Automatic Monocular System for Human Fall Detection Based on Variations in Silhouette Area

Behzad Mirmahboub, Shadrokh Samavi, Nader Karimi, and Shahram Shirani*

***Abstract*—Population of old generation is growing in most countries. Many of these seniors are living alone at home. Falling is among the most dangerous events that often happen and may need immediate medical care. Automatic fall detection systems could help old people and patients to live independently. Vision-based systems have advantage over wearable devices. These visual systems extract some features from video sequences and classify fall and normal activities. These features usually depend on camera's view direction. Using several cameras to solve this problem increases the complexity of the final system. In this paper, we propose to use variations in silhouette area that are obtained from only one camera. We use a simple background separation method to find the silhouette. We show that the proposed feature is view invariant. Extracted feature is fed into a support vector machine for classification. Simulation of the proposed method using a publicly available dataset shows promising results.**

***Index Terms*—Classification, fall detection, silhouette area, view invariant, visual surveillance.**

I. Introduction

HOPE for life is an important factor in modern societies. Improvements in human life conditions and better healthcare systems have increased the average age of the population in developed countries. The result is that the number of older people is increasing. Changes in human life style force many of these older people to stay home alone apart from their relatives. According to statistics, falling on the ground is one of the most important dangers that threaten the life of these lonely people [1]–[9], [11], [12], [15]–[18]. Falling can cause severe injuries that need immediate medical care. But if a fallen person is unconscious and unable to call for help it can lead to irreversible maim and even death. Falling not only physically hurts the elderly people but also harms them mentally; because the fear of falling will prevent their normal activities.

Automatic fall detection systems help the elderly people and patients to live independently without attendants. Such devices reduce burden on healthcare system since there is no need for full time nurses. Therefore, there is a big potential market for fall detection devices. Commercial systems are mainly divided into wearable and ambience devices [1].

Wearable devises use sensors such as micro switches, spirit levels, accelerometers, and gyroscopes that are embedded in garments or walking sticks. These sensors can detect changes in posture or motion of the wearer. If an abnormal change in posture is detected an alarm is sounded. Wearable devices are simple and cheap but they disturb the user's normal life. As a result many people are reluctant or forget to use them [1].

Ambience devices use vibration or pressure sensors that are installed in the floor or under the bed. They can detect the presence and position of the person. These devices are cheap and do not disturb the user; but they produce a lot of false alarms. In order to reduce false alarms some research works proposed to integrate information from different sensors [2]–[4].

To overcome the shortcomings of the previously mentioned systems, several attempts have been done in recent years to use camera-based systems [5]–[9]. Visual systems have the advantage of not disturbing the normal life of the user but the disadvantage of collecting information about one's private life. The individual's personal privacy prevents nurses from recording and watching original outputs of these visual systems. Therefore, automatic detection systems are needed. These systems extract some features from video frames and make decisions based on them [5]–[13].

In this paper, a fall detection system is proposed that uses silhouette area as a feature. The quality and accuracy of the silhouette extraction is not of importance in our method that makes it advantageous over other silhouette-based schemes. Only one camera is required in each room. The direction of the movement of the person with respect to the camera does not affect the accuracy of the proposed system. We describe two simple background separation methods and demonstrate that silhouette area of a person contains characteristics suitable for detection of fall. Then, a mathematical analysis to confirm the relation between silhouette area and a fall event is presented. We introduce two features that are obtained from silhouette area and claim that they are suitable for fall detection. In this paper, we also present a brief introduction to support vector machines (SVMs) and propose a general framework for combining several SVMs. Then, a dataset and a validation method are introduced. Classification is done separately based on the two proposed features and a combined system to improve the classification results is proposed.

The rest of this paper is organized in the following manner. Section II reviews some previous works in vision-based fall detection systems. Section III describes two background

Manuscript received May 15, 2012; revised July 31, 2012 and September 20, 2012; accepted October 12, 2012. Date of publication November 20, 2012; date of current version January 16, 2013. *Asterisk indicates corresponding author.*

B. Mirmahboub, S. Samavi, and N. Karimi are with the Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran (e-mail: mirmahboub@yahoo.com; samavi1996@yahoo.com; nader_karimi_80@yahoo.com).

*S. Shirani is with the Department of Electrical & Computer Engineering, McMaster University, Hamilton, ON L8S4L8, Canada (e-mail: shirani@mcmaster.ca).

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/TBME.2012.2228262

separation methods that are used to find silhouette. In Section IV, we introduce silhouette area as a new feature that is robust to view point. Section V describes SVM classifiers. Section VI is dedicated to the experimental results of our proposed system. Finally in Section VII concluding remarks are offered.

II. Camera-Based Fall Detection

A. *Single-Camera Methods*

A number of camera-based fall detection systems use only a single camera for this purpose. Miaou *et al.* [5] used the video from an omnicamera mounted on ceiling. It compares an aspect ratio of bounding box of person against a threshold in consecutive frames and tries identifying fall events. Different people may need different thresholds. Toreyin *et al.* [6] used periodic nature of aspect ratio of the bounding box in walking. They removed stationary component of this signal using wavelet transform. Two Hidden Markov Models (HMMs), one for walking and another for falling, are trained with the high frequency part of the signal. In a classification step of [6], decision is made based on which a model produces higher probability. Audio signals are used in similar manner to differentiate between falling and sitting events. Qian *et al.* [7] defined two bounding boxes on a person's silhouette. The larger box encompasses the whole body and the smaller one is fitted on the lower part of the legs. Height of the big box is a measure of standing or sitting, while the variations of the smaller box can separate standing from walking. In [7], cascaded support vector machine (CSVM) is used for classification.

B. *Multicamera Methods*

The methods that were mentioned in the previous section use a single camera and assume that the person moves parallel to the image plane. If the person moves toward the camera, not much change is revealed from his/her bounding boxes. This major assumption, for the position of camera or movement of people, is impractical in real life. To solve this problem, several research works have attempted to use multicamera systems. Rougier *et al.* [8] defined costs and calculated them for consecutive frames using shape matching. These costs are criteria for amounts of deformations and are fed to a Gaussian mixture model for classification. Thereafter, the final result is obtained by voting among the four cameras. Auvinet *et al.* [9] reconstructed 3-D volume of the person from eight cameras using calibration information. If a big portion of the body volume is near the ground for a period of time they classify it as a fall.

C. *View-Invariant Methods*

Multicamera systems are successful in correctly detecting falls, but they are very complicated and therefore are less reliable to be implemented in real-life environments. To build a cost-effective commercial system we probably need a simpler yet accurate method. Several recent research works have tried to define view-invariant features from one single camera.

Ji and Liu [10] reviewed view-invariant methods for pose estimation and action recognition. According to the selection of a human prior model they divided pose estimation methods to three groups of 3-D model based, 3-D model free, and example based. Also action recognition methods can be separated to template-matching-based methods and state-space approaches. Fall detection is a subset of action recognition that only one action of falling is of important for us. Furthermore, all computations for fall detection must be done in real time.

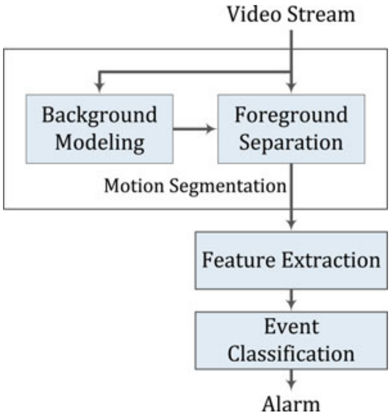

Fig. 1. General framework for automatic visual fall detection.

Shoaib *et al.* [11] detected the location of head and foot in each frame by ellipse matching. In the learning phase, the image is divided into equal-sized blocks. These blocks are categorized as the head blocks, foot blocks, and neutral blocks, according to the number of heads or feet that pass through them. A Gaussian distribution is assigned to each foot block that determines mean and variance of head vertical positions relative to that foot block. In the detection phase, after finding the head and foot positions, the difference between the current position of head and its expected position is calculated. This difference is compared with upper and lower bounds to distinguish between walking, bending, and falling.

Htike *et al.* [12] normalized silhouette to 128 × 128 pixels and selected 230 points on the outline of the silhouette. All pairwise distances between these points are calculated and a distance histogram is built. This is called chord distribution and is used to present 2-D poses. Various poses are generated using a computer program and are used to train 20 probabilistic pose models. In the classification phase, each model produces a probability for the input test image. The probabilities of all models form a pose excitation vector. This vector is normalized and is fed to a fuzzy-state HMM for classification.

D. *Our Proposed Method*

In most of the existing fall detection systems we can distinct three major steps as shown in Fig. 1. The algorithm starts with motion segmentation, which is the separation of a stationary background from a moving foreground. Then, some features are extracted from the foreground sequence where classification is done based on these features. Event classifier could be a simple thresholding or a complicated structure that needs to be trained beforehand [13].

The view-invariant methods that are aforementioned have achieved good results on their own datasets. But they are complicated systems that do not take into account the real characteristics of falling. For example, they did not mention the difference between falling and sitting down. We are proposing a single camera system that specially pays attention to the characteristics of falling and produces accurate results on a publicly available dataset. Our system is simple and accurate enough to make it a candidate for fabrication.

In this paper, we consider a single camera system with a fixed point of view. In our method, there is no need for the person to be viewed at a specific angle to distinguish his/her fall. We obtain silhouette of the person from two different background separation methods and use area of the silhouette as a robust feature with respect to view direction. Classification in our method is done using SVMs based on features extracted from silhouette area.

III. Background Separation

Separating moving objects in a scene, known as motion segmentation, is the first step in most automatic visual surveillance. Many methods exist for this purpose [14]. Each method is suitable for a special application. In this section, we mention two different methods for background separation. Each method produces a different silhouette with a different area.

A. Running Average Method

It is assumed that indoor environments do not change rapidly. Therefore, for segmentation we use running Gaussian average because it has low computational requirements and we do not need very high accuracy [14].

We keep one intensity value for each pixel in the background and update that as in

$$b_t = \gamma I_t + (1 - \gamma) b_{t-1} \tag{1}$$

where b_t is the background value and I_t is the pixel's current value for frame t. Both of them are integers between 0 and 255. Also, γ is a real number between 0 and 1 that determines the updating speed. At each frame t if $|I_t - b_t|$ is more than a threshold T_B, then the pixel belongs to foreground. After subtraction we perform an erosion step, using a disk structuring element with radius of 1, to remove the noise.

Output of the background separation algorithm is a binary image called "silhouette." In this image, black pixels correspond to stationary parts and white pixels show moving segments.

B. Modified Running Average

Equation (1) has a drawback that it unnecessarily updates all pixels. As a result an unwanted "shadow" appears in the silhouette that follows the moving object. Furthermore, if the object does not move for a while, the silhouette area gradually decreases because the background is updated and the stationary object becomes a part of the background. To suppress this drawback a modified version of (1) is usually used that prevents updating of the foreground [14]

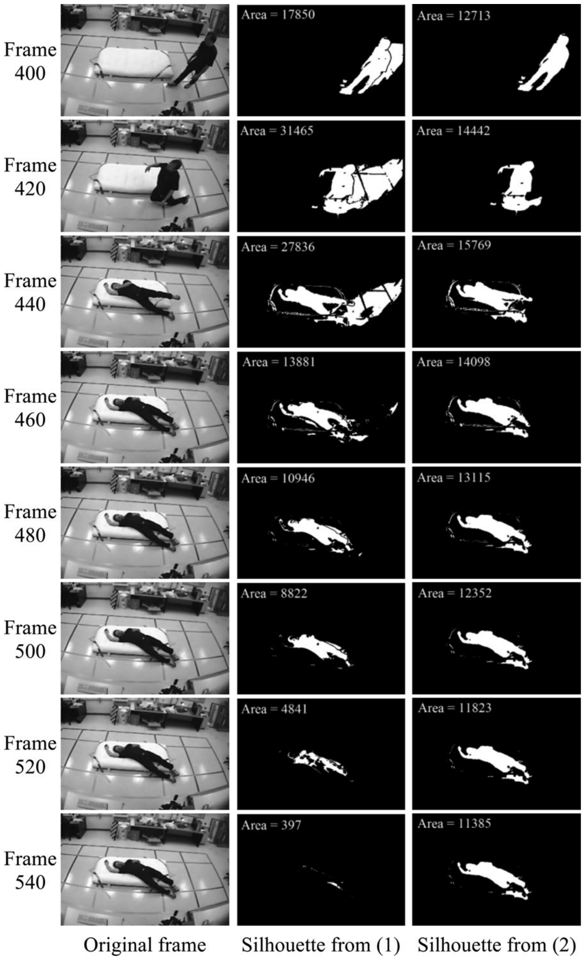

Fig. 2. Contrast between silhouettes that are produced from two background separation methods.

$$b_t = M b_{t-1} + (1 - M)[\gamma I_t + (1 - \gamma) b_{t-1}] \tag{2}$$

where binary value M is 1 for the foreground and is 0 elsewhere. The results of applying these two methods on a video in the used dataset are shown in Fig. 2. In this figure, each row shows an original frame along with two silhouettes obtained from (1) and (2).

IV. Feature Extraction

The second step in most visual fall detection systems, based on the block diagram of Fig. 1, is the feature extraction step. In this section, we first present our observations from the silhouettes produced from (1). These observations motivate us to select variations of silhouette area as a suitable feature for fall detection. Then, we mathematically show that these observations have generality and are not limited to the observed samples.

A. Observations

Aspect ratio and orientation of silhouette are widely used as features in fall detection algorithms [15], [16]. But they are not capable to distinguish fall events from normal activities

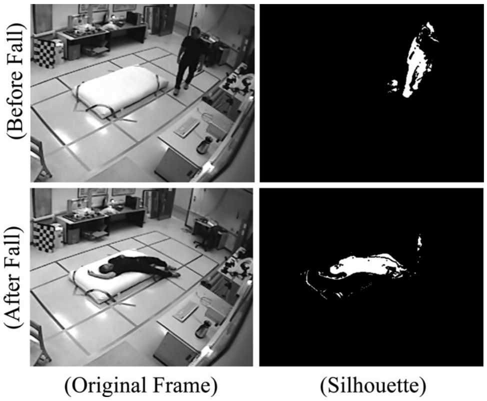

Fig. 3. Original frame and silhouette before and after a fall when falling is parallel to the image plane.

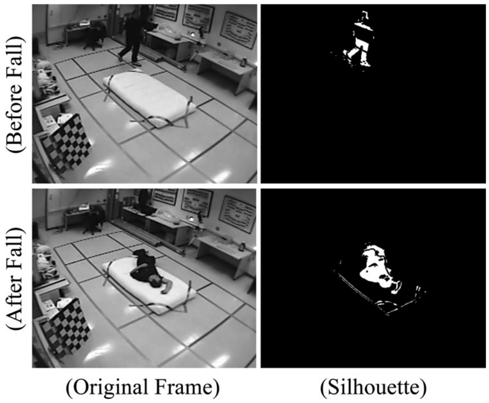

Fig. 4. Original frame and silhouette before and after fall when falling is perpendicular to the image plane.

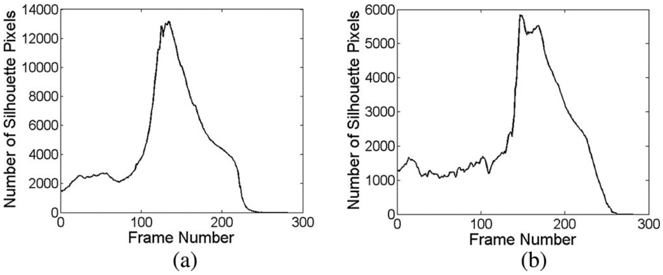

Fig. 5. Variations of silhouette area corresponding to falling occurred in (a) Fig. 3 and (b) Fig. 4.

when the person moves toward camera. In addition they cannot differentiate between falling and normal sitting down [17].

Fall events have special characteristics [18]. During the fall, vertical speed of the person increases. Falling person usually hits the environment strongly. After the fall a seriously injured person remains motionless on the ground. Any good feature to detect the fall should take these characteristics into consideration.

Motion speed of a person could help fall detection but measure of this factor could be erroneous [8]. In Fig. 2, we observed that using (1) for background subtraction results a shadow that follows the moving person. We found out that the size of this shadow is a measure of the motion speed [17]. The faster a person moves, the bigger shadow will follow him. Therefore, sudden increase in the silhouette area is an indication of rapid motion during the fall.

As the sample frames in Fig. 2 show the impact from a person falling would shake the surrounding environment. These vibrations in the environment are sensed as moving objects and are added to the silhouette area. Also, if the fallen person remains stationary, its silhouette from (1) is faded out as time passes. Hence, the shrinkage of the silhouette area is another characteristic of falling.

We consider one scenario of falling event that is captured from two different directions. Fig. 3 shows the original frame and its silhouette from (2) in one frame before falling and one frame after the fall when the person moves parallel to the image plane. Fig. 4 shows the same situations, captured from a different angle, when the direction of motion is perpendicular to the image plane.

Fig. 5 shows the variations in the silhouette area in case of a fall observed from two different viewpoints. In contrast to features such as aspect ratio or orientation, variation in the silhouette area, as a feature, follows almost the same pattern when the motion is parallel or perpendicular to the image plane. In either case, the silhouette area increases rapidly to a maximum value and then decreases to zero. This shows that variation in the silhouette area is robust to view direction.

The previous observations motivate us to use the variation of silhouette area produced from (1) as a proper feature to discriminate fall events from normal activities

B. *Mathematical Reasoning*

Silhouette from (1) will fade out for stationary person and also a shadow will follow a moving person. Here, we aim to find a mathematical basis for these two phenomena. Therefore, we first find the probability of a pixel being seen in a silhouette, after a certain number of frames, while the person remains motionless in order to confirm the fading of the silhouette.

Assume that an object appears in a scene and stays motionless. Then, the intensity of a pixel in the object in the current frame (I_t) is constant I. Equation (1) is a recursive formula that updates background pixels behind the stationary object. If we extend the b_{t-1} term, we can find a relationship between b_t and b_0 as in (3) where b_0 is the value of a background pixel for an empty scene, without any moving object, at the start of the algorithm. Also, b_t is the value of that background pixel after passage of t frames

$$b_t = (1-(1-\gamma)^t)I + (1-\gamma)^t b_0. \qquad (3)$$

Foreground pixels should satisfy $|I - b_t| > T_B$. By substitution of b_t from (3) into this condition we obtain

$$|-(1-\gamma)^t I + (1-\gamma)^t b_0| > T_B. \qquad (4)$$

We assume that I and b_0 are random variables between 0 and G that is the largest gray scale value in the image. We want to

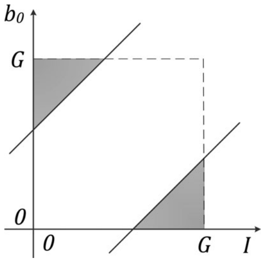

Fig. 6. Two triangles correspond to visible pixels based on (4).

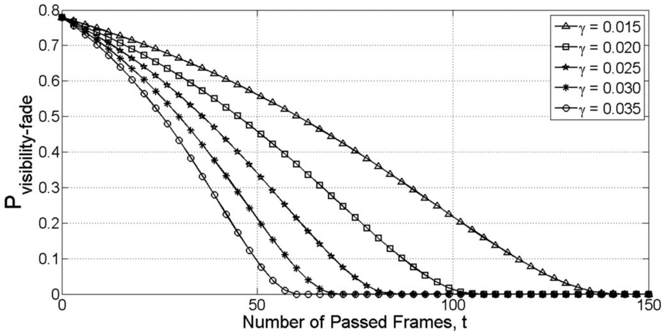

Fig. 7. Visibility probability of a stationary pixel as a function of passed frames.

find the probability of a pixel to be seen in the silhouette after t frames. This can be done by finding the portion of pixels that satisfy (4) using a simple linear programming. Fig. 6 shows the sample space for I and b_0. Pixels that satisfy (4) are shown as two equal triangles. Dimensions of these triangles are found by cross sections of the inequality (4) with boundaries. Then, the probability of one pixel that satisfies (4) is calculated by dividing the areas of the two triangles by the area of the square sample space

$$P_{\text{visibility\_fade}} = \left(1 - \frac{T_B}{G(1-\gamma)^t}\right)^2. \tag{5}$$

Fig. 7 plots $P_{\text{visibility\_fade}}$ for $T_B = 30$, $G = 255$ and different values of γ. The visibility probability (probability of a pixel being part of the silhouette) of one stationary pixel decreases as time passes. When this probability is multiplied by the number of pixels, the number of visible pixels or the silhouette area is obtained. As we see the area of a stationary object shrinks to zero during time. Higher values of γ lead to faster shrinking rate. This verifies our observation of shrinkage of a silhouette area after a person falls and remains stationary.

We have observed that the area of silhouette increases as the speed of the moving person or object increases. To confirm this analytically, here, we want to calculate the visibility probability of one pixel when it comes out from behind of a moving object. Consider an object with the length of L pixels that moves with a speed of f pixels per frame as shown in Fig. 8.

Value of a background pixel is initially b_0. While a point in the background is covered with the moving object, that location is updated according to (1) and its value is changed to b_t. Number of updates is $t = L/f$.

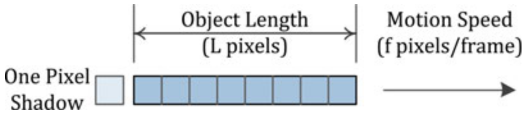

Fig. 8. Moving object with the length of L pixels and a shadow pixel that follows it.

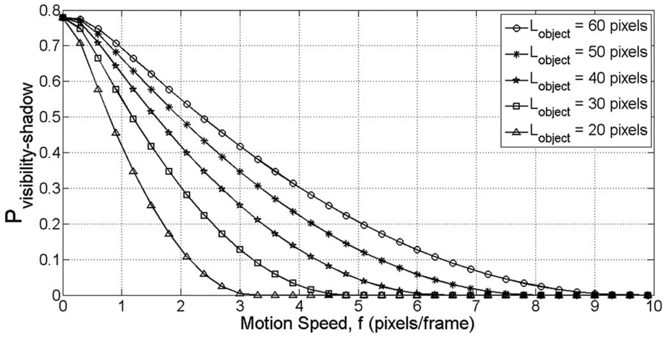

Fig. 9. Visibility probability of one shadow pixel that immediately follows a moving object.

Fig. 8 shows a background pixel immediately after emerging from behind of the moving object. Value of this pixel should be compared with the original background. This pixel is visible if satisfies $|b_0 - b_t| > T_B$. We can assume that t, number of frames elapsed during passing object, is small enough that the value of the original background stays unchanged. By substitution of b_t from (3) into $|b_0 - b_t| > T_B$ we obtain

$$|-(1-(1-\gamma)^t)I + (1-(1-\gamma)^t)b_0| > T_B. \tag{6}$$

We want to calculate the visibility probability of one pixel when it comes out from behind of a moving object. Such a pixel should satisfies (6). The probability is calculated similar to the previous discussion for Fig. 6 and we obtain

$$P_{\text{visibility\_shadow}} = \left(1 - \frac{T_B}{G(1-(1-\gamma)^{L/f})}\right)^2. \tag{7}$$

Fig. 9 plots $P_{\text{visibility\_shadow}}$ for $T_B = 30$, $G = 255$, $\gamma = 0.02$, and various object length. Visibility of shadow depends on the object size and its speed. Bigger objects impose more updates on the background pixels and increase the visibility probability, $P_{\text{visibility\_shadow}}$. In turn, higher speeds do not let the background pixels to be updated enough times and decrease the visibility probability.

Fig. 9 shows the visibility probability of one background pixel that is just released from a moving object. Shadow length depends on the number of visible pixels. In another words, given a visibility probability of a pixel, we should find how many frames it takes for that pixel to fade out. Using (5), we can find the number of frames t_p that takes for one pixel to reach a given probability $P_{\text{visibility\_shadow}}$

$$t_p = \log\left(\frac{T_B}{G(1-\sqrt{P_{\text{visibility\_shadow}}})}\right) / \log(1-\gamma). \tag{8}$$

Number of frames that takes for that pixel to disappear is $t_{\max} - t_p$, where $t_{\max}$ is the maximum number of frames that

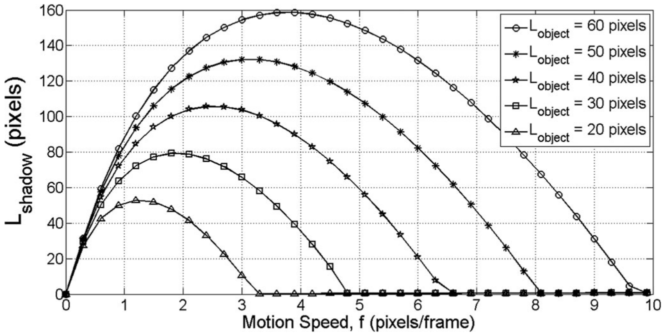

Fig. 10. Shadow length behind moving an object for various object lengths.

are required for all probabilities to become zero in Fig. 7. During that time the moving object will go forward at the speed of f pixels per frame and all pixels behind it will be visible. Then, the length of the shadow behind the moving object is calculated as (9)

$$L_{\text{shadow}} = (t_{\max} - t_p) \times f. \tag{9}$$

Fig. 10 plots L_{shadow} for $T_B = 30$, $G = 255$, $\gamma = 0.02$, and various object lengths L. As expected, bigger objects produce bigger shadows. The noticeable point in Fig. 10 is that as the motion speed increases, at first the shadow length increases up to a maximum value then it decreases to zero. No shadow will form behind the object if it remains motionless or if the object moves very fast. Whereas we defined the speed as pixels per frame, low frame-rate videos cause high motion speed and result in no shadow.

Fig. 10 shows that if the motion speed does not exceed a certain value, size of the shadow increases with the speed. In fact, in normal daily activities of a person, motion speeds are within the first region of graphs of Fig. 10 where an increase in the speed causes an increase in the area of the shadow. This confirms our observation about the dependency of silhouette area to the motion speed.

In this section, we mathematically confirmed that our observation about shrinkage of a silhouette area after a stationary period of a moving object and also our observation of the expansion of a silhouette with increasing speed of a moving object are correct. Hence, we can generalize these observations for the captured video sequences of a single camera.

C. Selected Features

We compute silhouette areas produced from (1) in p consecutive frames to form a p dimensional feature vector. Size of p is selected such that it covers all phases of falling including rapid movement during the fall and stationary period afterward.

As we see in Fig. 2, silhouette that is produced from (2) is almost the exact silhouette of the person and does not show the properties of a fall; such as rapid motion before the impact and stationary period after that. But we can expect the difference between silhouette areas from (1) and (2) to be another good feature for fall detection. Therefore, we introduce this difference as a new feature. We normalize the two area vectors to remove their scale difference and then subtract them to form the new feature vector. These two feature vectors are used in the classification step.

TABLE I
FAMOUS KERNEL FUNCTIONS

Linear	$K(x_i, x_j) = x_i^T x_j$
Polynomial of degree m	$K(x_i, x_j) = (x_i^T x_j + 1)^m$
Radial Basis Function (RBF)	$K(x_i, x_j) = e^{\frac{-\lVert x_i - x_j \rVert^2}{\sigma^2}}$

V. CLASSIFICATION

In this section, a brief introduction on the SVM classifier is first presented. Then, we propose a combined system that benefits from several features.

A. Support Vector Machines

SVMs and their extensions, often called kernel-based methods have been studied extensively and applied to various pattern classification problems [19], [20]. SVMs outperform other classifiers when the number of training data is small. SVMs are trained for two-class problems to find a direct decision function that maximizes the generalization capability.

Suppose we have N training samples of the form $\{x_i, y_i\}_{i=1}^N$ where $x_i \in \Re^p$ is p-dimensional input vector and y_i is scalar output that is labeled with 1 and -1. The goal is to find an optimal hyperplane in the form $D(x) = w^T x + b$ that has maximum margin from the two classes. Slack variables ξ_i are defined that let some of the samples to be located inside the margin. The problem is solved in [19] by maximizing

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \left(K(x_i, x_j) + \frac{\delta_{ij}}{C} \right). \tag{10}$$

This is an optimization problem called dual form of $L2$ soft margin and is subject to $\sum_{i=1}^{M} \alpha_i y_i = 0$ and $\alpha_i \geq 0$ where α_i are called Lagrange multipliers. Here, δ_{ij} is Kronecker's delta function, in which $\delta_{ij} = 1$ for $i = j$ and 0 otherwise. Also C is the margin parameter that determines how many samples are tolerable to be inside of the margin. Higher values of C result in lower error rates and narrower margins. $K(x_i, x_j)$ is the kernel function that maps input sample vectors to higher dimensions called feature space. This is done to increase the separability of samples. Some well-known kernel functions are listed in Table I.

Maximization of (10) requires solving a quadratic programming problem that finds a global maxima. After optimization, only samples inside the margin boundary will have nonzero Lagrange multipliers. These samples are called support vectors (SVs) and are used to build the final decision function. If S is a set of SV indices, the optimal hyperplane is obtained as

$$D(x) = \sum_{i \in S} \alpha_i y_i K(x_i, x) + b \tag{11}$$

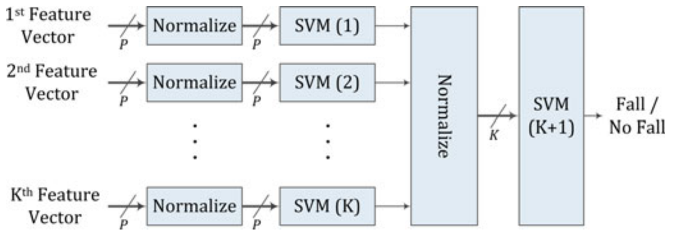

Fig. 11. Proposed combined classifier for fall detection.

where x is a test sample vector and the hyperplane offset b is calculated as

$$b = \frac{1}{|S|}\sum_{j\in S}\left(y_j - \sum_{i\in S}\alpha_i y_i\left(K\left(x_i, x_j\right) + \frac{\delta_{ij}}{C}\right)\right). \quad (12)$$

In the training phase, we assign label 1 and -1 to fall event and normal activity, respectively, and find α_i from (10). In the classification phase, we use (11) to find a label for a testing sample x. If $D\left(x\right) > 0$ we classify x as a fall event, otherwise it is classified as a normal activity.

B. Proposed System of Combined Features

If x is a p-dimensional input feature vector sent to a SVM, it should be normalized as x/x before entering the classifier. SVM is a classifier for two-class problems that produces a label that its sign determines the class. In order to use a mixture of different features, one simple method is concatenating several feature vectors to build a new long vector and giving that to SVM. But increasing the vector size will increase the complexity of the system. Also different features may need different kernel functions for good classification. The long feature vector may have physical incoherence and we may not be able to determine a suitable kernel for that. Hence, we propose a combined classifier as shown in Fig. 11.

If we have K feature vectors, each vector is fed into a separate SVM for classification. Output labels of these K SVMs will form a new K-dimensional feature vector that is fed to a final SVM to produce the ultimate decision about falling. Applying weight to each of these features would not improve the overall performance of the classifier. Using this proposed system we can clearly see whether adding a new feature would improve classification performance or not.

VI. Experimental Results

In this section, we first introduce our dataset and divide that into two groups. We tune our system with the samples in first group. Then, we use the samples in second group for validating the system.

A. Dataset

We used the fall dataset that is mentioned in [21] and [22]. The database consists of 24 scenarios. In each scenario, an actor plays a number of activities such as falling, sitting on a sofa, walking, pushing objects, etc. Each scenario is shot simultaneously with eight cameras, each from a different direction. All of the actions are performed by one person with different garment colors. We used all of the 736 actions of the dataset that are shot from different viewpoints as shown in Table II. Many of them include partial occlusion. A pseudofall is a none-alarming case, such as when the person falls on the ground but immediately stands up afterward.

TABLE II
Different Activities Depicted in Dataset

Type of activity			**Count**	**Label**
fall	with support	on hard surface	16	1
	w/o support	on mattress	72	1
		on hard surface	24	1
	start from sitting position (hard surface)		8	1
walking			192	-1
pseudo-fall			88	-1
sitting on	chair		8	-1
	armchair		32	-1
	sofa		40	-1
lying on the sofa			32	-1
standing up from sofa			16	-1
crouching			72	-1
dropping objects & crouching			8	-1
carrying objects	taking up		32	-1
	putting down		32	-1
	both		24	-1
taking off and hanging the coat			8	-1
sweeping			32	-1
Total			**736**	

Each activity is characterized with two-feature vectors as explained in Section IV-C. In the rest of the paper, for convenience, we use shorthand notations of Area_1 and Area_2 to refer to silhouette areas obtained from (1) and (2), respectively. Also we use Δ_{Area} to indicate area difference between (1) and (2). A fall event, followed by a stationary period, takes place in about 200 frames. These mentioned two features are computed in every frame. Hence, each of these feature vectors has 200 elements. To reduce complexity and also for noise removal, we choose one out of every five consecutive elements. The median of every five elements is chosen. Also, we normalize data to compensate for variations in distances of person from the camera and variations in people's heights. Therefore, each feature vector has 40 normalized elements.

B. System Tuning

In the background separation step, we empirically came up with an updating speed of $\gamma = 0.02$ and a constant foreground threshold of $T_B = 30$. Also, in the classification step we select a margin parameter of $C = 100$.

The following parameters are used to analyse the recognition results of the algorithm [18].

1) True positives (TP): number of falls detected correctly.
2) True negatives (TN): normal activities detected correctly.
3) False positives (FP): normal activities detected as fall.
4) False negatives (FN): number of falls detected as normal.
5) Sensitivity: ability to detect fall events

$$\mathrm{Se} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}). \quad (13)$$

TABLE III
HUMAN ACTIONS USED FOR CLASSIFIER TRAINING

Type of activity	**Count**	**Label**
Fall on mattress	64	1
Fall on hard surface	8	1
Walking	64	-1
Pseudo-fall	8	-1
Sitting on armchair	8	-1

TABLE IV
CLASSIFICATION RESULTS OF THE SYSTEM BASED ON Area_1 USING CROSS VALIDATION ON TRAINING DATA

Kernel	Linear	Polynomial (2^{nd} degree)	RBF ($\sigma = 1$)	RBF ($\sigma = 1.8$)	RBF ($\sigma = 3$)
TP	70	70	69	70	70
TN	77	79	78	79	75
FP	3	1	2	1	5
FN	2	2	3	2	2
Se(%)	97.22	97.22	95.83	97.22	97.22
Sp (%)	96.25	98.75	97.50	98.75	93.75
Ac (%)	96.71	98.03	96.71	98.03	95.39
Er (%)	3.29	1.97	3.29	1.97	4.61
SV	27	29	29	32	39

6) Specificity: ability to recognize normal activities

$$\mathrm{Sp} = \mathrm{TN}/(\mathrm{TN} + \mathrm{FP}). \tag{14}$$

7) Accuracy: correct classification rate

$$\mathrm{Ac} = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}). \tag{15}$$

8) Error rate: incorrect classification rate

$$\mathrm{Er} = (\mathrm{FP} + \mathrm{FN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}). \tag{16}$$

The entire simulations are implemented with MATLAB 7.10.0. In the used dataset, the complexity of samples (types of actions, occlusions, color of clothing, etc.) increases as the scenario's number increases. Hence, the first few scenarios are less complex and final scenarios are more complex ones. Generally, for training a classifier we should use good samples. Therefore, we use the 152 samples in the first nine scenarios to tune the system by finding suitable SVM parameters. The human actions that are in the first nine scenarios and are used for training are listed in Table III.

In order to determine suitable kernel functions for SVM, we use leave one record out cross validation. Therefore, we select one sample out of 152 for testing. Training of the classifier is performed with 151 remaining samples and values of TP, TN, FP, and FN are obtained for one testing sample. This algorithm is repeated 152 times until all samples are tested and their results are summed.

TABLE V
CLASSIFICATION RESULTS OF THE SYSTEM BASED ON Δ_{Area} USING CROSS VALIDATION ON TRAINING DATA

Kernel	Linear	Polynomial (2^{nd} degree)	RBF ($\sigma = 0.5$)	RBF ($\sigma = 0.7$)	RBF ($\sigma = 1$)
TP	72	72	72	72	72
TN	77	77	78	79	77
FP	3	3	2	1	3
FN	0	0	0	0	0
Se(%)	100.00	100.00	100.00	100	100.00
Sp (%)	96.25	96.25	97.50	98.75	96.25
Ac (%)	98.03	98.03	98.68	99.34	98.03
Er (%)	1.97	1.97	1.32	0.66	1.97
SV	16	16	81	21	17

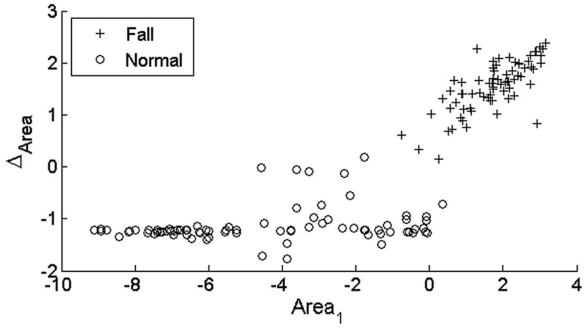

Fig. 12. Separability of 2-D samples that are formed by the outputs of Area_1 based classifier and Δ_{Area}-based classifier.

1) Single Feature System: Simulations are performed separately based on two proposed feature vectors of Area_1 and Δ_{Area} with linear kernel, polynomial kernel of second degree and radial basis function (RBF) kernels. The results are shown in Tables IV and V. We tried many radii for RBF, but only the results of the best one along with two others are listed for comparison. Using higher orders of polynomial kernel did not improve the classification performance and it may also cause overfitting. We also calculate the average number of SVs that each feature needs for classification. According to (11) the number of SVs affects the size of the optimal separating hyperplane. Our first priority for choosing a good kernel is its error rate. The second priority is the number of SVs that influence the complexity of the final system.

As we described in Section IV variations in the silhouette area from (1) can show some properties of falling. The best classification performance of Area_1 in Table IV is achieved with polynomial kernel that produces one FP and two FN with 29 SVs. Classification results based on Δ_{Area} in Table V reveal interesting results. Most kernel functions produce no FN. Best results are achieved using RBF kernel with radius of 0.7 that produces only one FP.

2) Two-Feature System: We observed that there is no sample that is classified incorrectly by both Area_1 and Δ_{Area}-based classifiers. This motivates us to combine these two features to improve the classification results. Each classifier outputs a label for an input feature vector. Fig. 12 plots the labels produced by the Δ_{Area}-based classifier versus the labels produced by the Area_1-based classifier. We clearly see that the samples are well separable in the 2-D space, while none of the classifiers alone can exactly separate the samples.

We use the formation of Fig. 11 to build the combined system based on the Area_1 and Δ_{Area} feature vectors. Labels produced by the two classifiers will form new 2-D samples. These new samples are fed to a third classifier. L1RO cross validation showed that third SVM can completely separate 2-D samples

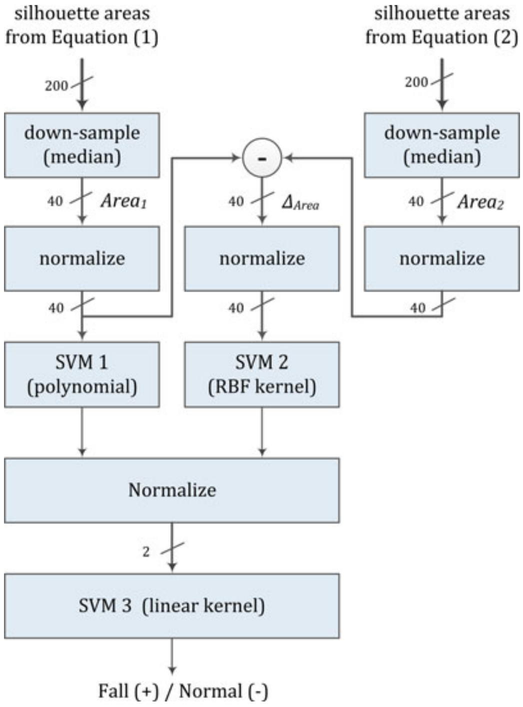

Fig. 13. Proposed system for automatic fall detection.

TABLE VI
SYSTEM CLASSIFICATION RESULTS FOR VALIDATION DATA

Classifier (feature)	SVM 1 ($Area_1$)	SVM 2 (Δ_{Area})	SVM 3 (Combined)
TP	48	48	48
TN	495	482	501
FP	41	54	35
FN	0	0	0
Se(%)	100	100	100
Sp (%)	92.35	89.93	93.47
Ac (%)	92.98	90.75	94.01
Er (%)	7.02	9.25	5.99
SV	29	21	5

using linear kernel and two SVs. The final proposed system for human fall detection is depicted in Fig. 13.

C. System Validation

After determining appropriate kernels for each SVM, we use all of the 152 samples in the first nine scenarios to train the system (to find α_i of (10)). Then, we tested the system with the other 584 samples. The classification results are shown in Table VI.

If we take into consideration all 736 samples, the error rate of the system will be 4.8%. All of the alarming fall events are correctly detected by our system and there are no false negative cases. Among the 35 false positives (false alarms), six sitting and 15 lying are not classified correctly. In these cases, the actor stays motionless after sitting or lying. Probably, every feature that takes characteristics of fall into consideration will have this shortcoming. One solution to this problem is defining inactivity zone in a furnished room. Usually, fall detection systems consider regions of a room such as bed or sofa as inactivity zones. If we define inactivity zones for the tested database our error rates would drastically reduce.

TABLE VII
COMPARISON OF CLASSIFICATION RESULTS FOR DIFFERENT FEATURES

Feature	Error Rate
Full Procrustes distance [8]	3.8 %
mean matching cost [8]	4.6 %
bounding box ratio [8]	43.4 %
2-D vertical velocity [8]	11.3 %
normalized 2-D vertical velocity [8]	22.7 %
Proposed silhouette area ($Area_1$ & Δ_{Area})	4.8 %

Furthermore, one FP happens when the actor enters the field of view of camera and quickly recedes from the camera. Also one FP happens when the actor approaches the camera and stands motionless. Three FP happen when the actor puts down objects and also nine pseudo falls are wrongly classified as falls.

D. Comparison

Rougier *et al.* [8] used two state-of-the-art features called "Full Procrustes distance" and "mean matching cost" with voting between 4 cameras and reported classification performance using dataset of [22]. We have used the same dataset. The comparison between our proposed features and a number of other features as reported in [8] is shown in Table VII.

We see that our proposed feature is better than three of the benchmarks that [8] was compared with. We also achieved comparable results with respect to the two complicated features of "full procrustes" and "mean matching cost." These features are obtained through voting among four cameras, as compared with our proposed single view feature. It should be mentioned that our proposed method, which has low computational burden, has the potential of hardware implementation inside a camera. The output of such a camera would be alarms and not video frames. With a complex algorithm, original videos may need to be transferred to a remote computer for software processing and voting. Any video frame that comes out of a surveillance camera could put personal privacy at stake.

VII. CONCLUSION

Serious dangers threaten the growing population of elderly people that live alone in their homes. Falling is a major hazard that requires automatic surveillance systems to detect potential hazardous situations to produce an alarm. These kinds of systems extract features from video sequences and decide base on them. In this paper, we exploited a drawback in one of the simplest background subtraction methods. We proposed to use variations in the silhouette area as a feature that is robust to view point. We showed mathematically and experimentally that variation in the silhouette area is a good indicator of rapid motion during a fall event, impact to the surrounding environment, and stationary period after that. Generally, in vision-based algorithms for fall detection, the extracted features strongly depend on the quality of the background separation process. Researchers try to obtain the moving foreground, as accurately as possible, using complicated and computationally demanding methods. The advantage of our system is that we do not need to find an accurate foreground at all, but we use its inaccuracy to reach our goal. This makes the final system simple and a perfect candidate for commercial implementation. We proposed to build a combined system of SVMs to use the benefits of silhouette areas from two different background subtraction methods. We tested our proposed system on a publicly available dataset and compared our results with recent works that have used the same dataset. We achieved error rate of about 4.8% that outperforms commonly used features such as "bounding box ratio" and "vertical velocity." Also, the proposed feature is comparable with very complex features of "Full Procrustes distance" and "mean matching cost." Our method has the advantage of being very simple and requires only one camera. Furthermore, our proposed method has the potential of hardware implementation inside a camera. The output of such a camera would be alarms and not video frames that guarantee personal privacy.

References

[1] X. Yu, "Approaches and principles of fall detection for elderly and patient," in *Proc. 10th Int. Conf. e-health Netw., Appl. Serv.*, Jul. 7–9, 2008, pp. 42–47.

[2] C. Doukas and I. Maglogiannis, "Emergency fall incidents detection in assisted living environments utilizing motion, sound, and visual perceptual components," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 2, pp. 277–289, Mar. 2011.

[3] F. Bianchi, S. Redmond, M. Narayanan, S. Cerutti, and N. Lovell, "Barometric pressure and triaxial accelerometry-based falls event detection," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 6, pp. 619–627, Dec. 2010.

[4] 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.

[5] S. G. Miaou, P. H. Sung, and C. Y. Huang, "A customized human fall detection system using omni-camera images and personal information," in *Proc. 1st Transdisciplinary Conf. Distrib. Diagnosis Home Healthcare*, 2006, pp. 39–42.

[6] B. U. Toreyin, Y. Dedeoglu, and A. E. Cetin, "HMM based falling person detection using both audio and video," in *IEEE 14th Signal Proc. Comm. Apps.*, Apr. 17–19, 2006, pp. 1–4.

[7] H. Qian, Y. Mao, W. Xiang, and Z. Wang, "Home environment fall detection system based on a cascaded multi-SVM classifier," in *Proc. 10th Int. Conf. Control, Autom., Rob. Vision*, 2008, pp. 1567–1572.

[8] 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.

[9] E. Auvinet, F. Multon, A. St-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.

[10] X. Ji and H. Liu, "Advances in view-invariant human motion analysis: A review," *IEEE Trans. Syst. Man Cybern., Part C: Appl. Rev.*, vol. 40, no. 1, pp. 13–24, Jan. 2010.

[11] M. Shoaib, R. Dragon, and J. Ostermann, "View-invariant fall detection for elderly in real home environment," in *Proc. Fourth Pacific-Rim Symp. Image Video Technol.*, Nov. 14–17, 2010, pp. 52–57.

[12] Z. Z. Htike, S. Egerton, and Y. C. Kuang, "A monocular view-invariant fall detection system for the elderly in assisted home environments," in *Proc. 7th Int. Conf. Intell. Environ.*, Jul. 25–28, 2011, pp. 40–46.

[13] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev.*, vol. 34, no. 3, pp. 334–352, Aug. 2004.

[14] M. Piccardi, "Background subtraction techniques: A review," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Oct. 10–13, 2004, vol. 4, pp. 3099–3104.

[15] S. Alaliyat, "Video-based fall detection in elderly's houses," Master's Thesis, Dept. of Computer Science and Media Technol., Gjøvik Univ. College, Gjøvik, Norway, 2008.

[16] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," in *Proc. 21st Int. Conf. Adv. Inf. Network. Appl. Workshops*, May 21–23, 2007, vol. 2, pp. 875–880.

[17] B. Mirmahboub, S. Samavi, N. Karimi, and S. Shirani, "View-invariant fall detection system based on silhouette area and orientation," in *Proc. Int. Conf. Multimedia Expo*, Jul. 9–13, 2012, pp. 176–181.

[18] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. O. Laighin, V. Rialle, and J. E. Lundy, "Fall detection - Principles and methods," in *Proc. IEEE 29th Annu. Int. Conf. Eng. Med. Biol. Soc*, Aug. 22–26, 2007, pp. 1663–1666.

[19] Shigeo Abe, *Support Vector Machines for Pattern Classification*, 2nd ed. London, U.K.: Springer-Verlag, 2010.

[20] A. Fleury, M. Vacher, and N. Noury, "SVM-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 274–283, Mar. 2010.

[21] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Multiple cameras fall dataset," DIRO - Université de Montréal, Tech. Rep. 1350, Jul. 2010.

[22] Multi camera fall dataset. (2010). [Online]. Available: http://vision3d.iro.umontreal.ca/fall-dataset/

Authors' photographs and biographies not available at the time of publication.