

# An Online One Class Support Vector Machine-Based Person-Specific Fall Detection System for Monitoring an Elderly Individual in a Room Environment

Miao Yu, Yuanzhang Yu, Adel Rhuma, Syed Mohsen Raza Naqvi, *Member, IEEE*,  
Liang Wang, *Senior Member, IEEE*, and Jonathon A. Chambers, *Fellow, IEEE*

**Abstract**—In this paper, we propose a novel computer vision-based fall detection system for monitoring an elderly person in a home care, assistive living application. Initially, a single camera covering the full view of the room environment is used for the video recording of an elderly person's daily activities for a certain time period. The recorded video is then manually segmented into short video clips containing normal postures, which are used to compose the normal dataset. We use the codebook background subtraction technique to extract the human body silhouettes from the video clips in the normal dataset and information from ellipse fitting and shape description, together with position information, is used to provide features to describe the extracted posture silhouettes. The features are collected and an online one class support vector machine (OCSVM) method is applied to find the region in feature space to distinguish normal daily postures and abnormal postures such as falls. The resultant OCSVM model can also be updated by using the online scheme to adapt to new emerging normal postures and certain rules are added to reduce false alarm rate and thereby improve fall detection performance. From the comprehensive experimental evaluations on datasets for 12 people, we confirm that our proposed person-specific fall detection system can achieve excellent fall detection performance with 100% fall detection rate and only 3% false detection rate with the optimally tuned parameters. This work is a semiunsupervised fall detection system from a system perspective because although an unsupervised-type algorithm (OCSVM) is applied, human intervention is needed for segmenting and selecting of video clips containing normal postures. As such, our research represents a step toward a complete unsupervised fall detection system.

**Index Terms**—Assistive living, fall detection, health care, online one class support vector machine (OCSVM), posture detection.

## I. INTRODUCTION

**A**GING populations are an increasing issue across the globe particularly in developed countries. As shown in [1], the

old-age dependence ratio (the number of people 65 and over relative to those between 15 and 64) in the European Union is projected to double to 54% by 2050. 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 can cause problems for an elderly person physiologically and psychologically; besides, although many falls do not result in injuries, 47% of noninjured fallers cannot get up without assistance and this period of time spent immobile also affects their health. The detection of falls is required and is very important for elderly people's assistive living. When an elderly person falls, a fall detection system will detect this fall event and an alarm signal will be sent to certain caregivers (such as hospitals, health centers, or relatives) by some modern communication methods, and these caregivers will then provide assistance.

Methods have been proposed for detecting falls and are mainly divided into two categories: noncomputer vision-based methods and computer vision-based methods. For different types of methods, different sensors are applied. The most widely used sensors in noncomputer vision-based methods include accelerometers, floor vibration sensors, and acoustic sensors. In [3], Karantonis *et al.* proposed a real-time classification system for the types of human movement associated with the data acquired from a single, waist-mounted triaxial accelerometer unit. This system was able to distinguish between periods of activity and rest, recognize the postural orientation of the wearer, and detect events such as walking and falling. According to their experimental results, a fall detection rate of 95.6% was obtained. Multiple accelerometer sensors were applied in [4] and the data collected were sent to a personal server for processing over a wireless link. Compared with other commercial fall detection systems, this fall detection system applied a distributed processing paradigm which can achieve real-time data processing for fall detection, with minimal computational and consumption costs. The experiment from a dataset of 31 persons showed a fall detection rate of 100% and a false detection rate of 4.3% can be achieved by the proposed fall detection system. Floor vibration and acoustic sensors were also used in several works such as [5] and [6]. Y. Zigel *et al.* in [5] 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%. In [6], an acoustic fall detection system (FADE) that would

Manuscript received July 11, 2012; revised May 21, 2013 and January 29, 2013; accepted July 17, 2013. Date of publication July 23, 2013; date of current version November 12, 2013.

M. Yu, A. Rhuma, S. M. R. Naqvi, and J. A. Chambers are with the School of Electronic, Electrical and Systems Engineering, Loughborough University, Loughborough, LE11 3TU, U.K. (e-mail: m.yu@lboro.ac.uk; a.rhuma@lboro.ac.uk; s.m.r.naqvi@lboro.ac.uk; j.a.chambers@lboro.ac.uk).

Y. Yu is with the Shandong University of Technology, Zibo 255049, China (e-mail: yuyzhdsut@sina.com).

L. Wang is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100864, China (e-mail: wangliang@nlpr.ia.ac.cn).

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

Digital Object Identifier 10.1109/JBHI.2013.2274479

automatically signal a fall to the monitoring care giver was designed. A circular microphone array was applied to capture sounds in a room; when a sound was detected, FADE located the source, enhanced the signal, and classified it as “fall” or “non-fall”; and the sound source’s height information 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).

In the last ten 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. Compared with non-computer vision-based methods, computer vision-based methods have the following advantages: 1) they are nonintrusive, an elderly person need not wear some special equipment such as an accelerometer; 2) they are not easily affected by noises in the environment (suffered by floor vibration and acoustic sensors-based methods).

In [7], calibrated cameras were used to reconstruct the 3-D shape of people, and fall events were detected by analyzing the volume distribution along the vertical axis. When the major part of this distribution was abnormally near the floor during a predefined period of time, an alarm indicating a fall was triggered. A graphic processing unit was applied for efficient computation of the 3-D shape and the experimental results showed good performance of this system using multiple cameras (achieving 99.7% detection rate or better with four cameras or more). In [8], 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 elderly people perform their activities. The proposed framework was extremely flexible and rules can be modified, added, or removed to allow for per-resident customization. This system was tested on a dataset which contained 14 fall activities and 32 nonfall activities; all the fall activities were correctly detected and only two nonfall activities were mistaken as fall activities (100% fall detection rate and 6% false detection rate), which showed an acceptable level of performance.

Some supervised pattern recognition methods were applied in [9]–[11] for classifying different postures and activities for fall detection. In [9], Mihailidis *et al.* used a single camera to classify fall and nonfall activities. Silhouette features, lighting features, and flow features were extracted to allow the system to be robust to lighting, environment, and the presence of multiple moving objects. Three pattern recognition methods (logistic regression, neural network, and support vector machine) were compared in [9] and the neural network achieved the best performance with a fall detection rate of 92% and a false detection rate of 5%. In [10], four different types of postures (stand, bend, sit, and lie) were classified by a directed acyclic graph support vector machine classifier; the classification results, together with the floor region detected during a floor detection phase, were applied to detect falls. The fall detection system was tested on a

dataset of 15 people; a high fall detection rate (97.08%) and very low false detection rate (0.8%) were achieved. Instead of posture classification, Thome *et al.* [11] proposed a method based on short video sequence activity classification. In this study, a novel method was proposed to extract a person’s 3-D orientation information from multiple uncalibrated cameras. From extracted orientation information from a short video sequence, an improved version of hidden Markov model (layered hidden Markov model) 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) to make a thorough performance assessment.

The main problem for supervised fall detection methods is that they do not provide a person-specific solution for individuals. A large dataset needs to be constructed initially for training the supervised classifier (which should contain the data collected from many people in different views) for a supervised fall detection system; if a person does not fit the dataset very well (such as if he/she is obese), a good performance can definitely not be obtained for this specific person. Moreover, supervised fall detection methods will be affected by occlusions which happen in a real home environment. In order to solve these problems, unsupervised algorithms can be exploited. As described in [12], an unsupervised learning algorithm solves the problem of finding the hidden structure in unlabeled data or the normal model which unlabeled data follow. So, we can collect data (such as features extracted from postures or short video clips) from a particular elderly person’s daily activity video stream and these data can be used to construct the daily activity model with some unsupervised learning algorithm, and this model can then be used to distinguish falls and normal activities.

Representative works in applying unsupervised algorithms for fall detection include [13] and [14]. In [13], a ceiling-mounted, wide-angle camera was used for video recording and the particle filter technique was applied to track the human body with an ellipse model. From the tracking results, they obtained the position information and for normal activities; this was used to find the “usual activity region” by using an expectation maximization method. A fall was detected when a person’s position was outside the “usual activity region” for a certain time longer than the preset time threshold. In [14], a shape matching technique was used to track the person’s silhouette through the video sequence. By some shape analysis methods, the shape deformation was then calculated from the obtained silhouettes. The shape deformation along with the inactivity time of an old person were used as features to construct a Gaussian mixture model (GMM) to describe a person’s normal activity. This GMM was then used to detect falls and a multiple cameras scheme was applied to guarantee good performance. Although these works use unsupervised algorithms and provide person-specific solutions, either a specific camera (wide-angle) needs to be mounted at a particular position (ceiling) as in [13] or the data extracted to construct the normal model in both works were insufficient to describe fully an elderly person’s normal activities. The GMM is moreover rigid, and sometimes, it is not adequate to be applied to represent the normal model. Both of these two works also do not consider the problem of normal model updating, which is important because the elderly person’s activities will change throughout their daily lives. Considering these limitations, there

is therefore a clear business case for a relatively inexpensive video-based fall detection system for an enclosed environment which can operate with a retrofitted camera and can update the normal activity model of a specific individual through online learning.

In this study, we propose a person-specific fall detection system based on a novel unsupervised algorithm which is termed as the online one class support vector machine (OCSVM) classifier. First, the codebook (CB) background subtraction method [15] is applied to extract the human body silhouette from the frames in manually segmented video clips containing normal postures and some postprocessing is applied to improve the results. In order to fully describe the posture, three types of features, including ellipse features, shape structure features, and position features are extracted. After the extraction of these features, an online OCSVM is applied to describe the normal region described by these features for distinguishing normal and abnormal postures, which can also be updated to adapt to new postures. To further improve fall detection performance, we add two rules to reduce the false alarms, one rule is to measure the amplitude of the movement; if there is not a large movement, a fall will not be reported even though abnormal postures are detected by the online OCSVM. The other is the duration of an abnormal posture; a fall is reported only if the duration of an abnormal posture is longer than a threshold; this will effectively avoid false alarms when the person occasionally bends quickly and, for example, ties their shoes. This proposed fall detection system can achieve good performance which will be confirmed in Section III of this paper. A flowchart of the proposed fall detection system is shown in Fig. 1, the details of which will be presented in the next sections.

## II. METHODS

### A. Background Subtraction

Background subtraction is a common approach for discriminating moving objects from the background in visual surveillance [15]–[19]. In our fall detection system, we use the CB method [15] because of its advantages. There is no parametric assumption on the CB model and it shows the following merits as proposed in [15]: 1) resistance to artifacts of acquisition, digitization, and compression; 2) capability of coping with illumination changes; 3) adaptive and compressed background models that can capture structural background motion over a long period of time under limited memory; and 4) unconstrained training that allows moving foreground objects in the scene during the initial training period.

The CB method is available for both color and gray-scale images; it is a pixel-based approach, and initially, a CB 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 CB is constructed for this pixel, which includes a certain number of codewords. Each codeword, denoted by  $\mathbf{c}$ , consists of an RGB vector  $\mathbf{v} = (R, G, B)$  and a six-tuple  $\mathbf{aux} = (\hat{I}, \check{I}, f, \lambda, p, q)$ . The meanings of the six parameters in  $\mathbf{aux}$  are described in Table I:

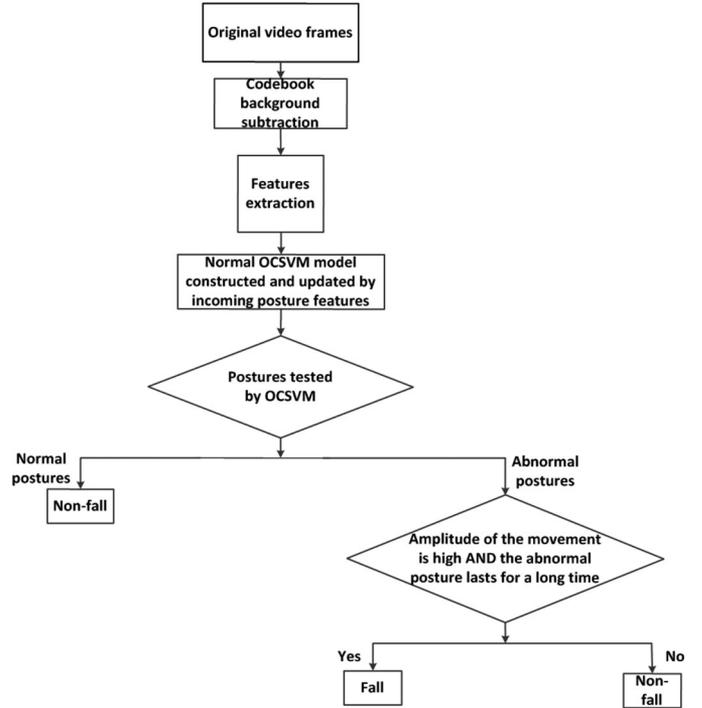


Fig. 1. Flowchart of the proposed fall detection system. A normal OCSVM model is constructed and updated online with the extracted features and the decision to determine fall or nonfall is made by combining the OCSVM classification result and two rules which measure the movement amplitude and duration of an abnormal posture.

TABLE I  
DEFINITIONS OF THE SIX PARAMETERS IN  $\mathbf{aux}$

$\hat{I}$	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.
$\lambda$	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 [15] and the trained CBs of pixels are then used for background subtraction purpose. For an incoming color frame  $\mathbf{f}$ , its pixel  $\mathbf{f}(x, y) = (R(x, y), G(x, y), B(x, y))$  (a 3-D vector) is determined as a foreground or background pixel by comparing  $\mathbf{f}(x, y)$  with codewords in the CB 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  $\varepsilon$  is a preset threshold value for comparison,  $I$  represents the L2-norm of  $\mathbf{f}(x, y)$ , and  $\hat{I}$  and  $\check{I}$  are the first two parameters of the six-tuple  $\mathbf{aux}$  vector of the codeword  $\mathbf{c}$ .

The  $\text{colordist}(\mathbf{f}(x, y), \mathbf{c})$  measures the chromatic difference between two color 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  $\mathbf{v} = (R, G, B)$  of codeword  $\mathbf{c}$ , and  $\|\cdot\|$  and  $\langle \cdot \rangle$  denote, respectively, the L2-norm and dot product operations.

The 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,  $\alpha$  and  $\beta$  are fixed to be 0.5 and 2 for background subtraction, which have been found empirically to be suitable values.

An important problem in background subtraction is background model updating, because the background will not remain constant (such as with gradual light change, or movement of furniture). The CB background subtraction method, therefore, provides a background model updating scheme. The matched codeword according to (1) is updated as shown in [15]. Besides, an additional cache model is introduced; if one codeword in the cache model is matched with 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 CB. 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 CB. Through the background model updating scheme, we can cope with change of the background in an indoor environment.

The obtained raw background subtraction results generally contain many noise artifacts, which include small ‘‘salt and pepper’’ noises [20] and large noises caused by movement of furniture. In order to remove such noises, some postprocessing (mentioned in [10] with some associated results) is applied to improve the background subtraction results.

## B. Features Extraction

1) *Ellipse Features*: The first set of features we extract from the human body silhouette is obtained from ellipse fitting. As proposed in [21], a moment-based method is applied to fit the ellipse. For a binary image  $f(x, y)$ , its moments are given as

$$m_{pq} = \sum_{x,y} x^p y^q f(x, y) \quad \text{with } p, q = 0, 1, 2, 3, \dots \quad (4)$$

By using the first- and zero-order spatial moments, we can compute the center of the ellipse  $(\bar{x}, \bar{y})$  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

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

where the central moment can be calculated as

$$u_{pq} = \sum_{x,y} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad \text{with } p, q = 0, 1, 2, 3, \dots \quad (6)$$

The remaining parameters to determine an ellipse are the major semiaxis  $a$  and the minor semiaxis  $b$ ; these two parameters can be obtained by calculating the greatest and least moments

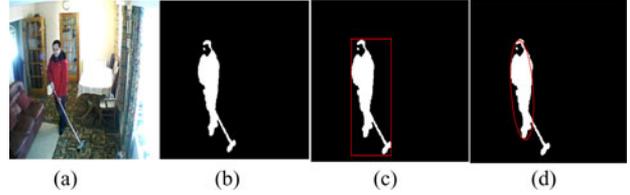


Fig. 2. Rectangle fitting and ellipse fitting results. (a) Original image for a person with a broom. (b) Background subtraction result (c) Rectangle fitting result. (d) Ellipse fitting result.

of inertia; here, we denote them as  $I_{max}$  and  $I_{min}$ . They can be calculated by evaluating the eigenvalues of the covariance matrix:

$$J = \begin{pmatrix} u_{20} & u_{11} \\ u_{11} & u_{02} \end{pmatrix} \quad (7)$$

$I_{min}$  and  $I_{max}$  are the smallest and largest eigenvalues of matrix  $J$ , respectively, calculated as

$$I_{min} = \frac{u_{20} + u_{02} - \sqrt{(u_{20} - u_{02})^2 + 4u_{11}^2}}{2} \quad (8)$$

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

After obtaining  $I_{max}$  and  $I_{min}$ , the major semiaxis  $a$  and minor semiaxis  $b$  can be calculated as

$$a = (4/\pi)^{1/4} \left[ \frac{I_{(max)^3}}{I_{min}} \right]^{1/8} \quad (10)$$

$$b = (4/\pi)^{1/4} \left[ \frac{I_{(min)^3}}{I_{max}} \right]^{1/8}. \quad (11)$$

The ellipse fitting result is shown in Fig. 2; for comparison, the simple blob-based rectangle fitting result used in [22] is also presented; we can see that the ellipse fitting is better fitted to the human body region for a person with a broom. After ellipse fitting, the orientation of the ellipse (denoted as  $\theta$ ) and the ratio between  $a$  and  $b$  (denoted as  $\rho$ ) are taken as features to describe a human body posture’s general property.

Features obtained from the ellipse fitting can describe postures in a general way, but definitely, 2-D features alone cannot fully describe postures in detail for distinguishing different postures, as shown in Fig. 3. So, in order for a more detailed posture description, other features are needed.

2) *Shape-Structure Features*: More details of a posture can be reflected by the posture’s shape and structure information, which can be extracted from a single centroid context method as proposed in [2]. Initially, the perimeter contour of a human body posture is extracted by some contour detection method [20], which is represented as  $\text{contourlist} = [\text{point}_1, \dots, \text{point}_N]$  where  $\text{point}_i$  is a particular point on the perimeter contour. One example of an extracted perimeter contour is shown in Fig. 4(c). Not all the points in the down-sampled  $\text{contourlist}$  are necessary to represent a boundary, only the points with high curvature are the ‘‘key points’’ which determine a boundary shape and these points form a more concise perimeter contour representation [see Fig. 4 (d)]. For a 2-D point  $p$ , its angle can be calculated by

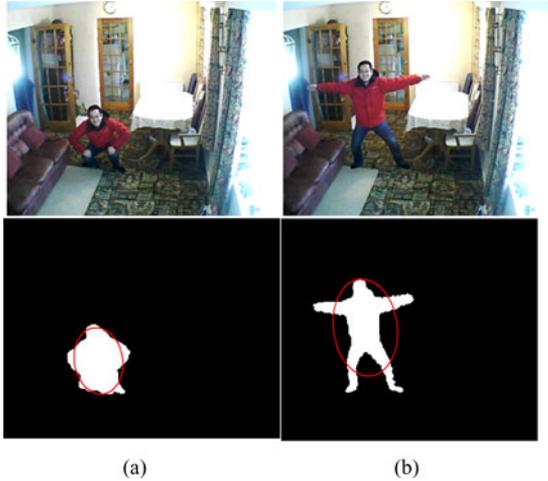


Fig. 3. Ellipse fitting results for two postures: (a) crouching with  $\theta = -0.2010$  and  $\rho = 0.7702$ ; (b) stretching with  $\theta = -0.1902$  and  $\rho = 0.7517$ . Although the two postures are obviously different, the obtained ellipse features are very similar.

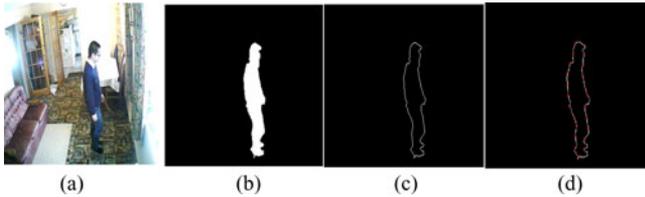


Fig. 4. Result of the perimeter contour detection. (a) Original image. (b) Background subtraction result. (c) Perimeter contour detection result. (d) Key points detection results.

$$\text{angle}(p) = \arccos \frac{\|p - p^+\|^2 + \|p - p^-\|^2 - \|p^+ - p^-\|^2}{2 \|p - p^-\| \times \|p - p^+\|} \quad (12)$$

where  $p^+$  and  $p^-$  are selected from both sides of  $p$  along the boundary and satisfy  $d_{\min} \leq \|p - p^+\| \leq d_{\max}$  and  $d_{\min} \leq \|p - p^-\| \leq d_{\max}$ , where  $d_{\max}$  and  $d_{\min}$  are properly chosen thresholds. If  $\text{angle}(p)$  is low, then this point is taken as a “key point.” For two “key points,” their distance should be larger than a proper threshold so that if several points with low  $\text{angle}(p)$  are near to each other, only one representation point is chosen as the “key point.”

The results of extracted “key points” can be further applied to extract the skeleton structure of a posture. As shown in [2], the constrained Delaunay triangulation technique is applied to divide a human posture into triangular meshes according to the “key points.” The centroids of the triangles are connected to form the skeleton. The result of extracting the posture of a human body skeleton is shown in Fig. 5.

After extracting the “key points” representing the perimeter contour information and triangular centroid representing the skeleton structure information, we apply an accurate and efficient single centroid context shape descriptor to “describe” the information and thereby obtain the corresponding features (the more complicated multiple centroid context method as in [2] can also be applied for a more detailed description; however, for the multiple centroid context method, the computational com-

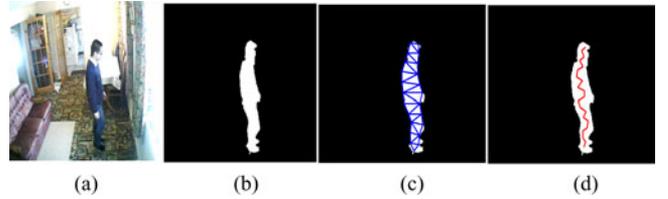


Fig. 5. Extraction of the posture of a human body skeleton. (a) Original image. (b) Background subtraction result. (c) Result of the constrained Delaunay triangulation. (d) Extracted skeleton by connecting the triangular centroid.

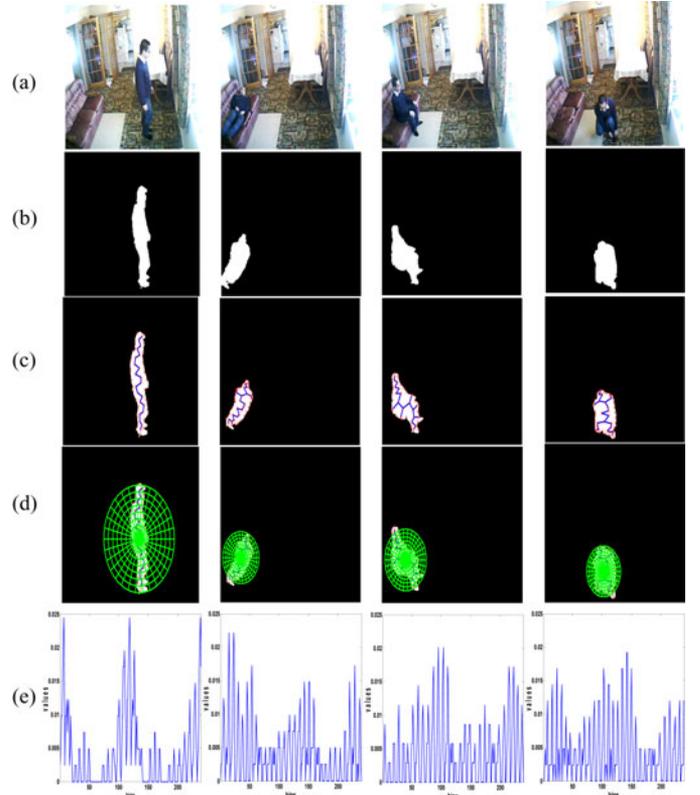


Fig. 6. Histograms for four postures obtained from centroid context descriptor. (a) Original images for four postures (stand, lie, sit, and bend). (b) Background subtraction results with postprocessing. (c) Extracted skeleton (marked as blue) and points (including the “key points” on the contour and triangular centroid, marked as red). (d) Polar coordinate system, which is composed of 8 shells and 30 sectors, total 240 bins. (e) Finally obtained histograms, the horizontal axis represents the indices of bins and the vertical axis represents the values of the bins.

plexity for feature extraction and posture comparison is much larger than in the single centroid context method, and the performances of these two methods are very similar according to [2], so in our paper, the single centroid context method is preferred. Initially, the center of gravity of the whole human body posture is calculated; then, fixing this centroid point as the origin, a polar coordinate system is constructed, which is equally divided into  $m$  shells and  $n$  sectors to form  $m \times n$  bins (here,  $m$  and  $n$  were chosen as 8 and 30, respectively, from empirical study), as shown in Fig. 6(d). An  $m \times n$  histogram is then constructed to obtain the spatial distribution of points. In our work, we made two modifications on the histogram construction procedure as proposed in [2].

- 1) Not only the triangular centroid points are used for histogram construction, the “key points” are also applied so that the constructed histogram can reflect both the skeleton structure and perimeter contour information.
- 2) Instead of simply calculating the number of points in each bin for histogram construction, an improved strategy is applied to get a more accurate histogram result: for a point (either the “key point” on the perimeter contour or the triangle’s centroid), if it is in the  $k$ th bin, then histogram values are updated as

$$h_{\text{new}}(k) = h_{\text{old}}(k) + 1 \quad (13)$$

$$h_{\text{new}}(k_{\text{neighbors}}) = h_{\text{old}}(k_{\text{neighbors}}) + 0.5 \quad (14)$$

where  $h_{\text{new}}(k)$  and  $h_{\text{new}}(k_{\text{neighbors}})$  represent, respectively, new histogram values of the  $k$ th bin and its neighbors after updating, and  $h_{\text{old}}(k)$  and  $h_{\text{old}}(k_{\text{neighbors}})$  represent old values, initially the histogram values are set to zero.

Finally, we normalized the histogram to make sure the summation of the values of all bins is unity and the histogram is obtained as the shape-structure feature. Fig. 6(d) shows the corresponding histograms for four postures of standing, sitting, bending, and lying. The resulting histograms are taken as the shape-structure features, which give a more detailed posture description compared with the 2-D ellipse features.

3) *Position Features*: Although the combination of the ellipse features and shape-structure features can describe a posture in a detailed way; however, they cannot distinguish similar postures at different positions (such as a lie posture on the ground and on the sofa). In order to solve this problem, we also incorporate position information into the final features. The centroid position  $(x, y)$  of the human body posture is recorded and concatenated with ellipse features and shape structure features as the final feature which is then used to train the OCSVM; all the feature components are normalized into  $[0,1]$  to keep the same scale.

### C. Online OCSVM

The concept of an OCSVM was first proposed in [23], which is a popular approach for detecting anomalies, compared with the single Gaussian model and GMM in [12]; OCSVM can describe the data in the feature space in a more flexible way (it does not assume that the data need to follow certain types of distributions). The basic idea behind the OCSVM is that given a dataset drawn from an underlying probability distribution  $P$  for the minority class, the OCSVM estimates a function  $f$  that is positive in a region  $S$  and negative in its complement in a mapped high-dimensional space, where  $S$  is the “most-likely region”—a subset of the input space such that a test point drawn from  $P$  lies outside of  $S$  equals some a priori specified value between 0 and 1. Fig. 7 illustrates the basic idea of an OCSVM; a hyperplane  $f(\mathbf{x}) = 0$  is found to separate the normal samples and outliers in the mapped high-dimensional space.

As it is mentioned in [24] and [25],  $f$  is a linear operator in a reproducing kernel Hilbert space (RKHS). At time instance  $t$ ,

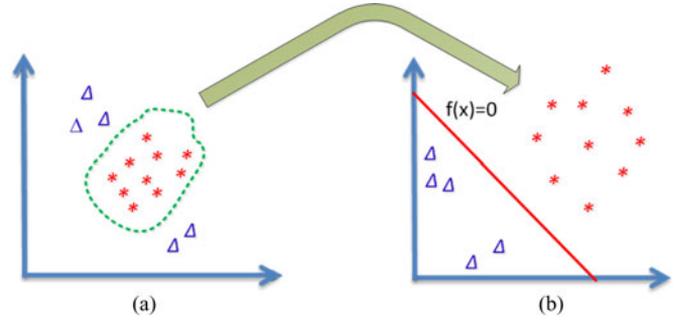


Fig. 7. Illustration of the basic idea of OCSVM, the normal samples are marked as red stars and the outliers are marked as blue triangles. (a) Sample distribution in the original space. The samples become linearly separable in a mapped high-dimensional space (b) and a hyperplane  $f(\mathbf{x}) = 0$  is applied to separate them.

can be explicitly represented as by the samples  $\mathbf{x}_1, \dots, \mathbf{x}_{t-1}$  as

$$f_t = \sum_{i=1}^{t-1} \alpha_i k(\mathbf{x}_i, \cdot) \quad (15)$$

where  $\mathbf{x}_i, i = 1, \dots, t-1$  are the incoming samples before time  $t$  and  $k(\mathbf{x}, \cdot)$  is a kernel function, and the popular RBF kernel ( $k(\mathbf{x}, \mathbf{y}) = \exp(-\lambda \|\mathbf{x} - \mathbf{y}\|^2)$ ) is used in this study.

We refer to the online OCSVM algorithm as proposed in [24] and [25] for estimating and updating  $f$ ; the online algorithm is adopted here for two reasons: 1) an online algorithm achieves a much faster training time than the traditional batch algorithm [23], especially for a large training dataset; 2) the online scheme provides an efficient and straightforward way for OCSVM model updating, which is very useful in our application to adapt new emerging postures into the already trained normal model. The online OCSVM algorithm is operated in a sample-by-sample way; when a new sample  $x_t$  arrives, the online OCSVM algorithm finds a new set of coefficients  $\alpha_i, i = 1, \dots, t$ , to determine a new function  $f_{t+1} = \sum_{i=1}^t \alpha_i k(\mathbf{x}_i, \cdot)$ . This can be achieved by minimizing a modified regularized risk for the online scheme as proposed in [24]:

$$R(f) = \frac{1}{2} \|f - f_t\|_H^2 + \eta \left( \frac{\lambda}{2} \|f\|_H^2 + C \cdot (\gamma - f(\mathbf{x}_t))_+ \right) \quad (16)$$

where  $\|\cdot\|_H$  means the RKHS distance [25] and the first term measures the RKHS distance of  $f$  from the previous predicted function, and the second term is the traditional regularized risk, which controls the complexity of  $f$  and the convex loss for the sample  $\mathbf{x}_t$ .

Due to the convex property of  $R(f)$ , the optimal  $f_{t+1}$  can be found by setting the gradient of  $R(f)$  to zero, by some algebraic operations as mentioned in [24]; finally, we obtain the optimal  $f_{t+1}$  as

$$f_{t+1} = \frac{1}{1 + \eta\lambda} f_t - \frac{C}{1 + \eta\lambda} \beta_t k(\mathbf{x}_t, \cdot) \quad (17)$$

where  $\beta_t \in [-1, 0]$  and normally  $\eta$  is set to one. We use an auxiliary variable  $\tau = \frac{\lambda}{1+\lambda}$  and rewrite  $f_{t+1}$  as

$$f_{t+1} = (1 - \tau) f_t - (1 - \tau) C \beta_t k(\mathbf{x}_t, \cdot). \quad (18)$$

Note previously, we represent  $f_{t+1}$  as the following form:

$$f_{t+1} = \sum_{i=1}^t \alpha_i k(\mathbf{x}_i, \cdot) \quad (19)$$

By comparing (18) and (19), we can see the coefficients of  $f_{t+1}$  can be updated as

$$\begin{aligned} \alpha_i &= (1 - \tau)\alpha_i, \quad i = 1, \dots, t-1 \\ \alpha_t &= -(1 - \tau)C\beta_t. \end{aligned} \quad (20)$$

Moreover, as discussed in [24] and [25], for the optimal  $\alpha_t$ , we need to make  $\gamma - f_{t+1}(\mathbf{x}_t) = 0$  to minimize the convex loss  $(\gamma - f(\mathbf{x}_t))_+$  in (16), which leads to

$$\gamma - ((1 - \tau)f_t(\mathbf{x}_t) + \alpha_t k(\mathbf{x}_t, \mathbf{x}_t)) = 0 \quad (21)$$

and the optimal  $\alpha_t$  is calculated as

$$\alpha_t = \frac{\gamma - (1 - \tau)f_t(\mathbf{x}_t)}{k(\mathbf{x}_t, \mathbf{x}_t)}. \quad (22)$$

From the above derivations, we can update the coefficients of  $f_{t+1}$  for a new incoming sample  $\mathbf{x}_t$  using (22) and (20). Note that  $\alpha_t$  must be truncated into the range of  $[0, (1 - \tau)C]$  if it lies outside this range after (22). As time increases, we can see the number of samples determining the OCSVM function  $f$  will become very large, which will lead to memory overload. So in the real application, we have to delete the previous samples whose coefficients become lower than a small threshold as  $f$  updates.

For the obtained  $f$  at a time instance  $t$ , we can determine whether the next incoming sample  $\mathbf{x}_{t+1}$  is a normal sample or not by

$$D(\mathbf{x}_{t+1}) = \begin{cases} 1, & \gamma - f(\mathbf{x}_{t+1}) \leq \text{threshold}_{\text{OCSVM}} \\ 0, & \text{otherwise} \end{cases} \quad (23)$$

where  $D(\cdot)$  is an indication function with “1” being normal and “0” being abnormal. The value of  $\gamma - f(\mathbf{x}_{t+1})$  is calculated and compared with a preset threshold to make a decision. As in [24], the value of  $\gamma$  is fixed to be one throughout this work.

#### D. Rules to Determine a Fall

Although the normal postures model constructed by the on-line OCSVM can effectively distinguish normal and abnormal postures; however, sometimes false alarms will still occur because not all the abnormal postures represent falls. In order to reduce false alarms, we introduce two rules when an abnormal posture is detected.

1) A fall is only reported when a large movement is detected. In [21], a measurement of the amplitude of movement is proposed by using the motion history image (MHI); however, the frame difference results used to construct the MHI are easily affected by noise and illumination change in the environment. In our work, we propose a new measurement based on the motion energy image (MEI) [26]. The amplitude of movement is measured by the area ratio (denoted as AR) between the area of certain number of MEI frames and the area of the current frame's foreground region. One example is shown in Fig. 8; from this figure, we can see as a large movement, fall activity

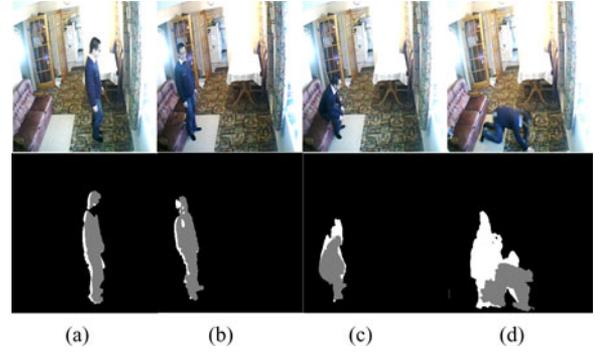


Fig. 8. Movement amplitude measurement for four activities. The first line shows the original images and the second line shows the MEI results (nonblack region with the current images' foreground regions marked as gray). The calculated AR values are: (a) walking, AR = 1.2284; (b) standing, AR = 1.1686; (c) sitting, AR = 1.3344; and (d) falling, AR = 2.2360.

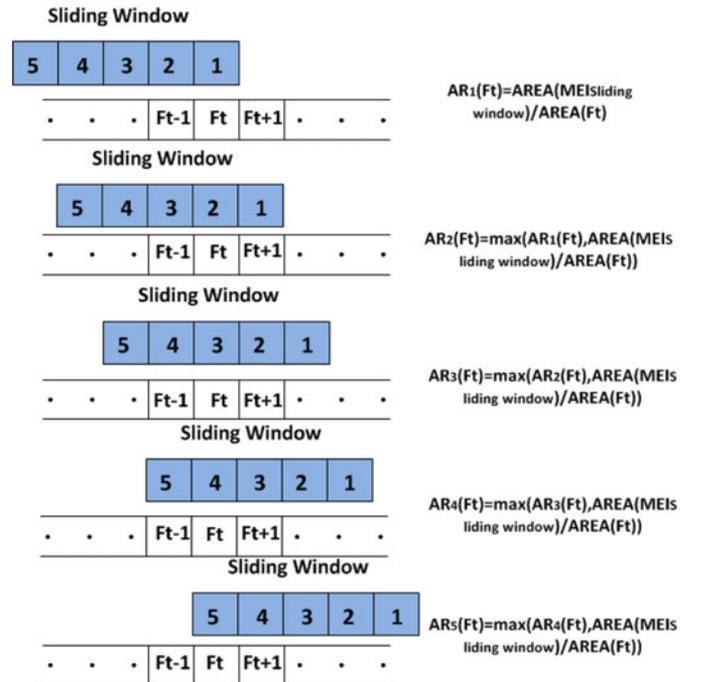


Fig. 9. Estimation of the AR value for a frame using a sliding window (a length of five). At each time, the sliding window moves forward over one frame and the AR value for frame  $F_t$  is calculated as the maximum value between the new calculated AR value and the previous one. The final AR value ( $AR_5$ ) is obtained when the sliding window passes over the frame.

has a larger AR value than the other three types of activities (walking, sitting, and bending).

In a video sequence, we use a sliding window method to estimate the AR value for each frame, which is illustrated in Fig. 9. For a particular frame  $F_t$ , the AR value is calculated as the ratio between the area of the MEI of the frames in the sliding window and the area of the  $F_t$ 's foreground region. For the next time, the sliding window moves forward over one frame and the new AR value is calculated, which is compared with the previous one and the larger value is then retained. The final AR value of  $F_t$  is obtained when the sliding window passes over this frame [as shown in Fig. 9(e)], which reflects the largest movement around this frame. The calculated AR value is compared with



Fig. 10. Real-home environment for experiments.

a threshold (denoted as  $\text{threshold}_{AR}$ ) to determine whether a large movement occurs or not.

2) Normally, after an old person falls, he/she will be most likely to lie on the ground for a certain time interval. So, a fall is reported only if the abnormal posture lasts longer than a time interval (denoted as  $\text{threshold}_{\text{abnormal interval}}$ ), this will avoid occasional abnormal postures which do not last for a predefined threshold (such as bending to fasten the shoe ties).

These two rules, together with the constructed OCSVM model, compose a robust fall detection system. Excellent fall detection performance can be achieved under the properly tuned parameters set, which is shown in Section III.

### III. RESULTS

#### A. Experimental Settings

All the experiments were performed in a real-home environment where the elderly people live, as shown in Fig. 10. We used a normal personal laptop with a configuration of Intel Core Two 2.10-GHz CPU with a 1.00-GB memory for data processing. A USB camera was connected to the laptop for recording the video streams, which was placed on the wall of the room close to the ceiling to cover the full view of the home environment. VC++ 6.0 and MATLAB R2010b were used for video processing (including foreground extraction, features extraction, and normal model construction and updating). The video sequence was recorded at a frame rate of 5 frames/s as in [14].

#### B. Background Subtraction

Three popular background subtraction methods (approximate median filter (AMF) method, GMM method, and CB method) are compared for extracting the human foreground region in the indoor environment. A video sequence of walking with the length about 30 s is recorded, and the three background subtraction methods with the tuned optimal parameters were tested on this sequence (initially, 50 background frames are used for background model training). For a fair comparison, no post-processing technique is applied. Fig. 11 shows the qualitative comparison results on several selective frames; we can see that the CB background subtraction achieves the best performance, most foreground regions are detected, and the least amount of shadow region is mistaken as the foreground. The reason behind it is that the CB method makes use of both the pixel's intensity and color information, compared with the standard AMF and GMM methods in which only the intensity information is used. For a comprehensive quantitative analysis, as in [27], the pre-

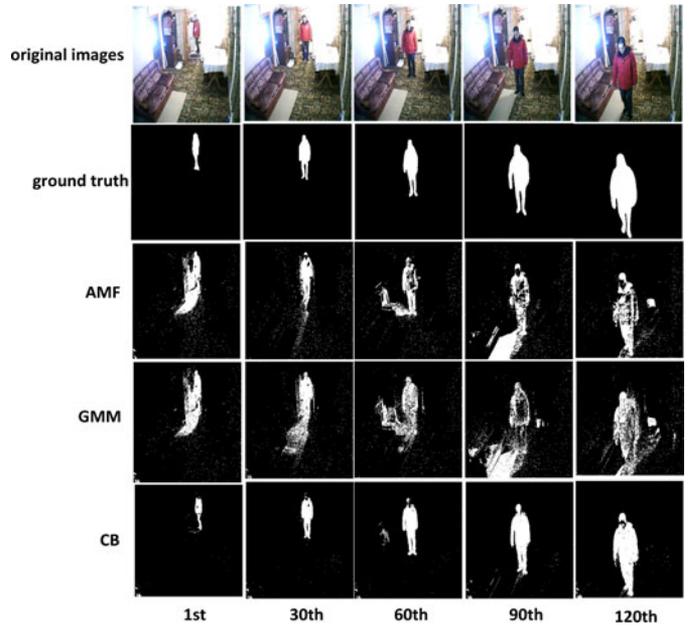


Fig. 11. Comparison of three background subtraction methods for the 50th, 100th, 150th, 200th, and 250th frames in a video sequence.

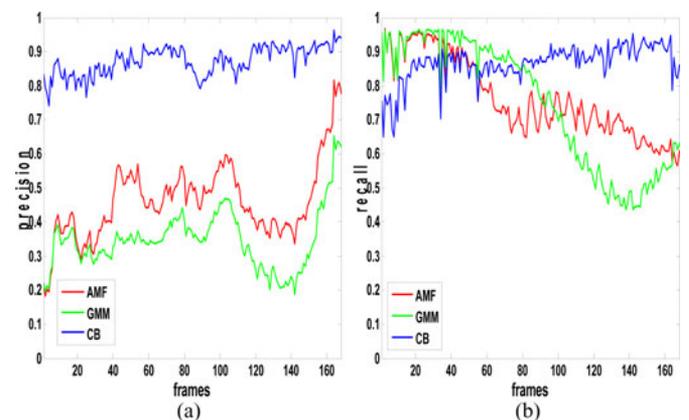


Fig. 12. Quantitative comparisons of three background subtraction methods for a video sequence. (a) Precision comparison. (b) Recall comparison.

cision (defined as the division between the number of correctly detected foreground pixels and that of totally detected foreground pixels) and recall (defined as the division between the number of correctly detected foreground pixels and that of total foreground pixels) are evaluated; for a perfect background subtraction result, both values should be unity. Precision and recall values of three background subtraction methods are calculated for every frame in the sequence; the results are presented in Fig. 12, from which we can see that the CB methods achieve a much better precision result for the whole sequence; for the recall, although initially the CB method is a bit worse than the AMF and GMM methods, the recall value of the CB method will not drop dramatically like the AMF and GMM methods (some foreground region is mistaken as the background due to intensity similarity) and after around 80 frames the recall value of the CB method remains the best.

TABLE II  
SUMMARY OF THE FREQUENCY OF REPRESENTATIVE ACTIVITIES OF AN ELDERLY PERSON DURING ONE WEEK

Activities	Descriptions	Frequency
Walking	The elderly person walks to move between different places of the room or do some cleaning activities.	16
Standing	The elderly person stands almost still to watch television.	6
Sitting	The elderly person sits to have a rest (either watching TV or eating fruits)	8
Lying	The elderly person lies on the sofa for a nap or watching TV.	8



Fig. 13. Sample postures of simulated normal activities; (a) lying, (b) standing, (c) walking, (d) sitting, and (e) sitting (with occlusion).

### C. Fall Detection System Evaluation

Twelve people (eight males and four females) are invited to participate in the experiments for the fall detection system evaluation. For each person, two datasets are simulated.

Dataset 1 consists of the normal postures captured from different simulated daily activities for constructing the OCSVM model. To simulate the real scenario, we interviewed a 75-year-old healthy old person and the frequency of representative activities during one week is summarized in Table II. According to this table, each participant simulates 38 activities (including 16 walking activities, 6 standing activities, 8 sitting activities, and 8 lying activities). And for each activity, a video clip of 15 s is recorded to capture postures representing the corresponding activity. Fig. 13 shows the captured postures for some simulated activities.

Dataset 2 is a test dataset which consists of 18 simulated falling activities in different positions of the room and 18 simulated normal activities for testing the performance of the fall detection system using the constructed OCSVM model by the

obtained posture features in dataset 1 and two rules proposed in Section II-D.

To evaluate the performance, two evaluation metrics are introduced, which include the true positive rate (TPR) and false negative rate (FNR), which are defined as:

$$\text{TPR} = \frac{\text{No. of correctly detected falls}}{\text{No. of falls}} \quad (24)$$

$$\text{FNR} = \frac{\text{No. of nonfalls which are mistaken as falls}}{\text{No. of nonfalls}}. \quad (25)$$

For a good fall detection method, TPR should be 1 and FNR should be 0.

The 12 people are divided into two groups of six people (each group contains four males and two females). The datasets (including the normal posture dataset and test dataset for every person) in the first group are used for tuning key parameters of the proposed OCSVM based fall detection system within certain ranges (the parameters and their corresponding ranges are presented in Table III). To save the computational cost for parameters searching, genetic algorithm is applied for searching the optimal parameters to maximize  $\sqrt{\text{TPR} * (1 - \text{FNR})}$  instead of the time-consuming grid search method (the genetic algorithm implemented in the MATLAB global optimization toolbox is used in this study). The obtained optimal parameter set is:  $\lambda = 15$ ,  $C = 1$ ,  $\tau = 10^{-5}$ ,  $\text{threshold}_{\text{OCSVM}} = 0.7$ ,  $\text{threshold}_{\text{AR}} = 1.75$ , and  $\text{threshold}_{\text{abnormal interval}} = 45$ , which achieves a TPR of 1 and FNR of 0 for the six test datasets (including 108 falls and 108 nonfalls) of the first group.

The second group of six people is used to test the generalization performance of the fall detection system. With the tuned optimal parameter set, the corresponding OCSVM model for every person is constructed by the normal posture dataset and the performance is evaluated by the corresponding test dataset. The fall detection performance on the six test datasets in this group (including 108 falls and 108 nonfalls) is presented in Table IV. For comparison, we also implement and give the performance of the state-of-the-art unsupervised method in [14] which uses the shape deformation features (including the mean cost and full procrustes distance between two sets of matched silhouette points in consecutive frames) to construct GMM for fall detection (Note, for a fair comparison, that the key parameters of the methods in [14], including the number of the Gaussian components, the threshold for distinguishing falls and nonfalls, the inactivity interval, etc., are also tuned to be optimal by genetic algorithm using the datasets in the first group). From this table, we can see that the proposed OCSVM-based fall detection method not only has a better fall detection performance than the method in [14], but also achieves a high efficiency (because the proposed method avoids the time-consuming points matching procedure).

TABLE III  
KEY PARAMETERS OF THE OCSVM-BASED FALL DETECTION METHOD AND CORRESPONDING TUNING RANGE

Parameters	Descriptions	Range for tuning
$\lambda$	kernel parameter for OCSVM	[0.2,20]
$C$	penalty parameter for OCSVM	[0.1,10]
$\tau$	decay rate for OCSVM	$[10^{-6}, 10^{-1}]$
$threshold_{OCSVM}$	Threshold parameter for OCSVM model for abnormal detection.	[0,2]
$threshold_{AR}$	Threshold parameter for AR value for large amplitude activity detection.	[1,2]
$threshold_{abnormal\ interval}$	Threshold parameter of the abnormal posture interval for fall confirmation.	[30,60] (in frames)

TABLE IV  
COMPARISON OF THE PROPOSED FALL DETECTION METHOD AND THE METHOD PROPOSED IN THE WORK OF C. ROUGIER *et al.*

	Proposed method	Methods in C. Rougier <i>et al.</i> 's work	
		GMM+meancost	GMM+full procrustes distance
True Positive Rate	100%	81%	89%
False Negative Rate	3%	7%	11%
Approximate execution time(ms) <sup>1</sup>	50	200	200

<sup>1</sup>The execution time includes background subtraction time, feature extraction time and classifier (OCSVM or GMM) execution time.



Fig. 14. Introduced new sitting posture. (a) Chair is put at a new position. (b) Person sits at the new position to introduce a new sitting posture.

#### D. New Postures Adaptation

We have to remark that an old person's behavior will not be unchanged over time. Sometimes, his behavior changes and new postures emerge, so a good fall detection system needs to be capable of adapting to the changes. In the following, we give an example of adapting to new postures with the online OCSVM. Fig. 14 shows that a chair is put at a new position and a new sitting posture is introduced.

To adapt this new posture to the normal model, the online OCSVM scheme described in Section II-C is applied to update the trained OCSVM model using the features of this new posture extracted from a video sequence. For the updating procedure, we chose different  $C$  (penalty parameter) values and other parameters were kept the same as the tuned optimal ones; the evolutions of the OCSVM results for this sequence during the updating procedure are shown in Fig. 15. From this figure, we can see that by model updating, the OCSVM value for this new sitting posture increases with time; initially, the OCSVM value for this new posture is below the threshold, but this value will increase over time and exceed the threshold with the aid of the online OCSVM scheme for updating; besides, the parameter  $C$  controls the time for the new posture to be adapted to the normal model, while a larger  $C$  means a faster adaptation time.

In order to illustrate the advantage of OCSVM model updating, we present an example that a person who sits at the new position bends quickly to pick something (selective frame samples are shown in Fig. 16). And in Fig. 17, the AR values and OCSVM results are calculated for this fast bending sequence ex-

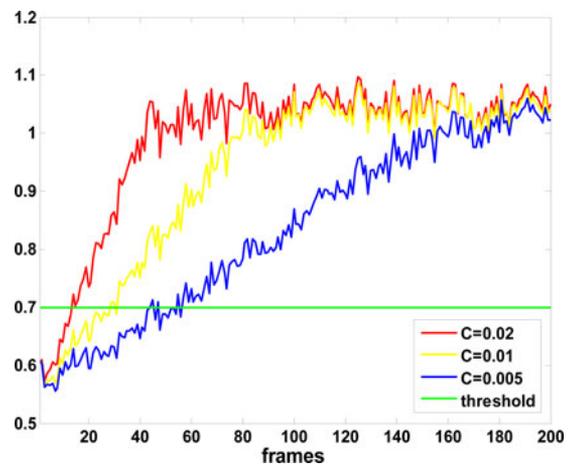


Fig. 15. Evolutions of OCSVM values using the online updating scheme with different penalty parameters  $C$ .



Fig. 16. Selective frame samples of fast bending to pick something at a new position.

ample. From the AR results shown in Fig. 17(a), we can see that large movement (bending) is detected during the initial frames; Fig. 17(b) shows the results of the OCSVM models with and without the updating procedure; we can see that results of the two models all fall below the threshold during the initial frames when bending occurs. However, for the updated OCSVM model, the value returns to be above the threshold when the person recovers to sit after a very short interval, so according to Rule 2, no falls are reported. Additionally, for the OCSVM model without updating, the value is always less than the threshold and a fall is wrongly reported when the abnormal state lasts for longer than 45 frames.

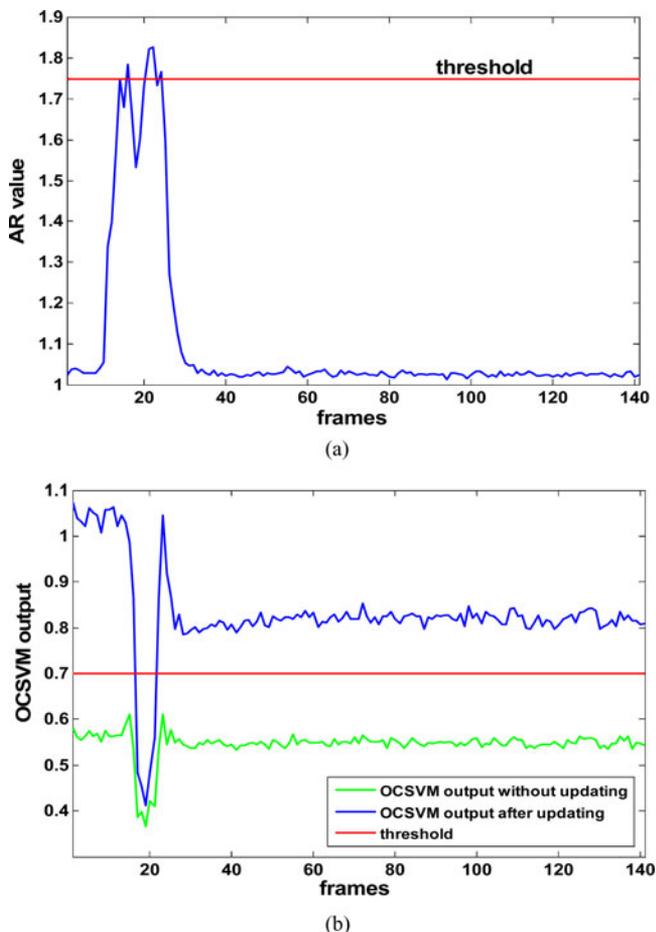


Fig. 17. Variations of (a) AR values and (b) OCSVM values for a video sequence of fast bending while sitting at new position. For comparison, both the values of the OCSVM models with and without updating are presented.

#### IV. DISCUSSIONS

The fall detection method proposed in this paper is based on the online OCSVM scheme for normal posture model construction, with two rules being introduced to reduce false alarms. Compared with the supervised methods for fall detection as proposed in [9]–[11], the proposed method need not have a training dataset which is obtained from different people captured in different views. The collected normal posture samples from the particular monitored elderly person can be used to construct a normal model to distinguish fall and nonfall activities with proper parameters, which provides a *person-specific solution*. Furthermore, unlike the supervised methods, the proposed method is not affected by occlusions because both the occluded and nonoccluded normal postures (samples are shown in Fig. 13) are used to construct the normal model to distinguish normal and abnormal postures.

Compared with traditional unsupervised methods in [13] and [14], our proposed fall detection system has advantages in both practicality and performance. There is no need for a wide-angle camera; only an ordinary USB camera is needed in our fall detection system and the camera needs not to be installed on the ceiling as in [13], which is more practical in the real applica-

tion. And compared with the method in [14], our fall detection system performs better both in fall detection rate and computational time as analyzed in the experimental evaluations. Besides, in [13] and [14], they did not consider the problem of updating (a batch algorithm was applied in both works for training and the initially constructed normal GMM model was regarded as permanent), which is not plausible in the real application because the normal postures will always change. In our work, we show that the new postures can be adapted to the normal model by the online OCSVM scheme.

However, in the proposed fall detection method, only features extracted from a single posture are extracted to construct the normal model. So, two activities of “lying on the bed” and “falling on the bed” cannot be distinguished (both end up with a very similar “lie” posture at similar positions); in order to distinguish these two activities, we need to extract features not only from just a frame, but from a short video sequence which contain more information of a particular activity. Besides, it is generally not enough to use only one camera in a large area room environment (sometimes the elderly person will be out of the range of the camera); in this case, multiple cameras are needed to cover all the area of the room and the results of different cameras are combined to make a final decision.

The proposed fall detection method is currently designed for monitoring a single elderly person staying alone at home. For multiple people case (such as some visitors), there is no need for the fall detection system to operate (if the elderly person falls, other people can be asked for help). Either the fall detection system will be turned OFF manually or automatically with the aid of some people counting technique as shown in [28] and [29] (if more than one person is detected by the people counting algorithm, the fall detection system is turned OFF automatically in the code). Sometimes, an elderly person may have a large size pet; in this case, only the extracted foreground region representing the human body silhouette is used for fall detection; some object classification technique such as [30] and [31] can determine whether the extracted foreground region is a human body silhouette or pet silhouette.

We remark that at the algorithm level, OCSVM is an unsupervised-type algorithm estimating the normal model from unlabeled data; however, at the system level, there is an act of supervision during the initial setup and during run time (the recorded video sequence is manually segmented into short video clips and the ones containing normal activities are selected for training and new postures adaptation as described early in the paper). In this sense, this proposed system is not a fully unsupervised fall detection system but rather a semiunsupervised one which exploits an unsupervised algorithm but needs human intervention to a certain extent as in [13] and [14]. In order to achieve a fully unsupervised fall detection system, advanced video segmentation algorithms as mentioned in [32] should be incorporated which can automatically segment the recorded video sequence into short video clips containing particular activities and corresponding features from these segmented short video clips can then be extracted and applied for the OCSVM model construction or adaptation. In this way, the human intervention can be avoided and a fully unsupervised fall detection system could then be constructed.

## V. CONCLUSION

In this paper, we have used a novel online OCSVM learning algorithm to detect falls for assisting an elderly person living alone at home. A single USB camera was placed properly to cover the full view of the home environment for recording. CB background subtraction was used for extracting the human body postures; a combination of three types of features, including ellipse features, shape-structure features, and position features was extracted from the initially selected video clips containing normal postures to build the normal model by an online OCSVM scheme, which is flexible and can be updated to adapt to new emerging postures. Two rules were also introduced to reduce the FNR of the proposed fall detection system. The experimental results showed that our proposed system can achieve a good performance with a very economical configuration (only a normal USB camera and a personal laptop are needed). Currently, this system remains as a semiunsupervised fall detection system because although an unsupervised learning algorithm is applied, there is still need for human intervention for the segmentation and selection of video clips. It can potentially be extended to a fully unsupervised fall detection system by incorporating novel automatic video segmentation algorithms. Besides, further improvements of the proposed system can also be obtained by using more elegant features from short video sequences which distinguish different activities more effectively, and multiple cameras which cover all the area in a large area room environment for better monitoring. Such extensions and improvements of this proposed system validated on larger datasets are our next research steps.

## ACKNOWLEDGMENT

The authors wish to acknowledge the valuable help of the Associate Editor and anonymous reviewers in improving the quality of their paper.

## REFERENCES

- [1] G. Carone and D. Costello, "Can Europe afford to grow old?," *Int. Monetary Fund Finance Dev. Mag.*, vol. 43, no. 3, 2006.
- [2] J. Hsieh, Y. Hsu, H. Liao, and C. Chen, "Video-based human movement analysis and its application to surveillance systems," *IEEE Trans. Multimedia*, vol. 10, no. 3, pp. 372–384, Apr. 2008.
- [3] D. Karantonis, M. Narayanan, M. Mathie, N. Lovell, and B. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Trans. Inf. Technol. Biomed.*, vol. 10, no. 1, pp. 156–167, Jan. 2006.
- [4] M. Estudillo-Valderrama, L. Roa, J. Reina-Tosina, and D. Naranjo-Hernandez, "Design and implementation of a distributed fall detection system personal server," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 874–881, Nov. 2009.
- [5] 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 Trans. Biomed. Eng.*, vol. 56, no. 12, pp. 2858–2867, Dec. 2009.
- [6] Y. Li, K. Ho, and M. Popescu, "A microphone array system for automatic fall detection," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 2, pp. 1291–1301, May 2012.
- [7] 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 Trans. Inf. Technol. Biomed.*, vol. 15, no. 2, pp. 290–300, Mar. 2011.
- [8] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He, "Recognizing falls from silhouettes," in *Proc. 28th IEEE EMBS Annu. Int. Conf.*, New York, NY, USA, 2006, pp. 6388–6391.
- [9] M. Belshaw, B. Taati, J. Snoek, and A. Mihailidis, "Towards a single sensor passive solution for automated fall detection," in *Proc. 33rd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Boston, MA, USA, 2011, pp. 1773–1776.
- [10] M. Yu, A. Rhuma, S. Naqvi, J. Chambers, and L. Wang, "Posture recognition based fall detection system for monitoring an elderly person in a smart home environment," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 6, pp. 1274–1286, Nov. 2012.
- [11] N. Thome, S. Miguët, and S. Ambellouis, "A real-time, multiview fall detection system: A LHMM-based approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1522–1532, Nov. 2008.
- [12] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
- [13] H. Nait-Charif and S. McKenna, "Activity summarisation and fall detection in a supportive home environment," in *Proc. 17th Int. Conf. Pattern Recognit.*, 2004, pp. 323–326.
- [14] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 5, pp. 611–622, May 2011.
- [15] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using code-book model," *Real-Time Imag.*, vol. 11, no. 3, pp. 172–185, 2005.
- [16] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.
- [17] T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in presented at the IEEE Frame-Rate Appl. Workshop, Corfu, Greece, 1999.
- [18] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," presented at the Int. Conf. Comput. Vis. Pattern Recognit., Fort Collins, CO, USA, 1999.
- [19] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in presented at the *Eur. Conf. Comput. Vis.*, Dublin, Ireland, 2000.
- [20] R. Gonzalez, *Digital Image Processing*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2008.
- [21] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," presented at the 21st Int. Conf. Adv. Inf. Network. Appl. Workshop, Niagara Falls, ON, Canada, 2007.
- [22] C. Juang and C. Chang, "Human body posture classification by a neural fuzzy network and home care system application," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 37, no. 6, pp. 984–994, Nov. 2007.
- [23] B. Scholkopf, J. Platt, J. Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.
- [24] L. Cheng, M. Gong, D. Schuurmans, and T. Caelli, "Real-time discriminative background subtraction," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1401–1414, May 2011.
- [25] J. Kivinen, A. Smola, and R. Williamson, "Online learning with kernels," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.
- [26] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.
- [27] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.
- [28] A. Chan and N. Vasconcelos, "Counting people with low-level features and Bayesian regression," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2160–2177, Apr. 2012.
- [29] Y. Hou and G. Pang, "People counting and human detection in a challenging situation," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 1, pp. 24–33, Jan. 2011.
- [30] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1747–1761, Oct. 2009.
- [31] F. Lecumberry, A. Pardo, and G. Sapiro, "Simultaneous object classification and segmentation with high-order multiple shape models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 3, pp. 625–635, Mar. 2010.
- [32] Y. Zhang, *Advances in Image and Video Segmentation*. Hershey, PA, USA: IGI Global, 2006.



**Miao Yu** was born in China in 1986. He received the B.Sc. degree electronic and information engineering from the Shandong University of Science and Technology, Qingdao, China, in 2003. He then received the M.Sc. (with Distinction) in digital communication systems in 2008, together with the best student award, from the Department of Electronic and Electrical Engineering, Loughborough University, Loughborough, U.K., where he also received the Ph.D. degree in the area of fall detection for the elderly by exploiting video information.



**Yuanzhang Yu** was born in China in 1957. He received the B.Sc. and M.Sc. degrees in chemistry from Shandong University, Jinan, China, in 1982 and 1986, respectively.

From 1986 to 2006, he was with the Research Institute of Chinese Petroleum and Chemical Corporation as a Senior Engineer. Since 2006, he has been a Professor in the Department of Chemical Engineering, Shandong University of Technology, Zibo, China.



**Adel Rhuma** was born in Libya in 1971. He received the B.Sc. degree in computer engineering in 1998 from the Engineering Academy, Tajura, Libya, and the M.Sc. degree in remote sensing image processing and application from the University of Dundee, Dundee, U.K. He is currently working toward the Ph.D. degree in the area of fall detection for the elderly by exploiting video information.

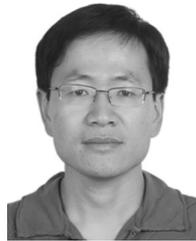


**Syed Mohsen Raza Naqvi** (S'07–M'09) received the B.Eng. (First Class) degree in industrial electronics engineering from the IIEE/NED University of Engineering and Technology, Karachi, Pakistan, in 2001, and the Ph.D. degree in signal processing from Loughborough University, Loughborough, U.K., in 2009.

Before his postgraduate studies in the U.K., he worked for research and development in Pakistan from January 2002 to September 2005. He is currently a Lecturer in image and video processing in

the School of Electronic, Electrical and Systems Engineering, Loughborough University. Prior to this faculty position, from July 2009 to September 2012, he was a Postdoctoral Research Associate in the Engineering and Physical Sciences Research Council of the U.K.-funded projects. He has authored or coauthored around 45 research outputs with main focus on his research area of multimodal (audio–visual) speech processing and his research interests include nonlinear filtering, data fusion, and multitarget tracking.

Dr. Naqvi is a member of the IEEE Signal Processing Society. He was a TPC member of the 16th International Conference on Information FUSION.



**Liang Wang** (SM'09) received the B.Eng. and M.Eng. degrees from Anhui University, Hefei, China, in 1997 and 2000, respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, in 2004.

From 2004 to 2010, he was a Research Assistant at the 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. He is currently a Professor of the Hundred Talents Program at the National Lab of Pattern Recognition, Institute of Automation, CAS. His major research interests include machine learning, pattern recognition, and computer vision. He has published widely within highly ranked international journals such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and IEEE TRANSACTIONS ON IMAGE PROCESSING, and leading international conferences such as Conference on Computer Vision and Pattern Recognition, International Conference on Computer Vision, and Industrial Conference on Data Mining.

Prof. Wang is a member of the British Machine Vision Association. He is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART B: CYBERNETICS, *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. He received several honors and awards such as the Special Prize of the Presidential Scholarship of the CAS.



**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 the 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 coauthor of the books *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability* (New York, NY, USA: Wiley, 2001) and *EEG Signal Processing* (New York, NY, USA: 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 (IIEE). He was the Technical Program Chair of the 15th International Conference on Digital Signal Processing 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, Prague, Czech Republic. He received 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. He 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).