International Journal of Soft Computing 12 (4): 253-262, 2017
ISSN: 1816-9503

Improved Elderly Fall Detection by Surveillance Video using Real-Time Human Motion Analysis

O. Dorgham, Sanad Abu Rass and Habes Alkhraisat
Prince Abdullah Bin Ghazi Faculty of Information Technology,
Al-Balqa Applied University, 19117 Al Salt, Jordan

Abstract: Statistical studies show that around 28-35% of older people aged 65 and over fall each year. This percentage increases to 32-42% among those over 70 years of age. These figures explain the dramatic increase in the number of systems that have been developed in recent years with aim of detecting falls. In this study, we propose, implement and evaluate a multiphase system framework to analyze human motion in real time to detect falls among the elderly. The system phases consist of background subtraction to extract the foreground of the frame for further analysis object classification which performs some morphological operations and draws the contours of the detected objects to identify human bodies object tracking to reduce false alarms and fall detection to detect the occurrence of falls based on the bounding rectangle and surrounding points contour drawing methods and by utilizing dual-camera verification as well as a method of leg detection using a camera situated above the subject. The system starts with a background subtraction phase to detect moving objects. After that the moving object is classified as a human body or not. Objects classified as human bodies are then tracked to detect falls. The experimental results showed that the system has high performance and accuracy and that it can implement and process live videos and report falls instantly. Our system can process videos at an approximate frame rate of 20 FPS (using an Intel 2.8 GH quad core processor with 4 GB RAM) and with an accuracy of 88.1%.

Key words: Fall detection, real-time analysis, human activity recognition, surveillance video, object classification

INTRODUCTION

The population of elderly people has rapidly increased over the last two decades. This is partly due to the good healthcare available in many societies. Nevertheless we still need to take into consideration some particular issues when taking care of elderly people, one of which is elderly falls and their detection.

Studies show that falls are a major public health problem and concern among older people (Igual *et al*., 2013). Monitoring an unattended older person at home is a challenging problem for both clinicians and biomedical engineers. There are a number of reasons why older people fall, many of which are beyond our capacity to control. What is important in this context is to detect the fall early enough. Early detection of falls may help reduce the time that the elderly remain lying on the floor after falling (referred to as a "long lie"). The length of time spent lying on the ground is a key factor that determines the severity of a fall.

One observation to take into consideration is that older fallers are usually unable to get up again by themselves without direct assistance. Clinical studies show that a long lie can lead to hypothermia, dehydration, bronchopneumonia and pressure sores (Igual *et al*., 2013). Thus, early detection of falls is particularly critical if the older person lives alone or loses consciousness after falling.

In this study, we propose a computer-based solution to this problem. Specifically, we propose a video-processing method based on a Human Activity Recognition (HAR) System to detect human falls. Such systems help to recognize some the daily life activities of people and they depend on the information collected by cameras and/or other sensors to recognize a targeted activity, falls in our context.

There are two main techniques that can be applied to detect falls: a wearable fall detection system by Chen *et al*. (2006) which is an assistive wearable device whose main objective is to send alerts when a fall event

Corresponding Author: O. Dorgham, Prince Abdullah Bin Ghazi Faculty of Information Technology, Al-Balqa Applied University, 19117 Al Salt, Jordan

has occurred and been detected and template matching Wilson and Farnandez (2006) or analyzing a video sequence to detect a fall. The current work adopts the second method.

The motivation for this research is the fact that a relatively large number of people who fall, die because they do not receive healthcare after falling (Igual *et al*., 2013). Therefore, we need to develop an efficient HAR system to solve this problem. In general, HAR can be used in many applications to detect any human motion activity. In our context, we use it for elderly fall detection.

The primary objective of this work is to propose and implement a framework for real-time system recognition of physical activity and fall detection. The proposed framework is designed to process surveillance videos to detect when elderly people fall. It does this by processing the sequence of images in the video in real time, extracting human bodies from those images and then recognizing the activity of these human bodies and determining whether those activities are falls or not. Through experimental evaluation, we demonstrate that the proposed system is able to achieve high performance and high accuracy.

Our proposed system consists of the following phases: background removal, object classification, object tracking and finally, fall detection. The last phase applies two algorithms, one which is newly proposed in this study and termed the "Points Alarm algorithm". The other is the rectangle alarm algorithm which is adopted from the literature.

Foroughi *et al*. (2008) developed a method to detect falls by analyzing the posture of the human body while doing activities. First, background subtraction is used in order to focus only on moving human bodies. After that, an approximated ellipse is drawn around the detected human body to give more precise details about its shape and orientation. To discriminate abnormal behaviors from normal activities, two values are used: the orientation standard deviation and the ratio standard deviation of the ellipse. After that, projection histograms of the silhouette of the frame are produced to recognize the shape deformation. Next, a normalization step is performed on the histograms using a discrete Fourier transform to give better results. In addition the research came up with the idea of tracking the human head in order to determine whether the recognized detected activity is a fall or not. This idea is based on the fact that the human head moves more than the rest of the body during a fall. Finally, they used neural networks to train the system to predict falls (Foroughi *et al*., 2008).

Chen *et al*. (2010) introduced a fall detection system that is based on an ingenious combination of skeleton feature and human shape variation. This system is able to successfully and efficiently distinguish "fall-down" activities from "fall-like" ones. The system starts with background subtraction so it can then use object tracking. Next the human skeleton is identified and extracted from images in hand. The distance between the two sampling skeletons determines what the authors refer to as "the posture change". In this work, also the researchers use an ellipse to approximate the detected human body's shape and orientation and to compute the ratio of its major and minor semi-axes. This helps in accurately detecting the human shape change rate. To analyze the posture of the human skeleton the authors use the Douglas-Peucker method to approximate the contour of the foreground object with fewer vertices. After that they apply the constrained Delaunay triangulation technique to divide the foreground object into triangular meshes and then extract the human skeleton by using the depth first search algorithm on the centers of the triangular meshes. After that they calculated a distance map for the human skeleton and then detected the posture change by calculating the distance map of two human skeletons every 0.4 sec. Finally, to confirm that a fall has indeed occurred they determine whether or not the object is still on the ground based on the rate of change in the orientation of the ellipse, denoted $R\theta$ and the ratio of major and minor of the ellipse semi-axes, $R\gamma$ being smaller than a certain threshold level for a predefined period of time (Chen *et al*., 2010).

Stone and Skubic (2015) proposed a two-stage system to detect human falls. During the first stage the person's vertical state is characterized to produce a vertical state time series which is obtained by tracking the person over time. The second stage uses an ensemble of decision trees to confirm the fall. In more detail, their solution works as follows: background subtraction is used for characterizing the vertical state of the object (a human body). After that, so-called "on-ground events" are identified through temporal segmentation of the vertical state time series of tracked 3D objects. During the second stage, a decision tree is used to confirm the occurrence of a fall and this is done by analyzing the set of features extracted from the on-ground events. A vertical state has three features: the maximum height of the object, the height of the object's centroid and the number of elements of the discredited projection-ground features, however are: minimum vertical velocity, maximum vertical acceleration mean velocity, occlusion-adjusted change and minimum frame-to-frame vertical velocity (Stone and Skubic, 2015).

Rougier *et al*. (2011) proposed a method to detect falls which works by analyzing human shape deformation during a video sequence. A shape matching technique is used to track the person's silhouette throughout the video sequence. The shape deformation is then quantified from these silhouettes based on shape analysis methods. Finally, falls are distinguished from normal activities by using a Gaussian Mixture Model (GMM). The researcher also use a head tracking model to reduce false alarms (Rougier *et al*, 2011).

To conclude from our review of the available literature we noticed that many studies have been conducted in the field of fall detection but found that there was still room for improvement. Therefore, this research focuses on exploring ways to enhance the accuracy of fall detection such as by using the bounding rectangle the surrounding ellipse and posture recognition (Diraco *et al*., 2010) in addition to other techniques. Some research studies have made contributions to this field with respect to the performance of fall detection in real time. Our work presented here makes four contributions to the field of fall detection:

- We achieve high real-time performance. This achievement is partially due to the use of the GMM technique for background subtraction (Bouwmans *et al*., 2008)
- We use the surrounding points of the detected object (the human body in our context) in addition to the bounding rectangle. This helps in improving the accuracy of the system without adding much to the complexity of the proposed solution
- We use two cameras simultaneously in order to further increase the accuracy of the system. It should be noted that the two cameras are not used to build 3D shapes of the objects we are still working in the 2D space. However, the usage of two cameras helps to obtain more precise decisions regarding the occurrence of falls
- We propose a method of human-leg detection (in the images taken by the top camera) to accurately detect falls

MATERIALS AND METHODS

In this research, the framework utilized to detect falls consists of four main phases (Paul *et al*., 2013) which are described in the following sub-sections.

Detection of moving objects: Object detection (the human body in this context) is achieved through background removal. Objects are detected in a sequence of frames to make it easier to focus on human bodies instead of wasting time in analyzing the shapes of other unrelated things which are not important (e.g., a sofa, a table, a door or other object in the room). In the literature, the most heavily utilized approaches in this field are optical flow and background subtraction. We use the latter in this work.

Object classification: Object classification is defined as the ability to identify an object in a scene, so that the system can focus only on the target object. Object classification is a critical phase in this framework because the level of success of the following phases of the framework depends on the quality of the output of this phase. In fact, the quality and accuracy of the entire framework depends on this phase because this phase is responsible for giving the right information about the detected objects. If the classification is wrong the later phases of the framework will be erroneous and produce wrong results. Consequently, this leads to many false positives. The classification methods available in the literature can be divided into three main categories: shape-based methods, motion-based methods and texture-based method. We use a shape-based method to classify the objects because it gives good accuracy and has low time complexity and, thus, is suitable for a real time system by Paul *et al*. (2013).

Object tracking: Object tracking can be described as keeping an eye on a specific object in a scene and in the video sequence in order to extract some key pieces of information about that object over time (Yilmaz *et al*., 2006). The main purpose of tracking an object is to reduce false alarms. In other words, if a system detects an object and it is classified as a fallen human the system should not give the alarm unless the object has been tracked for some period of time. To elaborate by way of an example assume that there is a sudden change in the lightness of a room. This will produce a sudden change in the scene leading to a false positive case. In our system, such a case is not classified as a fall because it is not preceded by a proper object tracking state (Ji and Liu, 2010). The method used here is that of silhouette and contour tracking frames (Yilmaz *et al*., 2006) because the objects that are tracked are predefined to be humans only, so there is no need for a complex tracking algorithm to undertake the tracking. This also helps in creating a real time system for fall detection.

Fall detection decision: This is the last phase in our fall detection framework. In this phase, the decision is taken as to whether a certain detected human activity is classified as a fall or not. A general view of the fall detection system is shown in Fig. 1.

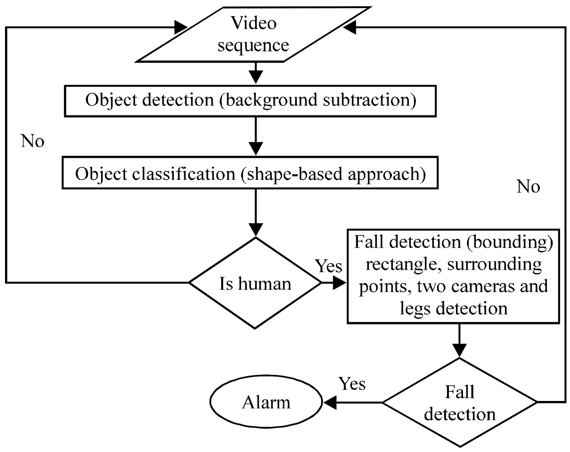

Fig. 1: General view of the fall detection system

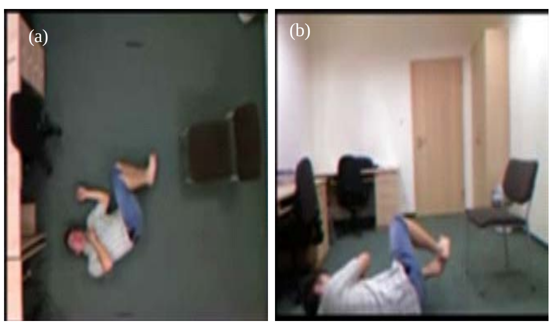

Fig. 2: a) View from the top camera and b) View from the bottom camera

The last phase of the fall detection framework involves making a decision regarding whether a certain detected human activity should be classified as a fall or not. However, we start by describing the system setup used in our experiments. In order to increase the accuracy of our system, we developed a system that uses two cameras. The first camera is placed on the ground and the other is hung from the ceiling of the room. The goal here is to achieve a double-view of the room. In our system, a fall-alarm is generated if and only if the two cameras both detect a fall instance. If only one camera detects a fall instance a fall-alarm is not generated. This helps reduce the number of false alarms. Figure 2 shows an example of the same scene detected by the two cameras.

Next, we give an example of how we perform the object detection background removal step. Background removal is done through background subtraction. Background subtraction is an elementary phase in most computer vision applications because it allows the extraction and identification of only the moving objects from a scene and eliminates the non-moving ones (e.g., furniture). This helps save time by removing the processing of static objects that appear over and over again in a video sequence by Pham *et al*. (2010). A number of methods have been proposed in the literature for this step (Comer and Delp, 1999; Diraco *et al*., 2010; Foroughi *et al*., 2008; Han *et al*., 2011). In this study, we implement the extended GMM Pham and Hurng (2010):

$$P(I_t)\sum_{i=1}^{M} w_i N\left(I_t; \mu_i; \sigma^2 I\right) \quad (1)$$

where, GMM stores M separated normal distributions for each pixel (parameterized by mean μ_i, variance σ^2_i and mixing weight w_i where i = 1, 2,…, M) with M typically between 3 and 5 (depending on the complexity of the scene). This leads to a probability distribution for pixel value I_t. Figure 3a and b show the results of background subtraction using the extended GMM.

Then, morphological operations (i.e., erosion, dilation, opening, closing) are used to clarify the features of the detected moving objects Comer and Delp (1999) as follows:

Erosion:

$$A \ominus B = \{z \in E \backslash B_z \subseteq A\} \quad (2)$$

Where:

E = An integer grid
A = A binary image in E
B = A structuring element
B_z = The translation of B by vector z

Dilation:

$$A \oplus B = \left\{z \in E \middle| \left(B^s\right)_z \cap A \neq \varnothing\right\} \quad (3)$$

Where:

E = An integer grid
A = A binary image in E
B = A structuring element
B_z = The translation of B vector z
B^s = The symmetric of B

Opening:

$$A \circ B = (A \ominus B) \oplus B \quad (4)$$

where, the opening of A by B is obtained by the erosion of A by B followed by dilation of the resulting image by B.

Closing:

$$A \cdot B = (A \oplus B) \ominus B \quad (5)$$

where, the closing of A by B is obtained by the dilation of A by B, followed by erosion of the resulting structure by B.

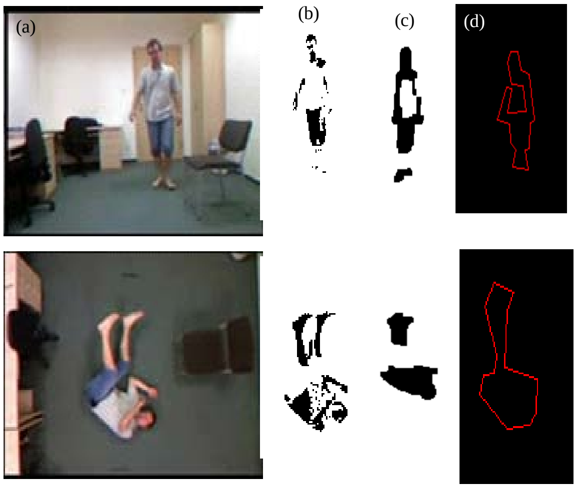

Fig. 3: a) Current frame; b) Current frame after background subtraction; c) Result of morphological operations and d) Drawing contours for the object

Morphological operations are used to clarify the object after it has been extracted from the background using background subtraction by using GMM as described above. Figure 3c shows results of this phase.

The next step is to draw contours around the detected moving objects (Yilmaz *et al*., 2006). A contour is defined as the outside shape of the detected moving object. It is easier to process the outside shape of an object because the inside of a detected object may contain noise. To draw contours around an object, we must highlight some surrounding points on the object and after that connect these points together to form the contour. This contour is used in the subsequent steps to classify the object as a human body or not. If the object is classified as human, it is tracked to detect any fall instances in the near future. Figure 3d shows the results of this step.

Object classification is an essential phase in a HAR application and all the phases after that depend on it. It helps the system to process and focus only on a targeted object or set of objects and, thus, reduces false alarms and wastes no time processing unrelated objects from the scene.

Many approaches have been used in the literature to classify objects in video sequences. Therefore, we chose a classification technique that was best suited to the environment in which the system will be used. The classification technique, we selected for our context is the ratio of the width and the height of the identified contour. This technique takes advantage of the height and width property of the detected objects. To apply this method, first a bounding box is drawn around the contour of the object. After that, threshold values for the maximum and

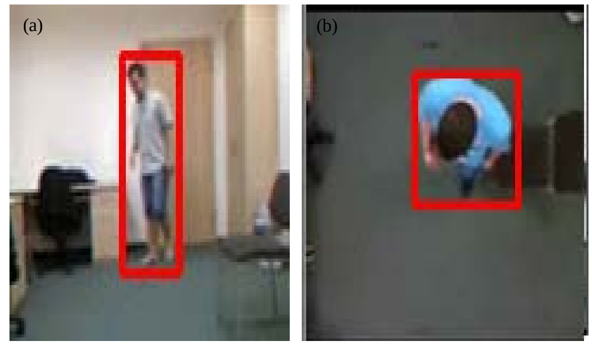

Fig. 4: Results for detecting standing: a) Sitting and b) Humans

minimum height and for the maximum and minimum width are selected for the vertical state of the object, depending on the location of each camera; if the camera is close to the object it will appear bigger and if it is far away the object will appear smaller. The decision in this step is based on the moving object's vertical state bounding box width and height. If the ratio of some given object is within the range of threshold values predefined for the system, it is considered a human body object as show in Fig. 4.

The next step is the object tracking step which aims at finding the same object detected in the previous step in a sequence of later frames. The literature contains a number of techniques to achieve this aim by Yilmaz *et al*. (2006) and Li *et al*. (2013) such as: point tracking where objects detected in sequential frames are represented by points and point matching is done. This approach requires an external mechanism to detect the objects in every frame (Yilmaz *et al*., 2006; Li *et al*., 2013). Kernel tracking where a kernel is defined as the object shape and appearance. For example, a kernel may be a rectangular template or an elliptical shape with an associated histogram. Objects are tracked by computing the motion of the kernel in consecutive frames (Yilmaz *et al*., 2006; Li *et al*., 2013).

Silhouette tracking where the information is encoded inside the object region (i.e., by appearance density and shape models). Given the object models, silhouettes are tracked by either shape matching or contour evolution. The latter can be considered as object segmentation applied in the temporal domain using the priors generated from the previous frames (Yilmaz *et al*., 2006; Li *et al*., 2013).

We use silhouette tracking in our implementation to track the humans in the surveillance videos because the environment of this system is simple, so it is better to use a simple tracking method to achieve real-time performance for fall detection.

The fall detection decision is the last and the most important phase in the system. After getting the frame from the video sequence and applying background subtraction the system performs morphological operations on the frame and then the contours of the image are drawn, so that they are ready to be classified. The classification phase uses the bounding box approach to determine whether the object is human or not. After that the silhouette image is tracked to find the same object throughout the video sequence and if the object is found and tracked the fall detection phase starts.

As mentioned above there are many approaches to detect a fall, one of which is shape deformation and this is the approach used in this work. Shape deformation is defined as massive changes in the shape of the object. For example, if a human is standing or walking and then suddenly falls down the outer shape of his/her body changes in an abnormal way (Han *et al.*, 2011). So, if the system is tracking the human and that human is in his/her vertical state the shape of their body will significantly change if they fall.

One of the main contributions of this work is presented next. After tracking the object and making sure that the object is human the system starts processing each frame to detect a fall as described in the following studies.

Determining the ground: Most of the time, falls result in a person landing on the ground. Thus, to achieve higher accuracy in detecting falls the ground in the scene needs to be identified. Note that a human might decide go to sleep on a couch or on his/her bed. The sleeping posture is almost the same as that of a person who has fallen on the ground. So, we believe it is better to determine the ground manually in the scene to reduce false alarms arising from such scenarios.

Bounding rectangle: In our system, each tracked object is bounded with a rectangle as mentioned earlier. The width and height of this rectangle are crucial for detecting whether a person has fallen or not. As mentioned above the system uses two cameras. The bottom camera (placed at ground level) represent the human's vertical state as a rectangle with a height that is bigger than the width (by at least two times). The top camera (placed at the ceiling) represents the human's vertical state as a square, most of the time as shown in Fig. 5a and b.

It is obvious that there is difference between the vertical states shown by the top and bottom cameras. Also, after a fall two different rectangles for the human body are detected by the cameras. By using proper threshold values for the two states one can accurately detect fall occurrences. These threshold values are: minimum and maximum height and minimum and maximum width, depending on the location of the camera and how far the camera is from the object in the normal case. Our system draws a bounding rectangle around the detected fallen object as shown in Fig. 5c and d.

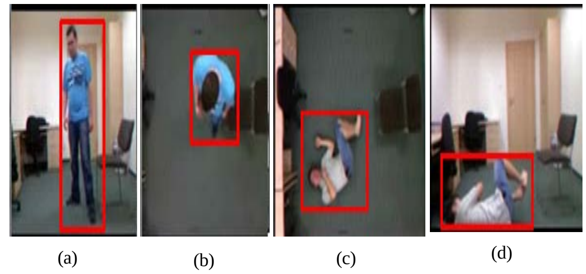

Fig. 5: a) The rectangle for the vertical state in the bottom camera; b) The rectangle for the vertical state in the top camera; c) The rectangle for the fall in the bottom camera and d) The rectangle for the fall in the bottom camera

Surrounding points: Now we present the first main contribution of this work. Above we mentioned a number of techniques that can be used to draw contours around detected moving objects. Basically, these contours are connected points surrounding the objects. We experimentally observed that it is efficient to depend on these points to detect falls.

Note that each object has a number of surrounding points depending on the height and width. Thus the number of these points differs for different objects depending on their dimensions. One possible indicator of falls can be based on the number of these points. That is, if the number of points is within a certain range of threshold values it can be determined whether a human has fallen or not. To elaborate by way of an example; when a person is standing in the bottom camera's view there are more surrounding points than the number of points when he/she is in the fallen position. The opposite is the case for the top camera view. Consequently, if the bounding rectangle method detects a fall and the surrounding points' method also detects a fall the system will consider this to be a real fall.

Dual cameras: The use of dual cameras is the second major contribution of this work. Although, this idea has been used in previous research by Rougier *et al.*, 2011), Auvinet *et al.* (2011) the difference in our case lies in the mechanism that we utilize to combine the decisions of the multiple cameras and the extracted information collected from each one of them to indicate whether a fall has occurred or not.

Further, we use dual cameras to observe the scene from different views to detect falls, not to draw a 3D view of the scene. Some prior research studies have used multiple cameras to generate a 3D posture of the body and after that to do the processing (Thome *et al.*, 2008). We believe that that type of approach, however will hardly achieve real-time performance. So in our method, the dual cameras observe the scene and if the dual cameras make a decision that there has been a fall the system will generate an alarm. Note that in our system the views detected by both cameras remain two-dimensional.

Leg detection: As mentioned above, some research studies have used head tracking to detect falls (Rougier *et al*., 2006). However, in our work to increase the accuracy of the top camera we assume that it is better to detect the legs of the detected human body. Note that when a person is standing, his/her legs are invisible to the top camera. However, when that person falls their legs become visible to the camera. One indicator that a true fall has occurred is the fact that the top camera has detected legs as illustrated in Fig. 6. This idea represents another contribution of this work. We used the surrounding point's method to detect the legs. So, depending on the surrounding points of the legs we determine whether they are legs or not. Also, Fig. 7 shows detailed description for the fall detection mechanism.

Fig. 6: Sample results for leg detection

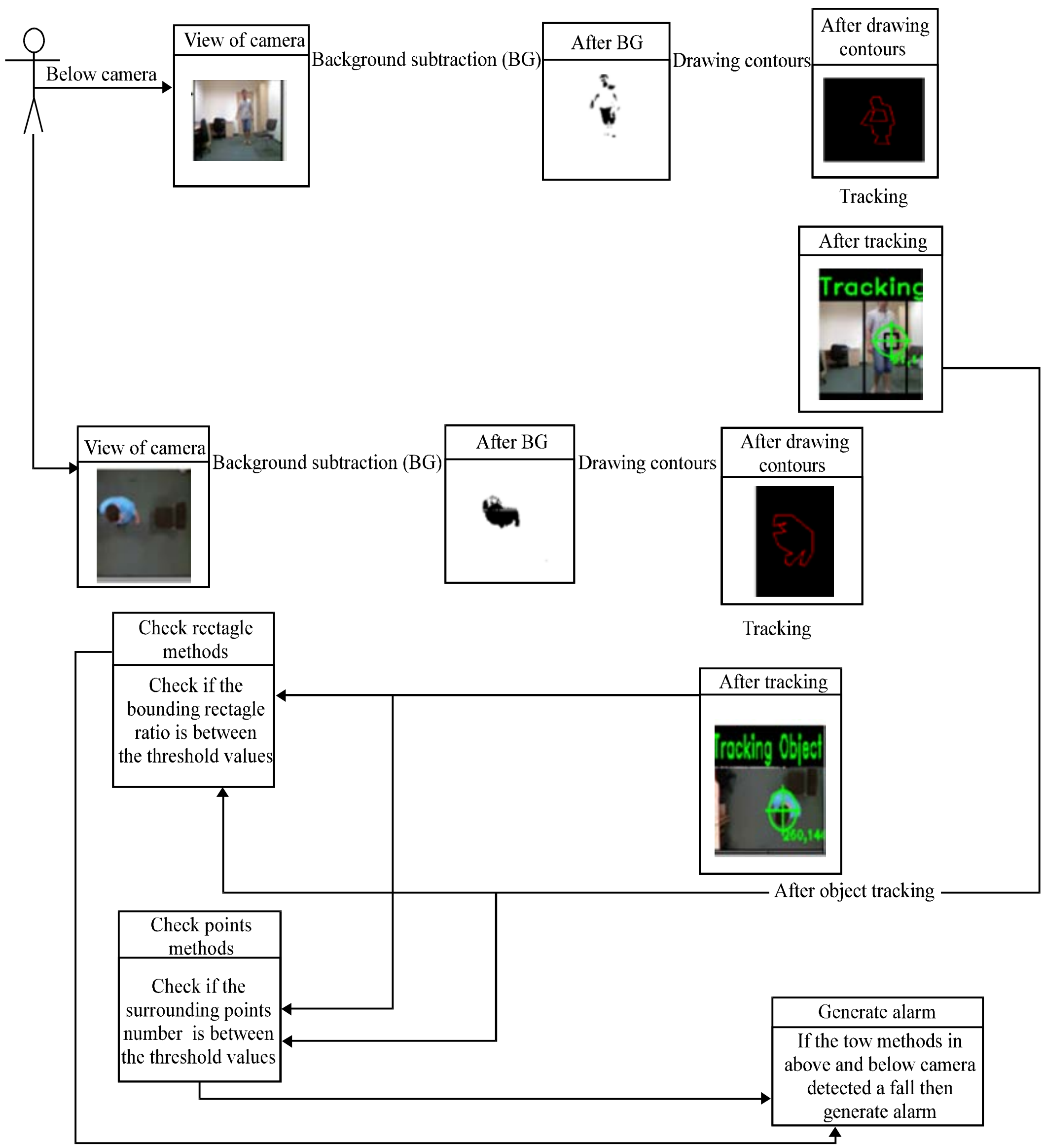

Fig. 7: Detailed description for the fall detection mechanism

RESULTS AND DISCUSSION

The proposed framework involving the proposed contributions was tested on two video files. The first one was collected from the bottom camera and the second one was from the top camera. The system concurrently performed the processing steps described above on both videos. If one camera detected a fall instance that particular camera changed its status to declare that it had detected a potential fall instance. If both cameras detected a fall the system produced an alarm for fall detection. However, if only one camera detected a fall the system did not generate an alarm. A sample of the results is presented in Fig. 8.

System performance: Based on our analysis of the experimental results the proposed system was able to achieve real-time performance. Our system could process up to 20 frames per second. The system was tested on two videos, one filmed using the top camera and the other by the bottom camera. The resolution of the videos was 340×260 pixels with an overall total frame number equal to 2506 per video. The test results for our system showed that the performance of the top camera and the bottom camera was almost the same, at approximately 20 FPS, using an Intel 2.8 GH quad core processor with 4 GB RAM.

System accuracy: A dataset from (Kwolek and Kepski, 2014) which contains 70 activities (30 falls+40 daily living) within 2506 frames has been tested using our system. From the bottom camera, our system successfully detected 26 real falls. Further, two real falls were detected by the bounding rectangle contouring technique only, one was detected by the surrounding points' method only and the last one was not detected at all.

From the top camera, our system successfully detected 23 out of the 30 falls. Further, 2 falls were detected by the bounding rectangle contouring technique only, 2 falls were detected by the surrounding points' method only and the last 3 were not detected at all. We estimated the accuracy of the system as 88.1% by using the following equation (Han *et al*., 2011):

$$\text{Accuracy} = \frac{(\text{TP}+\text{TN})}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \quad (6)$$

Where:

TP = True positive (real falls detected and reported by the system as true falls)
TN = True negative (false falls missed by the system as false falls)
FP = false positive (false falls detected and reported by the system as true falls)
FN = false negative (real falls missed by the system as falls)

Fig. 8: Sample results for the system alarm

Table 1: Values of TP, TN, FP and FN for both cameras

Camera	TP	TN	FP	FN	Accuracy (%)
Top camera	1632	580	114	180	88.2
Bottom camera	1327	974	94	111	91.8

Table 2: The accuracy of the two methods used to draw contours around moving objects (and the combination of the two)

Method	Bottom camera accuracy (%)	Top camera accuracy (%)	Overall system (%)
Accuracy			
Points method	91.6	94.5	91.6
Rectangle method	93.2	88.1	88.1
Combined methods	91.6	88.1	88.1

Based on our experiments, we found that: accuracy (top camera) = (2212/2506)×100% = 88.2%. Accuracy (bottom camera) = (2301/2506)×100% = 91.8%. Where values of TP, TN, FP and FN for both cameras are given in Table 1.

Note that our system will generate an alarm if and only if the two cameras (bottom camera and top camera) detect a fall. This approach improved the accuracy of our system. Table 2 shows the accuracy of the two methods used to draw contours around moving objects (and the combination of the two). Also, Fig. 9 shows a graphical representation of the system accuracy using the different methods.

The reasons for using two cameras in our approach, even though the accuracy of the bottom camera was found to be higher than that of the top camera are as follows.

Using two cameras gives a full view of the room, so the object will always be seen. Using two cameras reduces the number of false alarms because some objects might

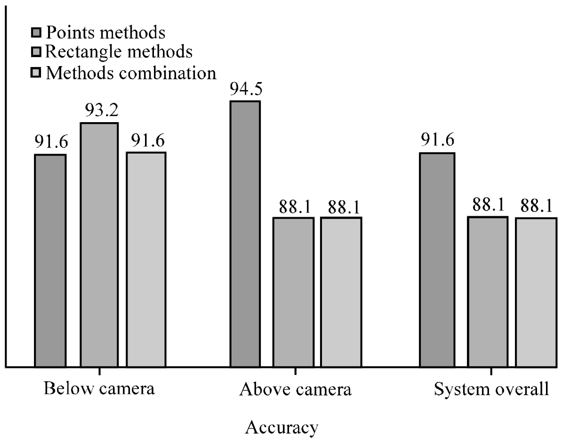

Fig. 9: System accuracy

have the same bounding rectangle or the same surrounding points as the fallen person, so it is better to use two cameras to make sure that the object identified is a person who has fallen to the ground, not anything else.

CONCLUSION

In this study, we presented a proposed system framework for analyzing human motion in real time to detect falls among elderly people. The proposed system consisted of the following phases:

Background subtraction in this phase the main goal is to extract the foreground of the frame to analyze it. Object classification after some morphological operations and drawing the contours of the object the system tries to classify the object to detect human bodies in the frame and to reduce unneeded object analysis. Object tracking: to make sure that the detected fall is actually a human fall the system tracks the object movement for some time before it decides it has detected a fall this is done to reduce false alarms.

Fall detection finally this phase detects a fall by using the bounding rectangle and surrounding points contour drawing methods as well as by utilizing dual-camera verification and leg detection by the top camera. The system starts with background subtraction to detect moving objects. After that the moving object is classified as human body or not. Then, the object classified as a human body is tracked to detect falls. The experimental results showed that the system has high performance and accuracy and can process live videos and report falls instantly. Our system can process videos at an approximate frame rate of 20 FPS (using an Intel 2.8 GH quad core processor with 4 GB RAM) and with an accuracy of 88.1%.

REFERENCES

Auvinet, E., F. Multon, A.A. Saint, J. Rousseau and J. Meunier, 2011. Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution. IEEE. Trans. Inf. Technol. Biomed., 15: 290-300.

Bouwmans, T., E.F. Baf and B. Vachon, 2008. Background modeling using mixture of gaussians for foreground detection-a survey. Recent Pat. Comput. Sci., 1: 219-237.

Chen, J., K. Kwong, D. Chang, J. Luk and R. Bajcsy, 2006. Wearable sensors for reliable fall detection. Proceedings of the IEEE 27th Annual International Conference on Engineering in Medicine and Biology Society, January 17-18, 2006, IEEE, Shanghai, China, ISBN:0-7803-8741-4, pp: 3551-3554.

Chen, Y.T., Y.C. Lin and W.H. Fang, 2010. A hybrid human fall detection scheme. Proceedings of the 2010 17th IEEE International Conference on Image Processing, September 26-29, 2010, IEEE, Hong Kong, China, ISBN:978-1-4244-7994-8, pp: 3485-3488.

Comer, M.L. and E.J. Delp, 1999. Morphological operations for color image processing. J. Electron. Imaging, 8: 279-289.

Diraco, G., A. Leone and P. Siciliano, 2010. An active vision system for fall detection and posture recognition in elderly healthcare. Proceedings of the International Conference on Design, Automation and Test in Europe, March 08-12, 2010, European Design and Automation Association, Leuven, Belgium, ISBN:978-3-9810801-6-2, pp: 1536-1541.

Foroughi, H., B.S. Aski and H. Pourreza, 2008. Intelligent video surveillance for monitoring fall detection of elderly in home environments. Proceedings of the 11th International Conference on Computer and Information Technology, December 24-27, 2008, IEEE, Khulna, Bangladesh, ISBN:978-1-4244-2135-0, pp: 219-224.

Han, J., M. Kamber and J. Pei, 2011. Data Mining: Concepts and Techniques. 3rd Edn., Morgan Kaufmann Publishers, USA., ISBN-13: 9780123814791, Pages: 744.

Igual, R., C. Medrano and I. Plaza, 2013. Challenges, issues and trends in fall detection systems. Biomed. Eng. Online, 12: 2-24.

Ji, X. and H. Liu, 2010. Advances in view-invariant human motion analysis: A review. IEEE. Trans. Syst. Man Cybern. Appl. Rev., 40: 13-24.

Kwolek, B. and M. Kepski, 2014. Human fall detection on embedded platform using depth maps and wireless accelerometer. Comput. Methods Programs Biomed., 117: 489-501.

Li, X., W. Hu, C. Shen, Z. Zhang and A. Dick *et al*., 2013. A survey of appearance models in visual object tracking. ACM. Trans. Intell. Syst. Technol., 4: 58-58.

Paul, M., S.M Haque and S. Chakraborty, 2013. Human detection in surveillance videos and its applications-a review. EURASIP. J. Adv. Signal Process., 2013: 1-16.

Pham, V., P. Vo and V.T. Hung, 2010. GPU implementation of extended gaussian mixture model for background subtraction. Proceedings of the 2010 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation and Vision for the Future, November 1-4, 2010, IEEE, Hanoi, Vietnam, ISBN:978-1-4244-8075-3, pp: 1-4.

Rougier, C., J. Meunier, A. St-Arnaud and J. Rousseau, 2011. Robust video surveillance for fall detection based on human shape deformation. IEEE. Trans. Circuits Syst. Video Technol., 21: 611-622.

Rougier, C., J. Meunier, S.A. Arnaud and J. Rousseau, 2006. Monocular 3D head tracking to detect falls of elderly people. Proceedings of the IEEE 28th Annual International Conference on Engineering in Medicine and Biology Society, August 30- September 3, 2006, IEEE, New York, USA., ISBN:1-4244-0032-5, pp: 6384-6387.

Stone, E.E. and M. Skubic, 2015. Fall detection in homes of older adults using the Microsoft Kinect. IEEE. J. Biomed. Health Inf., 19: 290-301.

Thome, N., S. Miguet and S. Ambellouis, 2008. A real-time, multiview fall detection system: A LHMM-based approach. IEEE. Trans. Circuits Syst. Video Technol., 18: 1522-1532.

Wilson, P.I. and J. Fernandez, 2006. Facial feature detection using Haar classifiers. J. Comput. Sci. Colleges, 21: 127-133.

Yilmaz, A., O. Javed and M. Shah, 2006. Object tracking: A survey. ACM Comput. Surv., Vol. 38. 10.1145/ 1177352.1177355.