Intelligent Video Surveillance for Monitoring Elderly in Home Environments

Conference Paper · November 2007

DOI: 10.1109/MMSP.2007.4412853 · Source: IEEE Xplore

Sabu Emmanuel
Kuwait University

Intelligent Video Surveillance for Monitoring Elderly in Home Environments

Arie Hans Nasution and Sabu Emmanuel
School of Computer Engineering
Nanyang Technological University, Singapore
{arie0001, asemmanuel}@ntu.edu.sg

Abstract—**In this paper we propose a novel method to detect and record various posture-based events of interest in a typical elderly monitoring application in a home surveillance scenario. These events include standing, sitting, bending/squatting, side lying and lying toward the camera. The projection histograms of segmented human body silhouette are used as the main feature vector for posture classification. k-Nearest Neighbor (*k-NN*) algorithm and evidence accumulation technique is proposed to infer human postures. With this technique we have achieved a robust recognition rate of above 90% and a stable classifier's output. The modified classifier structure also improves greatly the recognition rate of lying toward the camera events as compared to the result of classifier's structure in *GHOST* [1]. Furthermore, we use the speed of fall to differentiate real fall incident and an event where the person is simply lying without falling.**

Keywords—**home surveillance, fall detection, posture recognition, elder monitoring.**

I. Introduction

One of the application areas of video surveillance is monitoring the safety of elderly in home environments. Often elderly are left alone in the house during day time while other members of the family are at work or some elderly prefer to reside alone in their house. The elderly are prone to accidental falls causing serious/fatal injuries. It is known that falls are the leading cause of injury deaths among individuals who are over 65 years of age. In most of those fall cases, the injured person has difficulties for asking assistance from third parties. Thus, a system which provides automatic fall detection of elderly people living in home would be of great help in alerting others. In addition it would be of great help to doctors if they had access to the chronology of postures that the elderly had gone through before, during and after the fall event. Thus not only detecting and recording fall events, but also other posture-based events in a home environment are of utmost importance. The recorded events will provide doctors or geriatrics with greater insights into the condition of elder and will assist them in formulating better diagnosis and treatment plan. We consider five posture-based events of interest which consist of standing, sitting, bending/squatting, side lying and lying toward the camera as part of this research.

Many researches have been carried out on posture estimation and fall accident detection. The works in [1, 2] used the normalized vertical and horizontal projection of segmented object as feature vectors in frame-wise posture classification. However, the output of frame-wise classification is considered unstable [3]. We propose evidence accumulation technique to address this shortcoming. Ji tao et al. [4] proposed the use of CUSUM algorithm to achieve fall detection. The feature they use is the aspect ratio of the moving object's bounding box. Nait-Charif and McKenna [5] infers falling incident when target person is detected as inactive outside normal zones of inactivity. In [6], the 3D trajectory and velocity of head is utilized to detect fall incident. Miaou et al. [7] detects fall using aspect ratio of detected object with video captured using omni-directional camera. These existing fall detection schemes [2, 4, 5, 7] are unable to differentiate between real fall incident and an event where the person is simply lying without really falling. For example, in [2], falling event is inferred when the system detects a long permanence in the laying down static state. This might not be a valid falling situation as the person can simply be sleeping. Other systems use the audio information or using 3D trajectory and speed of head to infer real fall events [6]. These mechanisms tend to be more complex and need additional cost of sensor. We propose the use of falling speed to differentiate real fall incident and an event where the person is simply lying without falling.

In this paper, we present a novel method, which aims not only to detect and record fall events, but also other posture-based events in a home environment. In section II, we will explain our proposed method in detail. While in section III, we will discuss the experimental results, and in section IV conclusion and future work is presented

II. Proposed Method

In this paper, we consider indoor environment setting with single fixed camera monitoring static scene. We assume a point of view where human posture is easily recognizable without ambiguities.

The first step in our method is to obtain the segmentation of moving objects. We achieve this by adapting background subtraction approach developed by Stauffer and Grimson [8]. We remove the adaptive characteristic to prevent the eventual inclusion of static person as the background. This is motivated by the fact that elderly may spend long period of their time in a static position such as sitting or sleeping.

1-4244-1274-9/07/$25.00 ©2007 IEEE

The next step is feature extraction process for foreground object. We use horizontal and vertical projection histograms of segmented foreground and angle between last standing posture with current foreground bounding box as feature set for the task. The extracted projection histograms features will then be used as input for the classifier. The classifier is based on *k-NN* algorithm combined with evidence accumulation mechanism. When this *k-NN* based classifier detects lying toward the camera event, we use bounding box angle test for further asserting this event. This is because *k-NN* classifier would sometimes classify wrongly other postures as lying toward the camera. Finally, we use the falling speed to infer real falling events. These steps are summarized in Fig. 1.

We will now discuss how the projection histograms are computed, the posture classification using *k-NN* and evidence accumulation technique, bounding box angle test and real fall inference method in detail.

A. Projection Histograms

The horizontal and vertical projection histogram of foreground is obtained by calculating the number of foreground pixels row wise and column wise. Since projection histograms vary according to location of object in the scene, we need to perform a normalization step first. The normalization is done by rescaling the detected silhouette into a fixed vertical length of M pixels (in our method we fixed M = 128 pixels since it produces good results and runs fast) while keeping the same aspect ratio. This rescaling is done by bilinear interpolation method using openCV library. Then we perform foreground pixels calculation to obtain horizontal projections as follows.

Let us denote the foreground F as cloud of 2D points and (x_p,y_p) is pixel coordinate. The horizontal projection histogram $HZ(y)$ of foreground F can be defined as the cardinality of set of points as follows:

$$HZ(y) = |(x_p, y_p) \in F, y_p = y| \qquad (1)$$

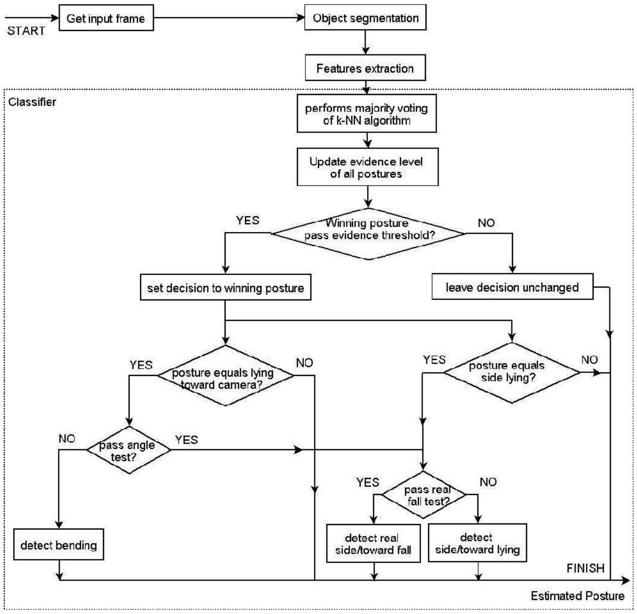
START
Get input frame
Object segmentation
Features extraction
Classifier
performs majority voting of k-NN algorithm
Update evidence level of all postures
YES
Winning posture pass evidence threshold?
NO
set decision to winning posture
leave decision unchanged
YES
posture equals lying toward camera?
NO
YES
posture equals side lying?
NO
NO
pass angle test?
YES
YES
pass real fall test?
NO
detect bending
detect real side/toward fall
detect side/toward lying
FINISH
Estimated Posture

Fig. 1. Overview of proposed method

To obtain vertical projection histogram, the normalization is again performed by rescaling the detected silhouette into a fixed horizontal length N = 128 pixels while keeping the same aspect ratio. Vertical projection histogram $VT(x)$ can then be computed as follows:

$$VT(x) = |(x_p, y_p) \in F, x_p = x| \qquad (2)$$

Since we fix 128 elements for each horizontal and vertical projection histogram, we will have in total a feature vector of length 256 elements as the input for the *k-NN* classifier. We use this feature vector for training and testing *k-NN* based classifier for postures classification.

B. Posture Classification using k-Nearest Neighbor (k-NN) Algorithm and Evidence Accumulation Technique

First, in the offline training phase, we compute m templates for each five postures of interest. Fig. 2 shows the five postures considered in the system. Let $HZ^c{}_{ij}$ and $VT^c{}_{ij}$ be the horizontal and vertical projection histogram for i-th template of j-th posture respectively, where i = *{0,1,...,m}* and j = {*standing, sitting, bending, side lying, lying toward*}.For each posture j, we have m training videos. And for each video, we construct horizontal and vertical projection histograms templates by averaging projection histograms obtained from frames in the video.

After the training phase, the computed projection histograms feature for current input frame is compared to stored posture templates. We use *k-NN* algorithm [9] to find the most similar posture. *k-NN* algorithm is chosen since it allows us to employ multiple templates for each posture which in turn can give better recognition results. In our method, we use three templates (m = 3) for each posture with k parameter of *k-NN* algorithm equals to five. These parameters are selected as they give good recognition results with fast classification speed.

In order to classify the posture of foreground object in current frame, we first compute the horizontal and vertical projection histograms from the current frame of video sequence denoted as HZ^v and VT^v respectively. Then we compute the distance of HZ^v and VT^v to stored template i of posture j as follows:

$$S_{ij} = \sqrt{\sum_{y=0}^{M} (HZ^V(y) - HZ_{ij}^C(y))^2 + \sum_{x=0}^{N} (VT^V(x) - VT_{ij}^C(x))^2} \qquad (3)$$

After computing S_{ij}, the *k-NN* algorithm is used to pick the winning posture w using majority voting. However, we will not directly choose posture w as the output of classifier. This is because this kind of frame-by-frame classification is considered not reliable as stated in [3].

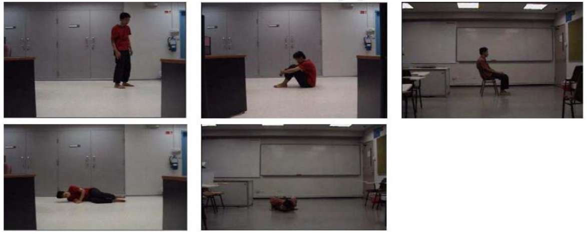

Fig. 2. Postures considered in the system. From the top left in clockwise direction: standing, bending, sitting, side lying and lying toward

The reason is due to the imperfect output of segmentation stage and also the presence of noise as well as shadows, which might lead the classifier to false classification. To minimize this instability at the output of classifier we propose the use of evidence accumulation technique. The classifier will not directly choose the voting winner of *k-NN* algorithm as the output, but instead, it will base the decision on the evidence level of each posture class.

Let us define minimum distance D_{min}, which is computed as:

$$D_{min} = \min (S_{iw}, \text{ where } i=\{0,1,...,m\} \text{ and } w \text{ is winning posture}) \quad (4)$$

During the frame-by-frame processing of video, we maintain evidence level E_j^t, where *j* refers *j*-th posture and *t* refers to *t*-th frame. Initially, these evidence levels are set to zero. We update evidence level of all postures using the following update rule:

$$E_j^t = \begin{cases} E_j^{t-1} + \dfrac{E_{const}}{D_{\min}}, & \text{for } j = w \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where E_{const} is a predefined constant. In the experiment we set the value of E_{const} to 10000.

The updated evidence levels E_j^t are then compared against a set of threshold values TE_j, which correspond to each posture. The moment E_j^t exceeds TE_j we decide posture *j* as the final decision output of classifier. If no E_j^t exceeds their corresponding TE_j, we keep choosing the most recent output of classifier as the current decision. Thus, transient change in the output of classifier will not be reflected in the final decision output as long as the evidence has not exceeded the threshold value

The evidence threshold values are chosen flexibly based on desired condition, if for example bending is not of interest to be detected, we can set its threshold to be high, while falling is considered as the most important event to detect, so naturally we will set the evidence threshold level to be low.

C. Bounding Box Angle Test

We classify the five postures including lying toward the camera in a single level manner. This single level classifier structure gives much better reliability of the system to detect the occurrence of falling toward the camera than a hierarchical level classifier structure as in [1]. However, there is some bending posture, which is sometimes falsely detected as lying toward camera. To filter out this invalid detection, we make use of bounding box angle test. This test is performed only when the output of classifier equals to lying toward the camera event. Bounding box angle θ can be defined as the angle formed by points *{RT1, Q, RT2}* as in Fig. 3, where *RT1* is the right top corner of current posture bounding box, *RT2* is the right top corner of bounding box in last detected standing position, and *Q* is a reference point taken as the right bottom point of standing box after alignment of both bounding boxes. Angle test is passed if θ is less than a threshold denoted as T_θ. If this test is passed then we say that the current posture is lying toward the camera, else the classifier decides as bending posture. This also means that we need to store and update the bounding box information of the last detected standing posture all the time. In our system, we set T_θ as 20^0.

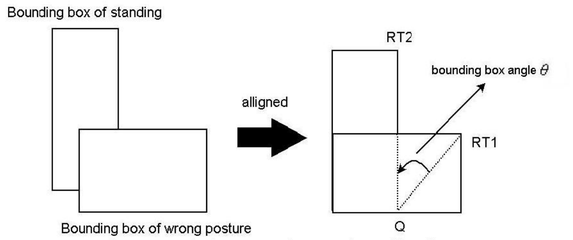

Fig. 3. Angle between the two bounding boxes

D. Real Fall Detection

A number of existing fall detection schemes [2, 4, 5, 7] are unable to differentiate between real fall incident and an event where the person is simply lying without really falling as mentioned in section I. We propose a simple and robust way to reduce the false classification of person lying on the floor as real falling. We make use of the temporal information about the last time standing posture was detected. We keep recording and updating the time when the last standing posture was recognized. When the posture estimator classifies current posture as side lying or lying posture, we check the time difference between time of lying t_{lying} and last standing $t_{standing}$. If t_{lying} - $t_{standing}$ is less than certain threshold T_{RF} we can assert that a real fall is taking place. In this way we can also somewhat sense the seriousness of a fall incident. The faster the fall is, the more likely that it is more serious. This sensing can be achieved by setting different time threshold values to judge the different seriousness level of fall incidents.

III. Experimental Results and Discussions

We have implemented the system using OpenCV library. It runs at 12-16 frames/second on an Intel Centrino Duo 1.83 GHz laptop PC with 1 GB of memory. Our dataset consists of 30 video clips with average duration of 2 minutes each. We assume a static background in an indoor environment with the distance of person to the camera is between 3-6 meters. In this experiment we want to evaluate the performance and accuracy of the system to recognize 5 posture-based events. The experiment is conducted by observing the video and noting the result of detection to see whether current event is classified correctly. Due to evidence accumulation technique, an event will not be instantaneously detected; however it takes on an average of 8 frames to accumulate enough evidence. In this experiment, we say that an event has been detected if before transition to another event, the system managed to output the correct event. Otherwise, we say that the event is undetected.

Table I presents the experimental results. N_a refers to number of actions/events, N_c is number of correctly detected events, N_f is number of falsely detected events and *R* is the percentage of correct detection (recognition rate). During this experiment, we use the following threshold values: $TE_{j=standing}$ = 30, $TE_{j=bending}$ = 240, $TE_{j=sitting}$ =450 , $TE_{j=lying}$ = 15, $TE_{j=lying\ toward}$= 15, E_{const} equals to 10000, and T_{RF} is set to 2 seconds.

The experimental result shows that the system has a robust recognition rate in detecting occurrence of considered events. To better understand the wrong classification results, we present the confusion matrix of the classifier output in Table II.

TABLE I. RECOGNITION RATE FOR VARIOUS EVENTS

Events	N_a	N_c	N_f	*R (%)*
Standing	365	355	10	**97.26**
Sitting	68	65	3	**95.59**
Bending	73	67	6	**91.78**
Lying	75	75	0	**100.00**
Lying toward	162	151	11	**93.21**

Bending and lying toward are mistaken because they sometimes produce pretty similar segmentation results. Standing event is wrongly detected as sitting when the segmented silhouette is disturbed by shadows.

The effect of evidence accumulation is also verified by comparing the output of classifier with and without the evidence accumulation technique. Fig. 4 shows a sample of this comparison. It can be seen that the output is less fluctuating with evidence accumulation in place. The trade-off is that the correct output response will be delayed for an average of 8 frames.

In section II.C, we mentioned that the modified classifier structure is performing better in recognizing events of lying toward the camera. To measure the increase in performance, we have compared the recognition rate of the proposed classifier structure with the hierarchical structure proposed in *GHOST* [1], which placed the detection of lying toward the camera posture in the second level of classifier.

Table III shows the comparison of both structures using the same set of videos. From this table, we can see that *GHOST* can only recognize 8.02 % of lying toward the camera events. In *GHOST,* the first level of classification includes bending and side lying. The side lying posture is then classified as right lying, left lying and lying toward the camera. During classification, it happens that most of lying toward the camera is detected as bending, and hence it results in low recognition rate of lying toward the camera events. This is because projection histogram of lying toward the camera is closer to bending. On the other hand, our modified structure is able to recognize up to 93.21 % of lying toward the camera events. This high recognition rate is because the inclusion of lying toward templates in the first level of classifier attracts the decision of classifier toward choosing lying toward the camera posture. We can see that there is a significant improvement in sensing lying toward the camera events using proposed classifier structure compared to *GHOST* structure.

TABLE II. CONFUSION MATRIX OF THE CLASSIFIER OUTPUT

	Classified As				
Events	Standing	Bending	Sitting	Lying	Lying toward
Standing	**355**	0	10	0	0
Bending	0	**67**	0	0	6
Sitting	0	3	**65**	0	0
Lying	0	0	0	**75**	0
Lying toward	0	11	0	0	**151**

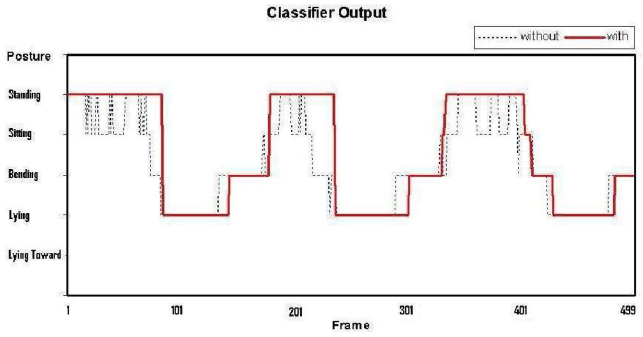

Fig. 4. Comparison between output of classifier with and without evidence accumulation mechanism

TABLE III. COMPARISON BETWEEN GHOST STRUCTURE AND PROPOSED STRUCTURE IN DETECTING LYING TOWARD THE CAMERA EVENTS

	GHOST structure			Proposed structure		
N_a	N_c	N_f	*R (%)*	N_c	N_f	*R (%)*
162	13	149	**8.02**	151	11	**93.21**

IV. CONCLUSIONS AND FUTURE WORK

This paper proposes a method for monitoring human posture-based events in a home environment with focus on detecting two types of fall which are side and fall toward the camera. The use of *k-NN* that represents postures with multiple templates has been shown to have a high recognition rate of about 90%. Evidence accumulation can smooth out the instability in the frame-wise classifier's output. The proposed modified structure is also exhibiting a significant improvement in detecting events of lying toward the camera compared to hierarchical structure in [1]. Future works will include the incorporation of multiple elderly monitoring which is able to monitor more than one person in the scene and also able to handle occlusions. The use of multiple cameras as in [3] is also a subject to be explored in the future work since it can make the system more robust to view point changes and scene occlusions.

REFERENCES

[1] I. Haritaoglu, D. Harwood, and L.S. Davis, "Ghost: A Human Body Part Labeling System Using Silhouettes," *14th International Conference On Pattern Recognition 1998.*

[2] R. Cucchiara, A. Pratti, and R. Vezzani, "An Intelligent Surveillance System for Dangerous Situation Detection in Home Environments," in Intelligenza Artificiale, vol. 1, n.1, pp. 11-15, 2004.

[3] R. Cucchiara, A. Prati, R. Vezzani, "Posture Classification in a Multi-camera Indoor Environment" in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 1, Genova, Italy, pp. 725-728, 11-14 Sept., 2005

[4] Tao, J., M. Turjo, M.-F. Wong, M. Wang, and Y.-P. Tan, "Fall Incidents Detection for Intelligent Video Surveillance," *ICICS* 2005.

[5] H. Nait-Charif and S.J. McKenna, "Activity Summarisation and Fall Detection in a Supportive Home Environment," *Proc. of the 17th Int. Conf. on Pattern Recognition*, Aug. 2004.

[6] C. Rougier, J. Meunier, A. St-Arnaud, J. Rousseau, "Monocular 3D Head Tracking to Detect Falls of Elderly People," *International Conference of the IEEE Engineering in Medicine and Biology Society*, Sept. 2006.

[7] S.-G. Miaou, P.-H. Sung, and C.-Y. Huang, "A Customized Human Fall Detection System Using Omni-Camera Images and Personal Information," *Proc. of Distributed Diagnosis and Home Healthcare (D2H2) Conference*, 2006.

[8] C. Stauffer, and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-time Tracking," *Computer Vision and Pattern Recognition*, 1999.

[9] R.O. Duda, P.E. Hart, and D.G. Stork, "Pattern Classification. 2nd ed ," 2001: Wiley Interscience.