only a small portion of the human body region is in the ground), so they are not detected as falls.



Fig. 13. Posture samples simulated by different participates in different orientations: (a) stand (b) sit (c) lie and (d) bend.

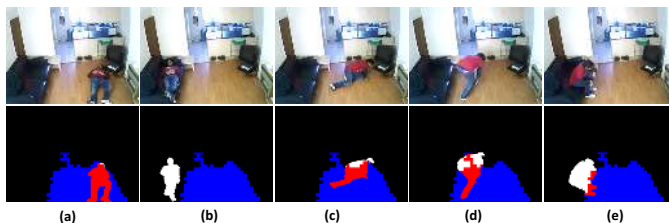


Fig. 14. Different cases of fall and non-fall activities. a) Fall on the floor; b) Lie on the sofa; c) Fall on the floor; d) Bend to fasten the shoe tie; e) Sit on the sofa. For (a) and (b), the postures are classified as ‘lie’ and for (c), (d) and (e), the postures are classified as ‘bend’. The blue region represents the detected floor, the red region represents the intersected part of the foreground human body with the detected floor region, the white region represents the foreground human body part which is not intersected with the detected floor region and the remaining background region is marked as black. Our proposed system successfully classifies (a), (c) as falls and (b), (d) and (e) as non-falls.

For evaluating this fall detection system, each person is asked to simulate 16 fall activities and 16 non-fall activities in different directions. A total number of 240 fall activities and 240 non-fall activities are recorded, which are used to evaluate the proposed fall detection system. For classifying one person’s activity, the postures from other people in the recorded posture dataset are used to construct the DAGSVM classifier. Final results are given in Table V, which show that 233 out of 240 (97.08%) falls can be detected while only 2 out of 240 (0.8%) non-falls mistaken as falls; and a high fall detection rate is obtained with a very low false detection rate.

IV. DISCUSSION

The proposed fall detection system is a posture classification based fall detection system, so it has the advantages of the traditional

TABLE V
EVALUATION OF THE PROPOSED FALL DETECTION SYSTEM.

	No. of activities	Detected as falls	Detected as non-falls
Falls	240	233	7
Walking around	60	0	60
Sitting on the sofa/chair	60	2	58
Bending	60	0	60
Lying on the sofa	60	0	60

posture classification based system: (1) It is convenient (elderly people need not wear a sensor) and it is not affected by background noise in the environment (suffered by acoustic sensors and floor vibration sensors methods). (2) If the training dataset is large enough, the well-trained classifier can effectively distinguish different types of postures, which are used for fall detection. And this procedure (including parameter optimization and classifier training) is totally automatic and there is no need for us to set some thresholds like the threshold-based methods in [7], [8] and [9]. (3) Because the posture classification based method is performed at the frame level, there is no need for a video segment which is necessary in the video clip based fall detection methods such as [14] and [15]. And compared with the video clip based method, it is computationally efficient (because it only extracts features from a single frame rather than from a whole video clip).

Compared with the traditional posture classification based fall detection systems [10], [11], [12] and [13], our proposed system has the following characteristics, which are the novelties of our work:

a) An effective background subtraction technique and a background model updating scheme are used to extract the human body and cope with the background change (such as gradual light change); and a novel three step blob operation strategy is applied, which selectively absorbs any moved furniture and ghosting into the

background model immediately.

b) Instead of rectangle fitting used in [10] and [11], a more noise-robust ellipse fitting is applied to extract the human body information in a better way (as shown in Fig. 5).

c) DAGSVM is applied for posture classification. DAGSVM shows an advantage in posture classification over other traditional classifiers as presented in Table III and the advantage is more obvious in a noisy dataset in Table IV.

d) Detected floor information is used to distinguish falls and lying on bed/sofa. In [11], distinguishing these two activities is obtained by comparing the transition time between ‘lie’ with other postures; however, under this method, some fast lying is mistaken as a fall (a false detection of 15.56% in [11]), which can be corrected by using the detected floor information as proposed in our method. From Table V, we can see that all the lying on sofa activities are correctly defined as non-fall activities.

However, some of the problems in [10], [11], [12] and [13] still occur in our system. One important problem is that the proposed system is designed for monitoring a single person (detecting falls of an elderly person living alone at home), it is not adequate for some special cases, such as multiple people in the home environment and when an elderly person has a large size pet (for a small size pet, it can be removed from the foreground region by using a blob size threshold). If there are multiple people in the home environment, however, there is no need for the fall detection system to operate (if the elderly person falls, other people will ask for help). In this case, either the fall detection system is turned off manually or automatically with the aid of some people counting technique as shown in [33] and [34] (if more than one people are detected by the people counting algorithm, the fall detection system is turned off automatically in the code). For the case of an elderly person with a large size pet, only the extracted foreground region corresponding to the human body silhouette is used for fall detection, whether the extracted foreground region is a human body silhouette or pet silhouette can be determined by some object classification technique, such as [35] and [36].

Another problem is occlusion in the indoor environment. The room setting in a real indoor environment is sometimes very cluttered, which will cause occlusions. And sometimes the occlusion is very serious (such as a person is totally behind the furniture), which will definitely deteriorate the performance of the fall detection system. In order to solve this problem, more than one camera can be used to make sure that the human body is visible in at least one camera’s view plane. For each camera, the proposed fall detection method is performed and results are combined in a proper way (such as majority voting) to make a decision.

V. CONCLUSION

In this work, we proposed a new computer vision based fall detection system based on posture recognition. A codebook background subtraction algorithm was adopted to extract the foreground region, the background was updated to cope with the change of the background model and some post-processing was used to improve background subtraction results. Ellipse fitting and projection histogram were used as the global and local features to describe postures. And the DAGSVM scheme was applied to obtain better multi-class classification results on the recorded posture dataset. Fall activities were detected if three conditions set according to fall characteristics were met. From the experimental results, we can see that our fall detection system achieves a good performance with a fall detection rate (97.08%) and a very low false detection rate (0.8%) on a 15 person dataset. However, as we have discussed, multiple moving objects and occlusions are two problems needed to be solved for our fall detection system, which can be addressed by using a multiple cameras scheme with adding corresponding modules for people counting and object classification.

ACKNOWLEDGEMENTS

The authors thank the Associate Editor and the anonymous reviewers for improving the clarity of our paper, and the 15 people

who consented to be part of the experiments. This work is partially supported by the Engineering and Physical Sciences Research Council of the UK, and a grant of the Hundred Talents Program of the Chinese Academy of Science.

REFERENCES

- [1] G. Carone and D. Costello, “Can Europe afford to grow old?,” *International Monetary Fund Finance and Development magazine*, vol. 43, no. 3, 2006.
- [2] J. Halter, J. Ouslander, M. Tinetti, S. Studenski, K. High, S. Asthana, and W. Hazzard, “Hazzard’s geriatric medicine and gerontology,” *Sixth Edition, McGraw-Hill*, 2009.
- [3] P. Veltink, H. Bussmann, W. Vries, W. Martens, and R. Lummel, “Detection of static and dynamic activities using uniaxial accelerometers,” *IEEE Transactions on Rehabilitation Engineering*, vol. 4, no. 4, pp. 375–385, 1996.
- [4] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jamsa, “Comparison of low-complexity fall detection algorithms for body attached accelerometers,” *Gait and Posture*, vol. 28, no. 2, pp. 285–291, 2008.
- [5] Y. Li, K. Ho, and M. Popescu, “A microphone array system for automatic fall detection,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 2, pp. 1291–1301, 2012.
- [6] Y. Zigel, D. Litvak, and I. Gannot, “A method for automatic fall detection of elderly people using floor vibrations and sound proof of concept on human mimicking doll falls,” *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 12, pp. 2858–2867, 2009.
- [7] C. Rougier and J. Meunier, “3d head trajectory using a single camera,” *International Journal of Future Generation Communication and Networking, invited paper for the special issue on Image and Signal Processing*, vol. 3, no. 4, pp. 43–54, 2010.
- [8] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, “Fall detection from human shape and motion history using video surveillance,” *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW), Niagara Falls, Ont., Canada*, 2007.
- [9] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier, “Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 290–300, 2011.
- [10] C. Juang and C. Chang, “Human body posture classification by a neural fuzzy network and home care system application,” *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 37, no. 6, pp. 984–994, 2007.
- [11] C. Liu, C. Lee, and P. Lin, “A fall detection system using k-nearest neighbor classifier,” *Expert Systems with Applications*, vol. 37, no. 10, pp. 7174–7181, 2010.
- [12] 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, no. 4, pp. 194–198, 2005.
- [13] M. Belshaw, B. Taati, J. Snoek, and A. Mihailidis, “Towards a single sensor passive solution for automated fall detection,” *33rd Annual International Conference of the IEEE EMBS, Boston, Massachusetts USA*, 2011.
- [14] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He, “Recognizing falls from silhouettes,” *Proceedings of the 28th IEEE EMBS Annual International Conference, New York City, USA*, 2006.
- [15] N. Thome, S. Miguet, and S. Ambellouis, “A real-time, multiview fall detection system: A lhm-based approach,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1522–1532, 2008.
- [16] S. McKenna and H. Nait-Charif, “Summarising contextual activity and detecting unusual inactivity in a supportive home environment,” *Pattern Analysis and Applications Journal*, vol. 7, no. 4, pp. 386–401, 2004.
- [17] D. Anderson, R. Luke, J. Keller, M. Skubic, M. Rantz, and M. Aud, “Linguistic summarization of video for fall detection using voxel person and fuzzy logic,” *Computer Vision and Image Understanding*, vol. 113, no. 1, pp. 80–89, 2009.
- [18] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, “Pfnder: Real-time tracking of the human body,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.
- [19] T. Horprasert, D. Harwood, and L. Davis, “A statistical approach for real-time robust background subtraction and shadow detection,” *In Proc. IEEE ICCV*, vol. 99, pp. 1–19, 1999.
- [20] C. Stauffer and W. Grimson, “Adaptive background mixture models for real-time tracking,” *Int. Conf. Computer Vision and Pattern Recognition, Fort Collins, CO, USA*, 1999.

- [21] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," *European Conf. Computer Vision*, 2000.
- [22] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using code-book model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.
- [23] R. Gonzalez, "Digital image processing," *Third Edition, Pearson Education*, 2008.
- [24] "http://opencv.willowgarage.com/wiki/cvblobslib, accessed February 2012," accessed February 2012.
- [25] W. Pratt, "Digital image processing," *Third Edition, John Wiley and Sons*, 2001.
- [26] R. Cucchiara, A. Prati, and R. Vezzani, "Vezzani: A multi-camera vision system for fall detection and alarm generation," *Expert Systems Journal*, vol. 24, no. 5, pp. 334–345, 2007.
- [27] V. Vapnik, "Statistical learning theory," *Wiley, New York*, 1998.
- [28] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," *Advances in Neural Information Processing Systems, MIT Press*, pp. 547–553, 2000.
- [29] M. Yu, A. Rhuma, S. Mohsen, and J. Chambers, "An improved directed acyclic graph support vector machine," *Journal of Measurement Science and Instrument*, vol. 2, no. 4, pp. 367–360, 2011.
- [30] J. Sun, C. Zheng, X. Li, and Y. Zhou, "Analysis of the distance between two classes for tuning SVM hyperparameters," *IEEE Transactions on Neural Networks*, vol. 21, no. 2, pp. 305–318, 2010.
- [31] C. Bishop, "Pattern recognition and machine learning," in: *Springer*, 2006.
- [32] F. Heijden, R. Duin, D. Ridder, and D. Tax, "Classification, parameter estimation and state estimation: An engineering approach using matlab," *Wiley*, 2004.
- [33] A. Chan and N. Vasconcelos, "Counting people with low-level features and Bayesian regression," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2160–2177, 2012.
- [34] Y. Hou and G. Pang, "People counting and human detection in a challenging situation," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 41, no. 1, pp. 24–33, 2011.
- [35] Y. Chen, L. Zhu, A. Yuille, and H. Zhang, "Unsupervised learning of probabilistic object models (POMs) for object classification, segmentation, and recognition using knowledge propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1747–1761, 2009.
- [36] F. Lecumberry, A. Pardo, and G. Sapiro, "Simultaneous object classification and segmentation with high-order multiple shape models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 625–635, 2010.



Mr. Miao Yu was born in China in 1986. He obtained his BSc degree in 2003 in Electronic and Information Engineering from Shandong University of Science and Technology, China. He then obtained an MSc with Distinction in Digital Communication Systems in 2008, together with the best student award, from the Department of Electronic and Electrical Engineering Loughborough University, U.K., where he is currently studying for a PhD in the area of fall detection for the elderly by exploiting video information.



Mr. Adel Rhuma was born in Libya in 1971. He gained his BSc degree in 1998 in Computer Engineering from the Engineering Academy, Tajura-Libya. He next obtained an MSc in Remote Sensing Image Processing and Application, from the University of Dundee, UK. He is currently working towards a PhD in the area of fall detection for the elderly by exploiting video information.



Dr. Syed Mohsen Raza Naqvi (S'07-M'09) received his B.Eng. degree in Industrial Electronics Engineering from IIEE/NED University of Engineering and Technology, Karachi, Pakistan, in 2001 and his Ph.D. degree in Signal Processing from Loughborough University, Leicestershire, U.K., in 2009. Before his postgraduate studies in U.K., he worked in research and development in Pakistan from January 2002 to September 2005.

Dr Naqvi is currently working as a Research Associate within the School of Electronic, Electrical and Systems Engineering at Loughborough. His research interests include multimodal signal processing, audio-visual speech processing, nonlinear filtering, data fusion, anomaly detection, embedded system design, and renewable energy.



Prof. Liang Wang received both the B. Eng. and M. Eng. degrees from Anhui University in 1997 and 2000 respectively, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences (CAS) in 2004. From 2004 to 2010 he worked as a Research Assistant at Imperial College London, U.K., and at Monash University, Australia, as a Research Fellow at the University of Melbourne, Australia, and as a Lecturer at the University of Bath, U.K., respectively. Currently, he is a Professor of the Hundred Talents Program at the National Lab of

Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, P. R. China.

His major research interests include machine learning, pattern recognition and computer vision. He has published widely within highly-ranked international journals such as IEEE TPAMI and IEEE TIP, and leading international conferences such as CVPR, ICCV and ICDM. He has obtained several honors and awards such as the Special Prize of the Presidential Scholarship of the Chinese Academy of Sciences. He is a Senior Member of the IEEE, as well as a member of BMVA. He is an Associate Editor of IEEE Transactions on Systems, Man and Cybernetics - Part B, International Journal of Image and Graphics, Signal Processing, Neurocomputing and International Journal of Cognitive Biometrics. He is a guest editor of seven special issues, a co-editor of five edited books, and a co-chair of seven international workshops.



Prof. Jonathon A. Chambers (S'83-M'90-SM'98-F'11) received the Ph.D. degree in signal processing from the Imperial College of Science, Technology and Medicine (Imperial College London), London, U.K., in 1990. From 1991 to 1994, he was a Research Scientist with Schlumberger Cambridge Research Center, Cambridge, U.K. In 1994, he returned to Imperial College London, as a Lecturer in signal processing and was promoted as a Reader (Associate Professor) in 1998. From 2001 to 2004, he was the Director of the Centre for Digital Signal Processing

and a Professor of signal processing with the Division of Engineering, King's College London, London. From 2004 to 2007, he was a Cardiff Professorial Research Fellow with the School of Engineering, Cardiff University, Wales, U.K. In 2007, he joined the Department of Electronic and Electrical Engineering, Loughborough University, Loughborough, U.K., where he heads the Advanced Signal Processing Group and serves as the Associate Dean Research with the School of Electronic, Electrical and Systems Engineering.

He is a co-author of the books *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability* (Wiley, 2001) and *EEG Signal Processing* (Wiley, 2007). He has advised more than 50 researchers through to Ph.D. graduation and published more than 350 conference proceedings and journal articles, many of which are in IEEE journals. His research interests include adaptive and blind signal processing and their applications. Dr. Chambers is a Fellow of the Royal Academy of Engineering, U.K., and the Institution of Electrical Engineers (IEE). He was the Technical Program Chair of the 15th International Conference on Digital Signal Processing (DSP 2007) and the 2009 IEEE Workshop on Statistical Signal Processing, both held in Cardiff, U.K., and a Technical Program Cochair for the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011), Prague, Czech Republic. He is the recipient of the first QinetiQ Visiting Fellowship in 2007 "for his outstanding contributions to adaptive signal processing and his contributions to QinetiQ" as a result of his successful industrial collaboration with the international defense systems company QinetiQ.

Prof. Chambers has served on the IEEE Signal Processing Theory and Methods Technical Committee for six years and is currently a member of the IEEE Signal Processing Society Awards Board and the European Signal Processing Society Best Paper Awards Selection Panel. He has also served as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING for three terms over the periods 1997-1999, 2004-2007 and 2011- (and is currently an Area Editor).