Multimed Tools Appl (2011) 51:697–721
DOI 10.1007/s11042-010-0649-1

Effective multimedia surveillance using a human-centric approach

Pradeep K. Atrey · Abdulmotaleb El Saddik · Mohan S. Kankanhalli

Published online: 12 November 2010

Abstract Large-scale multimedia surveillance installations usually consist of a number of spatially distributed video cameras that are installed in a premise and are connected to a central control station, where human operators (e.g., security personnel) remotely monitor the scene images captured by the cameras. In the majority of these systems the ratio of human operators to the number of camera views is very low. This potentially raises the problem that some important events may be missed. Studies have shown that a human operator can effectively monitor only four camera views. Moreover, the visual attention of human operator drops below the acceptable level while performing the task of visual monitoring. Therefore, there is a need for the selection of the four most relevant camera views at a given time instant. This paper proposes a human-centric approach to solve the problem of dynamically selecting and scheduling the four best camera views. In the proposed approach we use a feedback camera to observe the human monitoring the surveillance camera feeds. Using this information, the system computes the operator's attention to the camera views to automatically determine the importance of events being captured

This work was supported by the National Sciences and Engineering Research Council of Canada Discovery Grant 408206 and the University of Winnipeg Major Research Grant 607062.

P. K. Atrey (✉)
Department of Applied Computer Science, University of Winnipeg,
Winnipeg, MB, Canada R3B 2E9
e-mail: p.atrey@uwinnipeg.ca

A. El Saddik
Multimedia Communications Research Laboratory, University of Ottawa,
800 King Edward, Ottawa, ON, Canada K1N 6N5
e-mail: abed@mrclab.uottawa.ca

M. S. Kankanhalli
School of Computing, National University of Singapore, Kent Ridge, 117417, Singapore
e-mail: mohan@comp.nus.edu.sg

by the respective cameras. This real-time non-invasive relevance feedback is then augmented with the automatic detection of events to compute the four best feeds. The experiments show the effectiveness of the proposed approach by improving the identification of important events occurring in the environment.

Keywords Human-centered · Multimedia surveillance · Relevance feedback · Eyes tracking · Importance computation · Camera view selection and scheduling

1 Introduction

Today traditional analog CCTV surveillance systems are being replaced with the digital multimedia surveillance systems. These systems are commonly used in public places such as university, shopping malls and airports to record and monitor people's activities. Despite of being digital these systems are still often used in traditional manner. Numerous cameras installed at distributed places are connected to a central control station, where a human operator (e.g., a security personnel) remotely monitors the images captured by the cameras.

It has been observed that in a majority of these systems the number of surveillance cameras is very high (few hundreds), while the number of human operators who watch the camera images in the control room is quite less. A study has shown that the ratio of operators to cameras can be as low as 1:16 [8]. Also, a human operators visual attention drops below the acceptable level while performing the task of visual monitoring [9]. Besides, it has been found that a human operator can only effectively monitor four camera views at a time [29]. This practical constraint raises the issue which four camera views, out of several views, should be selected by the system for display at a particular time instant. We assume that the four selected camera views are displayed adjacent to each other at a higher resolution. As the "importance" of the events captured by the cameras usually changes over time, the selection of the most relevant camera views should be performed at a regular interval, which makes the problem of scheduling the camera views a challenging one.

In the context of surveillance and monitoring, the importance of an event can be perceived as the degree of deviation of the new observations from the normal happenings in the environment. While on one hand, the important events can be detected by the traditional computer vision techniques including feature extraction and classification; on the other hand, the importance of an event can also be determined by observing the human operator's attention to the respective camera views.

In this paper, we essentially address the problem of scheduling camera views for real time monitoring. The core idea of our approach is to adopt a human-centric approach in which the system computes human operator's attention to the camera views to automatically determine the importance of events captured by the respective cameras. Our aim is to reduce the human interaction and to make the monitoring process unobtrusive. To achieve this, the proposed method advocates to use a camera to capture the operator's watching behavior. In particular, the operator's eyes are tracked and his or her attention towards one of the four camera views is determined. Based on the attention of the operator, the importance of the camera views is computed and then the camera views are scheduled accordingly for display at a higher resolution.

The existing approaches to determine the important camera views include change detection in the subsequent video frames assuming that an event would trigger a change. The change-detection is performed by using different techniques such as frame differencing and foreground/background subtraction [18]. Although the change detection method may provide the basis of the initial selection of important camera views, it does not always reflect their true importance as this method can only indicate whether or not a change has occurred. This change may not always be a meaningful event. Moreover, it does not provide information about how important the detected change is i.e., whether it is corresponding to a normal or suspicious event. Therefore, to overcome these limitations we propose to follow a human-centric approach, which determines the importance of views based on a human operator's monitoring behavior. In our approach, the initial selection of relevant camera views is performed based on machine-enabled automatic event detection. Then this selection is validated by a human's monitoring and feedback. We envision that the proposed method can assist a human operator and reduce the burden of monitoring several camera views at a time. Our method has an added advantage of tagging important events and store them in a database, which makes the retrieval of these events easier.

To the best of our knowledge, the idea of using a feedback camera to observe human monitoring behavior for determining the importance of camera views in a CCTV surveillance system was introduced by Atrey et al. [4]. This work is an extension of [4] in the following ways. In [4], the authors have used change detection method for initial selection of relevant camera views; while in this work we employ event detection techniques to determine whether the event is normal or abnormal and accordingly assign machine-enabled importance level to them. Furthermore, this importance level is augmented by the importance computed based on human monitoring and feedback. Another difference between [4] and this work is that in [4], the scheduling is done based on a simple time-based strategy while in this work we determine the next relevant camera view based not only on their importance levels but also the spatial relationship among different camera views which we represent using a camera flow graph [3].

The rest of the paper is organized as follows. In Section 2, we discuss the works that are related to the contributions made in this paper. We formulate the research problem in Section 3. In Section 4, we present in detail our method of selecting and scheduling most relevant camera views. Section 5 presents the experiments and results. Finally, we conclude the paper with a discussion on future work in Section 6.

2 Related work

Since the main idea in this paper is to use a human's attention in selecting and scheduling the important camera views, we discuss here the past works that are related to human attention modeling.

There have been several works on human attention modeling in multimedia as well as non-multimedia research communities. For instance, neuroscience researchers Itti and Koch [12] presented a computational model for visual attention. Later, Itti and Baldi [11] also studied the effect of surprise on human attention. Also, the mathematicians Taylor and Fragopanagos [25] adopted an engineering control approach to model human attention and emotions. Most of the work on human

attention modeling done by non-multimedia researchers are based on a human's behavioral aspects e.g., eyes movement; while multimedia researchers have mostly modeled attention based on low-level multimedia features such as color, motion, etc that trigger human attention. However, there are a very few researchers, e.g., Wu et al. [31], who have considered both these aspects. Since our work is related to multimedia research here we will elaborate more on the works done by multimedia researchers.

There has recently been an emphasis in the multimedia community on using human feedback for various multimedia applications [7, 20]. Many researchers have modeled human attention for various applications. For example, Ma et al. [15] presented a user attention model for video summarization. In this work, the authors modeled visual attention using different video and audio features. Video features included object and camera motion, static background, and human face; while the audio features were audio silency, speech and music. Similar to [15], Liu et al. [14] derived a human attention model from visual and auditory features and applied it to action movie analysis, while Baumann et al. [6] emulated touch based attention using wearable hapics devices. In other work, Peters et al. [17] studied the effect of eye gaze and blinking for social interaction among computer cartoons. In the surveillance domain, Leykin and Hammoud [13] presented a method for determining human attention field in surveillance videos. Also, Wang et al. [30] presented a experiential sampling method to compute attention. In contrast to all these works which use only multimedia features in computing attention, we select and schedule the camera views based on their importance (or attention) level first by processing the multimedia features and later by tracking the eye of the human operator in an unobtrusive way. In another work, Vaiapury and Kankanhalli [26] presented a method to find interesting images by computing the attention based on media features as well as human feedback. However, in this work, human feedback was obtained by manual intervention in an obtrusive manner, while in the proposed method human feedback is captured in an unobtrusive way.

Table 1 summarizes the past works on attention modeling done by multimedia researchers. In this table, the works are presented from three aspects: (1) whether multimedia features are considered for computing attention? (2) whether human

Table 1 A summary of the past works on attention modeling in multimedia research

The works	Based on		
	Media content processing	Human's behavioral symptoms	Unobtrusiveness in capturing human's feedback
Ma et al. [15]	Yes	No	Not applicable
Peters et al. [17]	No	Yes	Yes
Liu et al. [14]	Yes	No	Not applicable
Vaiapury and Kankanhalli [26]	Yes	Yes	No
Leykin and Hammoud [13]	Yes	No	Not applicable
Baumann et al. [6]	No	Yes	Yes
Atrey et al. [4]	No	Yes	Yes
Wu [31]	Yes	Yes	Yes
Wang et al. [30]	Yes	No	Not applicable
This work	Yes	Yes	Yes

behavioral aspects are considered to compute human attention, and (3) if the answer to (2) is yes, whether human feedback was obtained in an unobtrusive manner? Of course, if the answer to (2) is 'No', the answer to (3) is 'Not applicable'.

From the perspective of considering both multimedia features and human behavioral aspects, our approach is similar to Wu et al. [31]. In this work, the authors considered both the viewpoints in computing attention: the causes that trigger human attention, and the effects (actions, behaviors) that are driven by human attention. The authors used a Bayesian network to integrate the contextual features (e.g., object properties, text characteristics, etc.) and behavioral symptoms (e.g., eye and head movement, fMRI imaging, etc.) in order to compute attention. Our work is different from [31] in that we do not combine the attention computed using these two aspects. Instead we use the importance level (or attention) computed using multimedia features only for the preliminary selection of camera views. Later this importance level is superseded by the importance determined based on a human's behavioral symptoms (i.e., eye movement). This makes sense under the assumption that human observation is often more accurate than the events detected by the machine.

Another work which is closely related to the proposed method is [28]. In this work, Vural and Akgul presented a method to construct the surveillance video synopsis based on eye-gaze. The method involved two steps: first frequency-based background subtraction and then tracking eye-gaze positions of human operator. In this work, the objective was to select important regions in a camera view; however in the proposed work, our objective is to select four most important camera views from many views.

3 Problem formulation

Let **S** be a surveillance system that has a set Γ of n non-overlapping surveilled regions or camera views at distributed places in a premises. Each of these surveilled regions may have different types of multimedia sensors $\mathbf{M} = M_1, M_2, \ldots, M_r$ installed in it. For example, M_1 could be a video camera and M_2 could be an audio sensor. The evidence from each camera view is obtained by processing and assimilating the data from the sensors in that surveilled region.

In the context of this surveillance system, we define the following terms:

- T is the *camera flow graph*. In T, each camera view C_i is represented by a vertex, and two vertices are joined by an edge (i, j) if and only if their associated cameras views C_i and C_j corresponds the adjacent surveilled regions. Here, adjacent surveilled regions are the regions in which people can move directly without passing through any other surveilled region. The label $T_{i,j}$ on the edge joining two vertices C_i and C_j represents the distance (in time units) between the regions that are covered by the corresponding cameras. In other words, people normally move from the region of camera C_i to the region of camera C_j in $T_{i,j}$ time units.
- $\tilde{\Gamma}(t) \subseteq \Gamma$ is the set of higher resolution camera views displayed at time t to be viewed by the human operator. In our case, we assume $|\tilde{\Gamma}(t)| = 4$ [29].
- $I_i(t) \in [0, 1]$, $1 \leq i \leq n$, is the importance level of the C_i camera view at a given time instant t. I_i can have two forms: I_i^{machine} and I_i^{human}, where I_i^{machine} is the importance level of the C_i camera view based on the event detected by the

system, and I_i^{human} is the importance level of the C_i camera view based on a human's feedback.

The objective is to determine a schedule of camera views between time instances t_1 and t_2 such that

$$\sum_{t=t_1}^{t_2} \sum_{j=1}^{|\tilde{\Gamma}(t)|} I_j(t), \qquad C_j(t) \in \tilde{\Gamma}(t) \tag{1}$$

is maximized.

3.1 An illustrative example

We discuss here an example that shows the construction of a camera flow graph. Assume that we have a surveillance system consisting of eight cameras installed in a building having two similar floors as shown in Fig. 1. Each floor has four cameras. Figure 1a and b show the floors 1 and 2, respectively. Cameras C_1 to C_4 are installed at floor 1 and C_5 to C_8 are at floor 2. The coverage area of these cameras is also shown in Fig. 1a and b. These floors are connected through three elevators denoted by E_1, E_2 and E_3. In this layout, it can be seen that the movement of people between different cameras will be distinct. For example, a person presently in the camera view C_1 can move to either of the views of cameras C_2, C_3 and C_4. However, a person

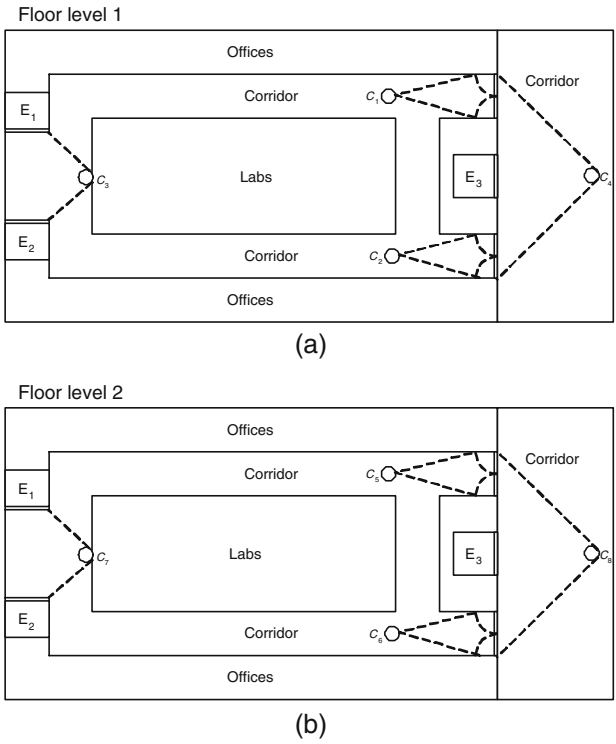


Fig. 1 An example layout of two floors of a building: **a** floor 1; **b** floor 2

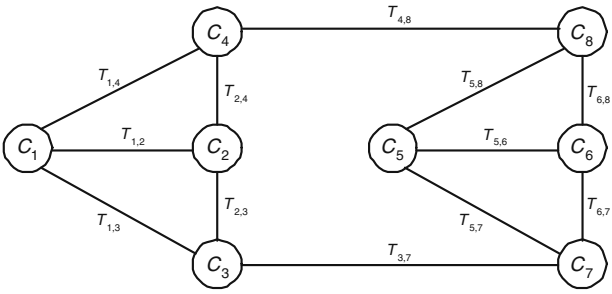


Fig. 2 Camera flow graph for the example shown in Fig. 1

presently in the camera view C_4 can move only to either C_1 and C_2 camera views (on floor 1) or C_8 camera view (on floor 2) via elevator E_3.

Figure 2 shows the camera flow graph which depicts the direct connectivity between cameras. The labels on the edges in the graph show the distance in average time units between the camera views.

4 Proposed method

The proposed method of selecting and scheduling the most relevant camera views uses automatic activity analysis (by machine) and human monitoring and feedback (by a feedback camera). The importance of each camera view is determined based on both these two factors. This is illustrated in Fig. 3.

First, the captured multimedia data is processed by the machine to detect the presence or absence of a meaningful event. By multimedia data we mean that the

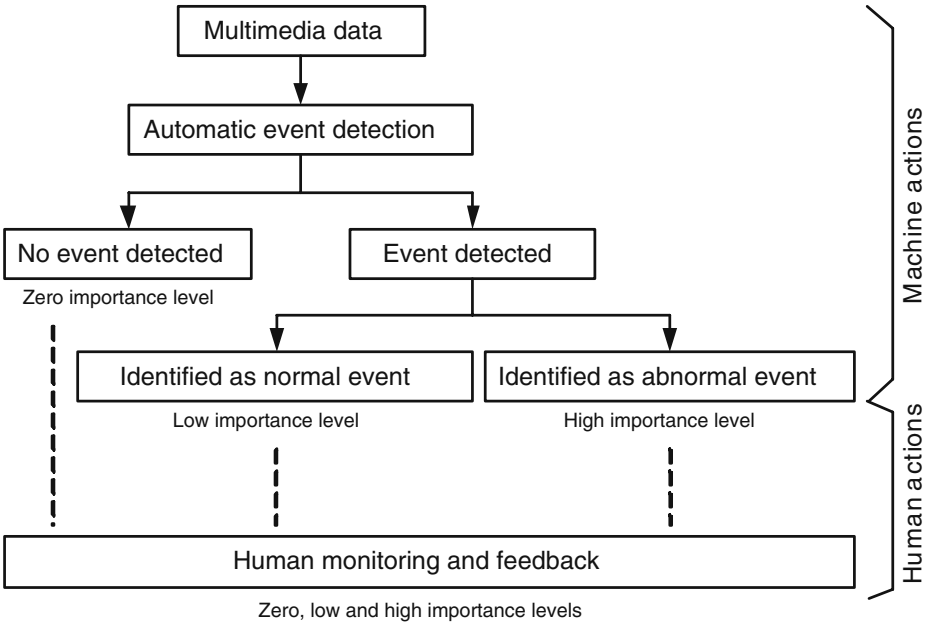


Fig. 3 Event detection hierarchy in the context of importance level

data can be captured via different types of sensors e.g., video camera, audio microphone, etc. If no event is detected in a camera view, a zero importance is assigned to it. However, if an event has been detected, it is further classified into a normal event or a suspicious event. If the system tags this as a normal event, a low importance is assigned to it. This is because a normal event would usually be of less interest from a surveillance perspective. However, if the system classifies an event into a suspicious category, a high importance is attributed to it. After the initial assignment of the importance levels has been done, the four camera views with the maximum importance levels are displayed at a high resolution for a human operator to watch.

Once the four camera views are selected for display, their importance is computed based on the attention of human operator. A feedback camera is used to observe a human's monitoring behavior. If the operator pays more attention to a camera than the others, it is inferred that the event captured by that camera view has a high importance and hence the corresponding camera view would have a high importance. The degree of attention is determined based on the duration for which the operator looks at a particular camera view. The more attention the operator pays to a camera view, the higher the importance it would have. Based on this feedback, importance of these four camera views are updated. Note that, here human operator's feedback for a camera view supersedes any previously assigned importance and it can increase the importance level of a camera view from low to high, and vice versa. This is because that the event detection performed by the system may not always be correct. In the cases where the incorrect detection takes place, it is corrected by the human's feedback.

In the following, we first describe the methods for computing the importance of a camera view based on: 1) automatic event detection (in Section 4.1), and 2) human's monitoring and feedback (in Section 4.2). In Section 4.3, we elaborate on the strategy of camera views transitions. Finally, in Section 4.4, we present the steps to select and schedule the camera views.

4.1 Camera view importance based on automatic event detection

As discussed earlier in Section 3, I_i^{machine} is the importance of a camera view determined based on the machine-identified events. This importance level is inferred from the certainty with which an event is detected in the camera view. In other words, the more the certainty with which an event is detected in a camera view is, the more the importance level that camera view would have. Certainty is an attribute that can be regarded as the probability of the occurrence of the event [10]. Based on whether the event is classified as normal or suspicious, a weight is assigned. Precisely, the importance of a camera view C_i is computed as:

$$I_i^{\text{machine}} = \frac{1}{Z} \times P(E|\mathbf{M})^w \tag{2}$$

where $P(E|\mathbf{M}) \in (0, 1)$ is the probability of occurrence of event E based on the data from multimedia sensors $\mathbf{M}$ and $w > 0$ is the weight used for normal and suspicious events. The value of w is determined as:

$$w = \begin{cases} < 1 \text{ if there is no event or a normal event detected by machine;} \\ > 1 \text{ if an abnormal event is detected by machine.} \end{cases}$$

In (2), Z is a normalization factor, which is given by:

$$Z = P(E|\mathbf{M})^w + (1 - P(E|\mathbf{M}))^w \tag{3}$$

To compute the probability $P(E|\mathbf{M})$, we adopt the multimedia assimilation framework that we proposed in our past work [5]. We describe it briefly as follows.

4.1.1 Multimedia assimilation model

Based on the sensors $M_1, M_2, \ldots, M_r$, the system outputs local decisions $p_k = P(E|M_k)$, $1 \leq k \leq r$, about an event E at a given time instant. $P(E|M_k)$ represents the probability of the occurrence of event E based on the kth sensor data. Along a timeline, as these probabilistic decisions are available, the system iteratively integrates using a Bayesian approach, all the decisions obtained based on the sensor data. The local decisions obtained based on the data of any two sensors M_{k-1} and M_k are assimilated using the following model:

$$p_{k-1,k} = \frac{p_{k-1} \times p_k}{p_{k-1} \times p_k + (1 - p_{k-1}) \times (1 - p_k)} \tag{4}$$

Furthermore, the agreement between different media streams and their confidence levels are also integrated in the assimilation model. We omit further description of assimilation for brevity and readers are referred to [5] for details.

4.2 Camera view importance based on human monitoring and feedback

4.2.1 Eye tracking

The objective of tracking the eyes of the human operator is to determine his/her attention to a particular camera view. A human operator can have four possible eye orientations when paying attention to the four camera views, as shown in Fig. 4. These are top-left, top-right, bottom-left and bottom-right. Note that the straight eyes orientation is not shown in the figure. In the straight eyes orientation, the operator is assumed to have equal attention to all the four camera views; hence this case has been ignored for computing relative attention.

Eye tracking has been an active area of research in the recent past [1, 32] and it has been used for various applications e.g., alerting drivers [23] and e-learning [2]. For eye tracking, we used the open source code TrackEye [22]. TrackEye provides functionalities of face and eye tracking. For face tracking, it uses the following two methods: (1) Continuously adaptive Means-Shift algorithm [27] and (2) Haar face detection method [16]. Eye tracking is also performed using two different methods: (1) Template-matching [19] and (2) Adaptive Eigen-Eye method [21].

The eye's positions are used to determine their orientation. For this, a Bayesian classifier is employed to categorize the given eyes positions into one of the following four orientations: top-left, top-right, bottom-left and bottom-right. The left eye positions for these four orientations are shown in Fig. 5.

Once the orientation of the eyes of the human operator is determined, the system observes the duration for which the operator continuously looks at a particular camera view. The longer this duration is, the higher the importance the camera view would have.

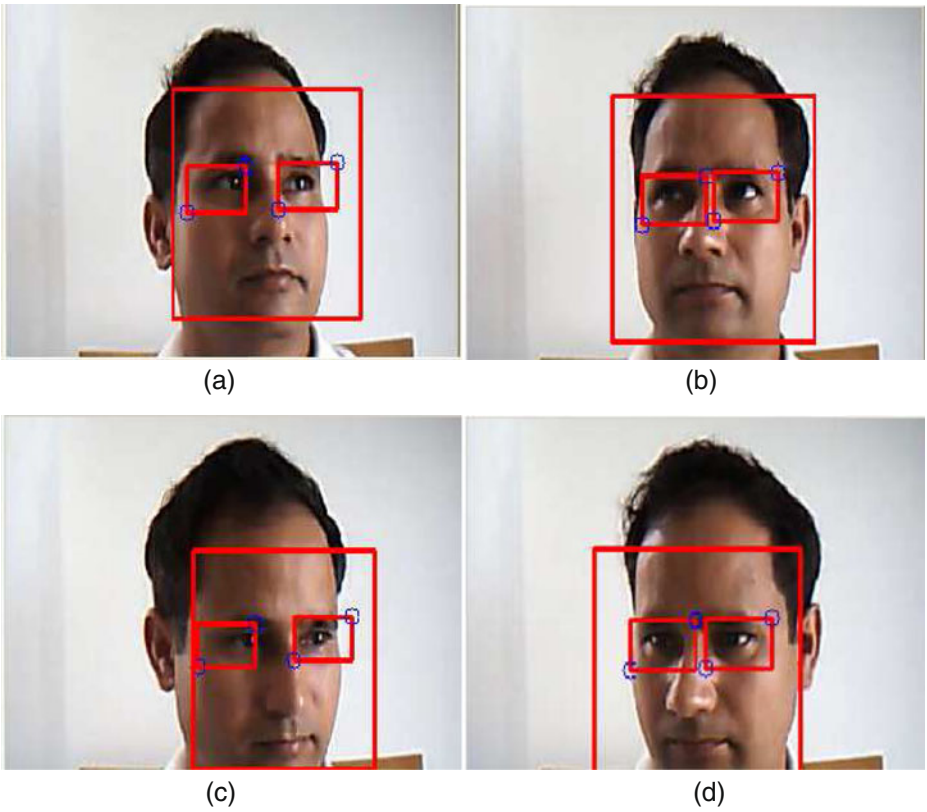
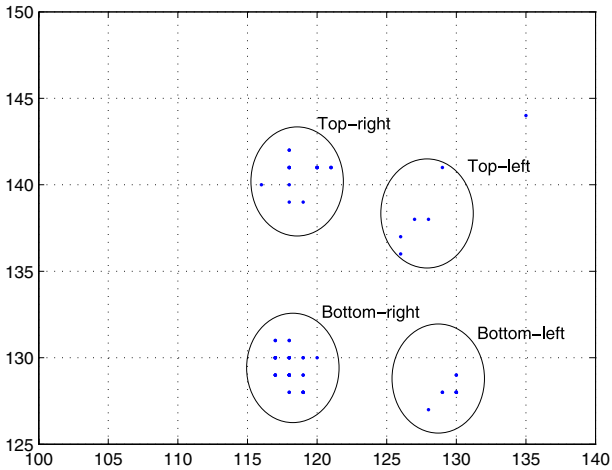


Fig. 4 Four eyes orientations for four camera views: **a** top-left; **b** top-right; **c** bottom-left; **d** bottom-right

Fig. 5 Left eye positions for four orientations



4.2.2 Importance computation

To quantify the amount of importance of a particular camera view, a *time-based strategy* has been adopted, which works as follows. Let Δ be the minimum time period for which the four camera views remain persistent once displayed. Normally, the operator, when looking straight, pays equal attention to all of the four views. However, once the operator observes an important event on a particular camera view, s/he starts concentrating on it. Let the operator spend $\gamma \leq \Delta$ time in looking straight. The rest of the time $(\Delta - \gamma)$ is used to find the attention of the operator in a particular view. Adopting the strategy that the operator would have attention to a particular camera view if s/he concentrates on it for a time period $\delta \geq (\Delta - \gamma)/|\tilde{\Gamma}(t)|$, the importance $I_i^{\text{human}} \in (0, 1)$ of a particular camera view C_i $(1 \leq i \leq |\tilde{\Gamma}(t)|)$ is computed using the following linear model:

$$I_i^{\text{human}} = \frac{|\tilde{\Gamma}(t)| \times \delta_i/(\Delta - \gamma) - 1}{|\tilde{\Gamma}(t)| - 1} \tag{5}$$

In the above model, the bounding values 0 and 1 of I_i^{human} are obtained at $\delta = (\Delta - \gamma)/|\tilde{\Gamma}(t)|$ and $\delta = \Delta - \gamma$, respectively.

4.3 Camera view transitions

The four most important camera views change over time as their importance levels change. This is determined based on automatic activity analysis and human monitoring and feedback. The transition of camera views is modeled by a state transition model as follows. The camera views $C_i, 1 \leq i \leq n$ represent n states, out of which four most important camera views represent the four important states. In this model, the state transition probability determines the probability of a camera view replacing the other camera view as one of the four most important camera views. Precisely, the probability $P(C_j|C_i)$ that the camera view C_j replaces the camera view C_i as one of the four most important camera views is computed as follows:

$$P(C_j|C_i) = \beta \times I_i^{\text{human}} + (1 - \beta)\times) I_j^{\text{machine}} \tag{6}$$

C_j is replaced by C_i after $T(i, j)$ time units with a probability $P(C_j|C_i)$. In (6), β is the relative weight assigned to human feedback over automatic event detection.

4.4 Selection and scheduling of camera views

The system performs the following steps to schedule the camera views:

1. For each of the camera views, apply event detection method on the multimedia data and compute $P(E|\mathbf{M})$ using the assimilation method given in [5]. Select $|\tilde{\Gamma}(t)|$ camera views that have maximum I_i^{machine}. If there are more than $|\tilde{\Gamma}(t)|$ views having the same I_i^{machine}, randomly select any $|\tilde{\Gamma}(t)|$ views from them.
2. Display the images of the selected $|\tilde{\Gamma}(t)|$ camera views for Δ duration. During this period, record the operator's watching behavior. For this purpose, the system uses a separate camera. The operator's facial images are processed and the eye positions are detected. From the position of the eyes, their orientation is

determined as described in Section 4.2.1. Based on the eyes orientation, the importance I_i^{human} of each of the $|\tilde{\Gamma}(t)|$ camera views is computed using (5).

3. The camera views that have importance $I_i^{\text{human}} > 0$ remain persistent for some time and then they are replaced by the camera view C_j that have maximum transition probability $P(C_j|C_i)$, $1 \leq j \leq n$, $j \neq i$, after $T_{i,j}$ time units.
4. Step 3 is continued until any of the camera views has some importance. If all the views are reset to the zero importance level, Step 1 is followed.

5 Experiments and results

5.1 Experimental setup and data set

To show the utility of the proposed method, we simulated a surveillance control room environment in which we assumed that 16 cameras were connected to a central control station and the four of these cameras were displayed at a higher resolution. We recorded audio-visual data in four corridors. Each corridor consisted one video camera (Canon VC-C50i) and one USB microphone as shown in Fig. 6. The volunteers were asked to perform various normal as well as abnormal activities. Normal activities include human walking (**W**), standing (**S**), talking (**T**) and door knocking (**K**); while the human running (**R**), shouting (**H**) and abandoning baggage (**A**) were the abnormal/suspicious activities.

The video cameras were placed in a way so that they covered the whole corridor. The unidirectional microphones were employed to capture the ambient sound. The cameras and the microphones were connected to a central PC and a Picolo-Pro video card was used to capture the image data. The data collected from these four corridors was chopped off into 16 parts to simulate 16 cameras. Figure 6 shows one floor of the building consisting of four corridors. The chopped data was assumed to be distributed over three other similar floors. These floors were connected with each other via stairs near their doors denoted by 'A' and 'B' in the figure. The camera flow graph for the four floors of the surveilled building is shown in Fig. 7. As shown in this figure, the four floors are connected to each other via camera views C_4, C_8, C_{12} and C_{16} on door 'A' side and C_3, C_7, C_{11} and C_{15} on door 'B' side.

The whole data set consisted of the recorded video images (of resolution 768 × 576 pixels at 1 frame per second) and the recorded audio (of 44.1 kHz) for more than twelve hours. A total of 119 events occurred during the twelve hours of recording. Table 2 shows the details of various events occurred in different camera views.

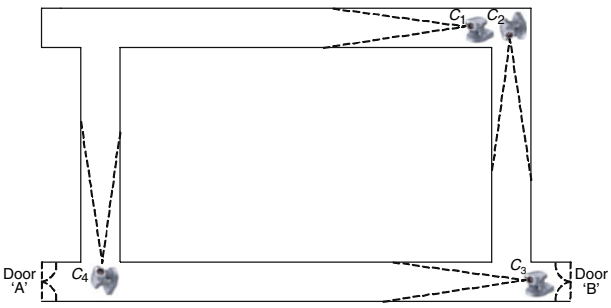


Fig. 6 Environment layout of one floor of the building

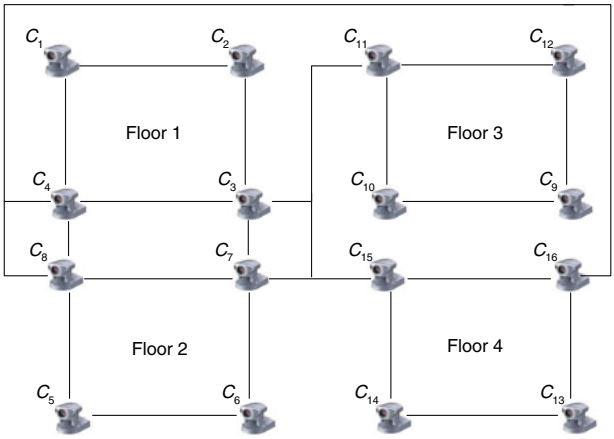


Fig. 7 Camera flow graph for the four floors of the surveilled building

The audio-visual data of all 16 camera views was processed to automatically detect the events, as will be described in Section 5.2, and subsequently the I^{machine} for these camera views was determined. The recorded video frames were played on a simulated camera panel (an electronic board of $4.5' \times 3'$ size) and a human operator was asked to monitor these images. A separate feedback camera was used to capture the eye movement of the human operator. The images of this camera were processed for the detection of eye position. Based on this information, the orientation of the eyes of the operator was determined (using the strategy discussed in Section 4.2.1), and then the proposed steps (described in Section 4.4) were used for selecting and scheduling the four camera views.

The test bed was developed using Microsoft Visual Studio on Windows platform. The GUI of our test bed is shown in Fig. 8. It consists of two main parts. The lower

Table 2 Events occurred in 16 camera views

Camera view	Number of events							Total
	S	**W**	**T**	**K**	**R**	**H**	**A**	
C_1	0	7	0	0	0	0	0	7
C_2	0	4	0	0	0	0	0	4
C_3	0	4	0	0	2	0	0	6
C_4	0	2	0	2	2	0	0	6
C_5	0	4	0	4	0	0	0	8
C_6	0	3	0	0	0	1	1	5
C_7	0	7	0	0	1	0	0	8
C_8	0	4	2	0	2	0	0	8
C_9	0	7	0	0	0	0	0	7
C_{10}	0	5	0	0	0	0	0	5
C_{11}	2	5	2	0	1	0	0	10
C_{12}	3	3	0	3	1	0	0	10
C_{13}	3	4	0	3	0	0	0	10
C_{14}	0	7	0	0	0	4	1	12
C_{15}	0	5	0	0	0	0	0	5
C_{16}	2	4	2	0	0	0	0	8
Total	10	75	6	12	9	5	2	119

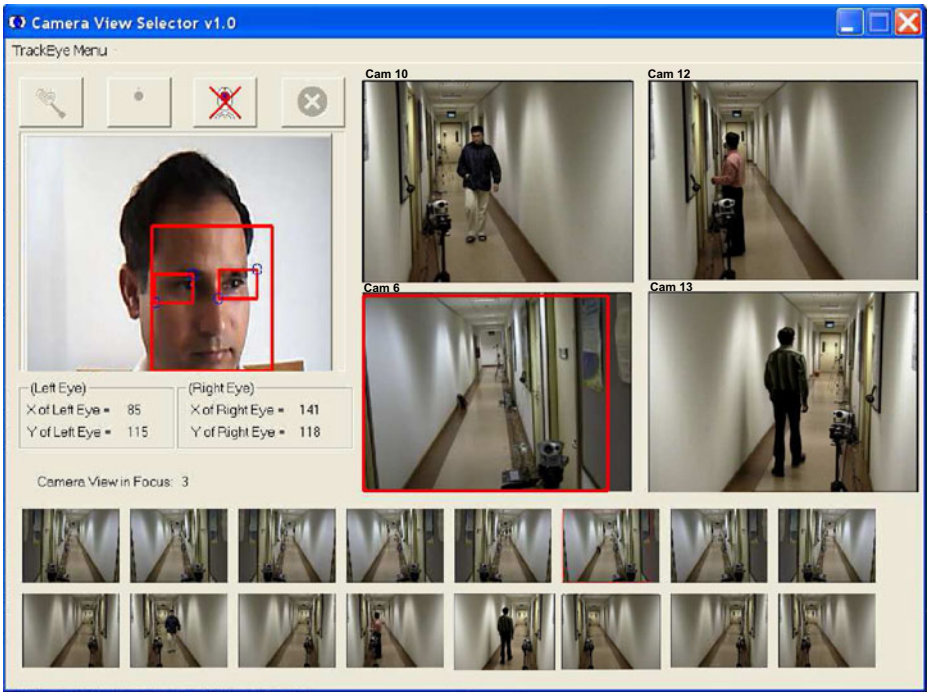


Fig. 8 The GUI of our test bed

part shows 16 small camera views, whereas the upper part displays—(1) the image of the operator showing his eyes movement, and (2) the best four camera views selected and scheduled using the proposed method. Based on the eyes orientation of the operator, one of the four camera views to which he pays more attention is highlighted with a rectangle on it (bottom-left camera view, Cam 6). The snapshot shown in Fig. 8 illustrates an instance when the operator is looking at the bottom-left camera view where an abandoned object was found.

The values of different parameters used in the experiments are given in Table 3.

5.2 Automatic event detection and I^{machine} computation

The human standing event and the presence of abandoned object were detected by processing the video data, while the human talking, shouting and door knocking events were detected using the audio data. The events of human walking and running were detected using both audio and video data. The audio-visual data processing and

Table 3 The values of different parameters used in the experiments

Parameter	Value
Δ	20 s
β	0.70
w for normal events	0.5
w for suspicious events	1.5

event detection steps are briefly described in the following subsections. For details, readers are referred to [5].

5.2.1 Event detection based on video stream

One of the primary steps for event detection in video is background/foreground modeling, which was performed using an adaptive Gaussian method [24]. Blob detection is performed by first segmenting the foreground from the background, and then by using the morphological operations (erode and dilation) to obtain connected components (i.e., blobs). The foreground segmentation is achieved by matching the three RGB color channels of the pixels within 2.5 standard deviations of the distribution. Based on the placement of cameras, the minimum area of blobs that would correspond to a human is manually determined.

Once the blobs have been detected in the video frames, their bounding rectangle is computed. The middle point of the bottom edge of the bounding rectangle is mapped to the actual ground location using the calibration information of the video cameras. This provides the exact ground location of a human in the corridor at a particular time instant.

Similar to [5], the system detects events of humans' standing/walking/running by processing the video frames. Based on the average distance traveled by humans on the ground in one second, a Bayesian classifier is first trained and then used to provide a probabilistic decision in favor of one of the classes—standing, walking and running.

To detect the abandoned object, we assume that—(1) the size of blob corresponding to an abandoned object is smaller than the size of the blob corresponding to a human, and (2) the blob corresponding to an abandoned object remains static in the video frame for a certain period. We employ a Bayesian classifier to categorize a given blob into a normal or abandoned object and obtain a probabilistic decision.

5.2.2 Event detection based on audio stream

To detect the events such as footsteps, talking, shouting and door knocking, first the recorded audio was divided into "audio frames" of 50 ms each. The frame size is chosen by experimentally observing that 50 ms is the minimum period during which an event, such as a footstep, can be represented. Then, a hierarchical (top-down) approach was adopted to model these events. The audio features such as the Log Frequency Cepstral Coefficient (LFCC) and the Linear Predictive Coefficient (LPC) were used at different levels of classification. The Gaussian Mixture Model (GMM) classifier is employed to classify every audio frame (of 50 ms) into audio events at different levels and the probabilistic decisions are obtained. At the top level, each input audio frame is classified as the foreground or the background based on the LFCC audio feature. The background is the environment noise, which represents 'no event' and is ignored. At the next level using LFCC, the foreground that represents the events are further categorized into two classes—vocal and nonvocal. At the next level, both vocal and nonvocal events are further classified into "talking/shouting" events based on LPC and the "footsteps/door knocking" events based on LFCC. Finally, at the last level, the footstep sequences are classified as "walking" or "running" based on the frequency of their occurrence in a specified time interval and the energy of the footstep sound samples.

5.2.3 Assimilation

The probabilistic decisions obtained based on different audio and video streams are assimilated every second for making an overall decision about the events. These overall decisions in each of the camera views are used to compute the I_i^{machine}, $1 \leq i \leq n$, using (2).

5.3 Results of human monitoring augmented with automatic event detection

In this section, we discuss the results from three different perspectives. First we present the results of camera view scheduling for the first 100 frames (see Section 5.3.1). Next we show the instances that demonstrates the utility of human monitoring over the automatic event detection (see Section 5.3.2). Finally, we present overall results to establish that the proposed method of selecting and scheduling the camera views helps the human operator in better identification of events occurring in the environment (see Section 5.3.3).

5.3.1 Camera view scheduling

The result of applying the proposed camera view scheduling method to first 100 frames of all the 16 cameras is shown in Fig. 9. In the figure, *x*-axis shows the frame number and *y*-axis represents the 16 camera views. The blue solid lines denote the events occurring in the camera views. The letters marked over these lines show the

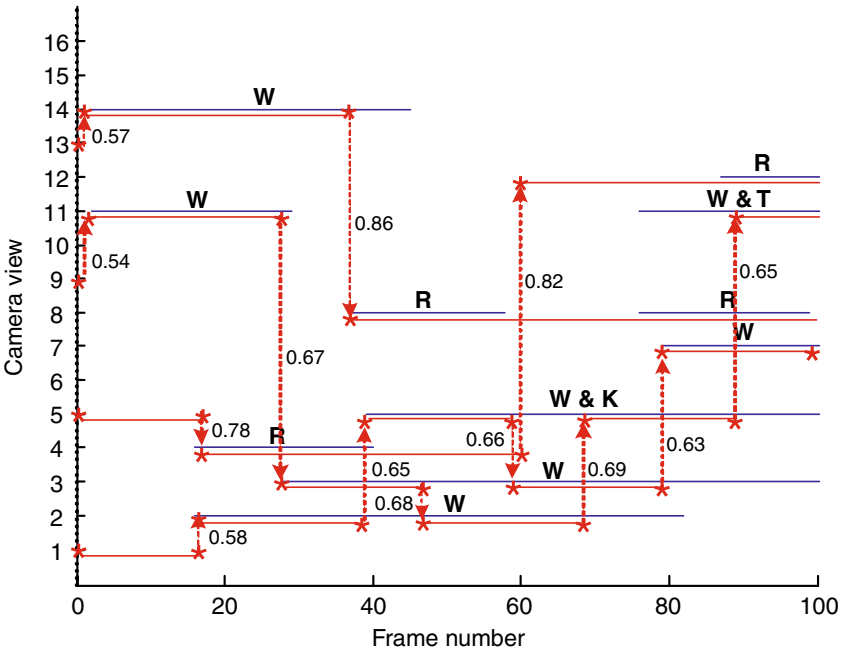


Fig. 9 Camera view schedule between frames 0 to 100

type of events, e.g., **W** indicates a walking event and **W & T** shows a walking-and-talking event. The dotted lines show the transition from one camera view to the other.

The scheduling starts with $\tilde{\Gamma} = \{C_1, C_5, C_9, C_{13}\}$ being the first four camera views randomly selected to be displayed at a higher resolution. At the next instant (frame 2), the system identifies that there are walking events occurring in the C_{11} and C_{14} views. which results in transition from C_9 to C_{11} and from C_{13} to C_{14} with transition probabilities $P(C_{11}|C_9) = 0.54$ and $P(C_{14}|C_{13}) = 0.57$, respectively (computed using (6)). The camera views C_1 and C_5 continued as there were no other events flagged by the system. At frame 16, these two camera views changed to C_2 and C_4, respectively, after the system reported two events **W** and **R** in those views. Note that the transition probability from C_5 to C_4 ($P(C_4|C_5)$) was 0.78, which was relatively higher than the others such as $P(C_{11}|C_9)$. This was because the system reported I_4^{machine} to be high as this was an abnormal event (running). The transition probability was even higher (0.86) in the case of transition from C_{14} to C_8 (at frame 27) since in this case I_{14}^{human} was also higher as the operator was looking at the walking event occurring in C_{14}.

Another interesting point to note in Fig. 9 is that the camera view C_4 continued even after frame 40 (when the running event was over). This was because that the running event was considered suspicious by the operator and he continued giving attention to this view even when the event finished. The C_4 changed to C_{12} at frame 60 as the C_{12} was adjacent to C_4 (see Fig. 7). Subsequently, the operator found another running event in C_{12} at frame 87. In fact, C_8 was also adjacent to C_4 but the transition from C_4 to C_8 did not take place because C_8 was already among $\tilde{\Gamma}$ at that time.

In summary, as can be seen in Fig. 9, the proposed camera view scheduling method did not miss any important event in the span of first 100 frames. Note that although we show here the scheduling for only first 100 frames, similar results were obtained across whole data set.

5.3.2 Specific cases

Here we show specific instances that establish the utility of a human's monitoring and feedback. Ideally, if the system reports an event to be normal, a human operator should also confirm it to be the same, and vice versa. In other words, $I^{\text{machine}} \downarrow \Rightarrow I^{\text{human}} \downarrow$ and $I^{\text{machine}} \uparrow \Rightarrow I^{\text{human}} \uparrow$. However, since existing event detection methods are always not 100% accurate, there could be a few instances when the system reports incorrectly. In such a case, the human operator should be able to correct it.

We illustrate this with four cases, out of which, two (Case 1 and Case 2) are normal cases in which human's feedback was in concordance with the system's decision. Other two (Case 3 and Case 4) are the cases where the system provided a faulty decision and it was corrected by the human operator's feedback. These cases are described as follows:

Case 1: $I^{\text{machine}} \downarrow \Rightarrow I^{\text{human}} \downarrow$. In this case, we present an instance when system detects a walking event to be of low importance (between 0.50 to 0.70) and accordingly the camera view containing that event is selected for display. The human operator looks at it and s/he confirms the same, and consequently the camera view is removed from the display after few frames.

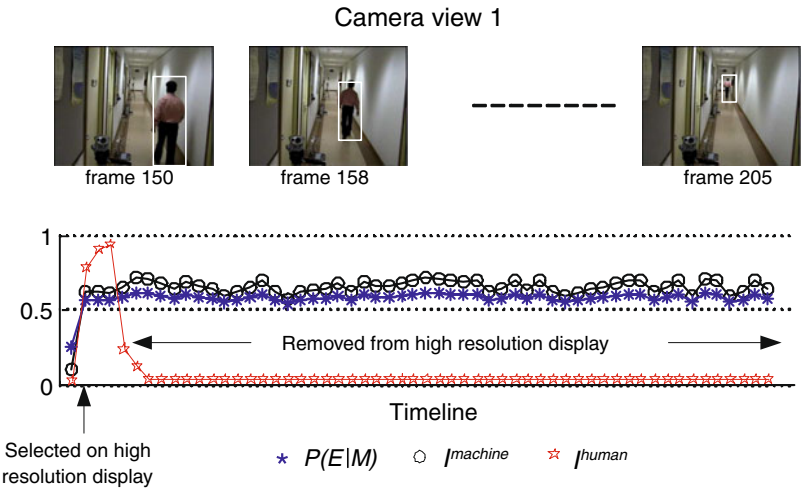


Fig. 10 $I^{\text{machine}} \downarrow \Rightarrow I^{\text{human}} \downarrow$

This is illustrated in Fig. 10. Note that this and next three figures show the following three plots: $P(E|\mathbf{M})$—the probability of the occurrence of the event, I^{machine}—the importance of the event detected by the system, and I^{human}—the importance of the event based on human's feedback.

Case 2: $I^{\text{machine}} \uparrow \Rightarrow I^{\text{human}} \uparrow$. Figure 11 shows an instance when system detects an abandoned object and tags it as a highly important event. Subsequently, it is confirmed by the feedback of human operator, and consequently the camera view containing this event continues to remain on the display.

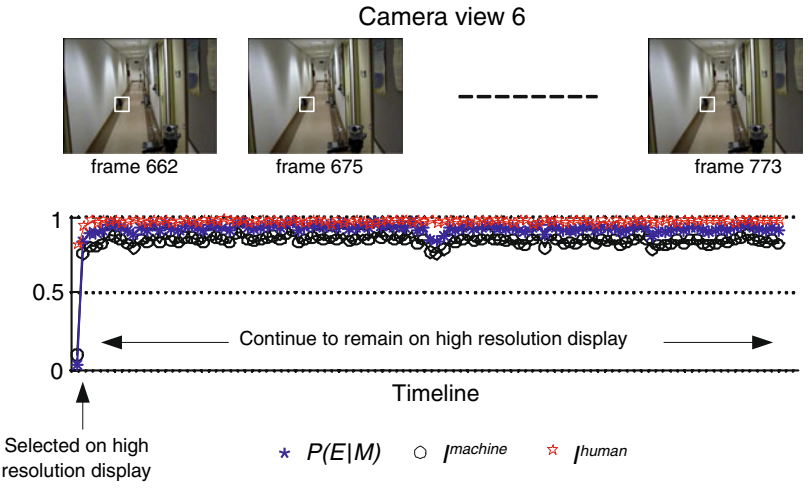


Fig. 11 $I^{\text{machine}} \uparrow \Rightarrow I^{\text{human}} \uparrow$

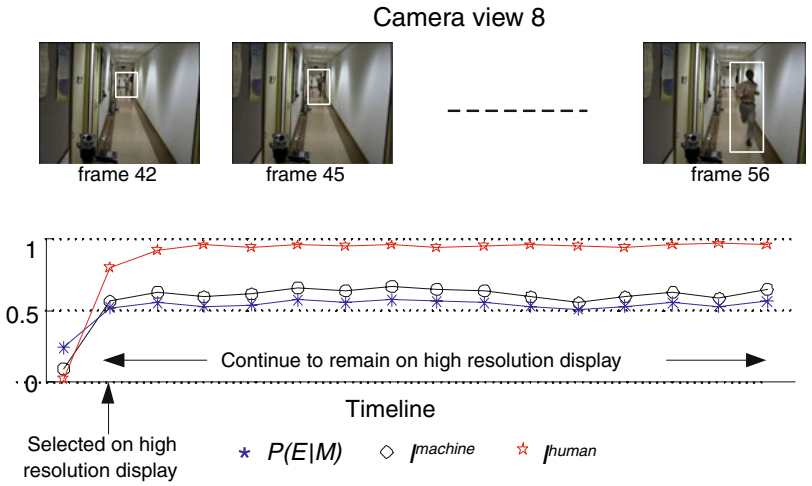


Fig. 12 $I^{\text{machine}} \downarrow \Rightarrow I^{\text{human}} \uparrow$

Case 3: $I^{\text{machine}} \downarrow \Rightarrow I^{\text{human}} \uparrow$. This case shows an instance when system incorrectly detects a running event (of high importance) as a walking event (of low importance). Subsequently, it is corrected based on human operator's monitoring and feedback, as depicted in Fig. 12.

Case 4: $I^{\text{machine}} \uparrow \Rightarrow I^{\text{human}} \downarrow$. This is another case where the system incorrectly identifies a walking event (of low importance) as a running event (of high importance). Once the human operators looks at it, s/he pays less attention to it and consequently the camera view is removed from the display. This is shown in Fig. 13.

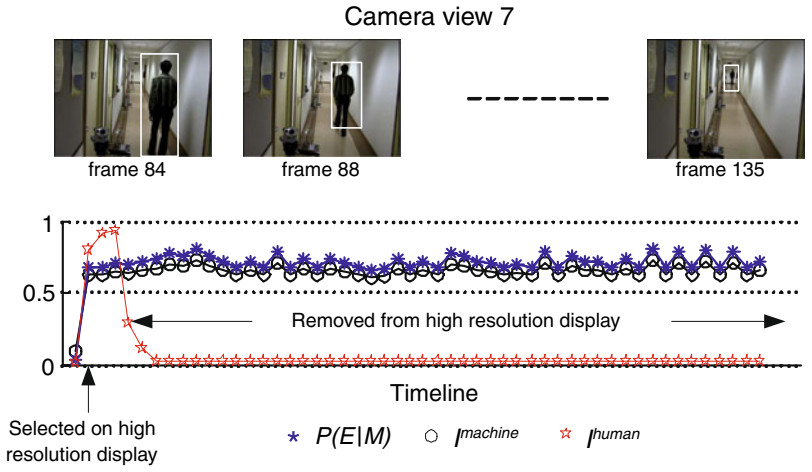


Fig. 13 $I^{\text{machine}} \uparrow \Rightarrow I^{\text{human}} \downarrow$

5.3.3 Events: identified versus missed

In what follows, we demonstrate the advantage of the proposed method by performing the following three test cases with an objective of identifying important events. In case 1, important events were identified using automatic event detection methods (as described in Section 5.2). In the second case, we examined how a human operator's feedback can help in identifying important events. In this case, we did not use the automatic event detection methods; and the camera views were selected and displayed at a high resolution display based on a simple change detection method [18]. The human operator was assumed to be knowledgeable of the important events (e.g., a person abandoning a bag in the corridor and/or a person shouting in the corridor). Case 3 involved inputs from both automatic event detection methods as well as the human operator. In this case, first the system computed the I^{machine} by detecting the important events. These important events were displayed at a high resolution display for the human operator to monitor. Based on the attention of the operator, the importance levels of the camera views I^{human} were calculated using (5). The results of these three cases are shown in Table 4.

As presented earlier in Table 2, out of a total 119 events, 16 events were of high importance (i.e., suspicious events) and the rest 103 were less important events (i.e., normal events). We analyzed that, out of 103 normal events, system could correctly detect only 71 events (68.9%). However, when the human operator was asked to monitor the events that were selected and displayed at a high resolution screen using a change detection method, the operator could correctly identify only 63 (61.1%) of the normal events. In this case, identification was poor due to the limitations of the change detection method. This is because at few instances the change-detection method failed to identify the important events as the change in the subsequent frames were found below threshold. On the other hand, when the human monitoring was augmented with the automatic event detection methods, the number of correctly identified events increased to 89 (86.4%). Similar trends were recorded for the events of high importance, though the number of correct identifications were slighly higher. This was due to the fact that suspicious events attracted the attention of the operator more than the normal events.

5.3.4 Eye tracking versus mouse clicking

The other experiment we performed was to examine the utility of the automatic eye tracking method (unobtrusive) over the traditional method (obtrusive) in which the operator identifies important events by mouse clicking. In both cases, attention of the operator was captured. In the traditional method, all the 16 camera views were monitored simultaneously; while in the eyes tracking method, only the four best

Table 4 Results of three test cases: (1) automatic event detection by the system, (2) human monitoring and feedback, and (3) both automatic event detection and human monitoring

Case	Number of less important events		Number of highly important events	
	Identified (%)	Missed (%)	Identified (%)	Missed (%)
1	68.9	31.1	70.5	29.5
2	61.1	38.9	64.2	35.8
3	86.4	13.6	87.5	12.5

Table 5 A comparison of eye tracking versus mouse clicking for camera view selection

Case	Number of events	
	Identified (%)	Missed (%)
The proposed method with the best four camera views monitored (view selection was through eye tracking)	86.6	13.4
Traditional approach when all 16 camera views monitored (view selection was by operator's clicks)	40.3	59.7

camera views that were selected and scheduled using our method were monitored. In both the cases, the operator was asked to record important events. While in the traditional method, the operator recorded the important events by clicking on the camera views; in the other case, they were recorded automatically based on the importance value computed using the proposed method (see Section 4.2). The results of these two cases are shown in Table 5.

In this experiment, we analyzed that out of 119 events, 103 events (86.6%) were identified by the operator when eye tracking was used; while only 48 events (40.3%) could be identified using the traditional monitoring approach. While using the traditional approach, the operator missed several important events, as he could not pay attention to all of the 16 camera views at once. However, with the eye tracking method, a set of events were initially displayed on the four camera views, which helped the operator in identifying important events. With the eye tracking method, the operator only missed 13.4% of events. This was mainly due to two reasons: (1) event detection method failed at a few instances, and since the missed events were not selected to be displayed on a high resolution screen, they were also missed by the human operator; and (2) there were some inaccuracies in eyes tracking at a few instances. Overall, the results showed that the eye tracking method helped the operator in finding more important events.

6 Conclusions

The method proposed in this paper is based on a human-centric approach and it dynamically selects and schedules the four best camera views in a surveillance environment. The importance of a camera view is computed based on an integrated mechanism of automatic event detection and a human's monitoring and feedback. The non-invasiveness in obtaining the human operator's feedback is the key feature of the proposed method. The experiments have shown that augmenting the human operator's feedback with the automatic event detection method has helped in finding more of the important events.

The two backbones of the proposed human-centric approach are event detection and eye tracking methods. Therefore, their accuracy matters for the success of the proposed approach. Individually, both category of methods suffer from their limitations. However, as has been shown in the paper, when these methods augment each other, we can obtain better results and can perform more effective surveillance.

References

1. Amarnag S, Kumaran RS, Gowdy JN (2003) Real time eye tracking for human computer interfaces. In: IEEE international conference on multimedia and expo. Washington DC, USA, pp 557–560
2. Asteriadis S, Tzouveli P, Karpouzis K, Kollias S (2009) Estimation of behavioral user state based on eye gaze and head pose application in an e-learning environment. Multimed Tools Appl 41(3):469–493
3. Atrey PK (2009) A hierarchical model for representation of events in multimedia observation systems. In: The 1st ACM international workshop on events in multimedia. Beijing, China, pp 57–64
4. Atrey PK, Hossain MA, Saddik AE (2008) Automatic scheduling of cctv camera views using a human-centric approach. In: IEEE international conference on multimedia and expo. Hannover, Germany, pp 325–338
5. Atrey PK, Kankanhalli MS, Jain R (2006) Information assimilation framework for event detection in multimedia surveillance systems. Springer/ACM Multimed Syst J 12(3):239–253
6. Baumann MA, MacLean KE, Hazelton TW, McKay A (2010) Emulating human attention-getting practices with wearable haptics. In: IEEE haptics symposium. Waltham, USA, pp 149–156
7. Davis M (2003) Active capture: integrating human-computer interaction and computer vision/audition to automate media capture. In: IEEE international conference on multimedia and expo, vol 2, pp 185–188
8. Dee HM, Velastin SA (2007) How close are we to solving the problem of automated visual surveillance: a review of real-world surveillance, scientific progress and evaluative mechanisms. Mach Vis Appl 19(5–6):329–343
9. Hampapur A, Brown L, Connell J, Ekin A, Haas N, Lu M, Merkl H, Pankanti S, Senior A, Shu CF, Tian YL (2005) Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. IEEE Signal Process Mag 22(2):38–51
10. Hossain MA, Atrey PK, Saddik, AE (2011) Modeling and assessing quality of information in multi-sensor multimedia monitoring systems. ACM Trans Multimed Comput Commun Appl 7(1)
11. Itti L, Baldi P (2009) Bayesian surprise attracts human attention. Vision Res 49(10):1295–1306
12. Itti L, Koch C (2001) Computational modelling of visual attention. Nat Rev Neurosci 2:194–203
13. Leykin A, Hammoud R (2008) Real-time estimation of human attention field in LWIR and color surveillance videos. In: IEEE international workshop on object tracking and classification in and beyond the visible spectrum. Anchorage, USA, pp 1–6
14. Liu A, Zhang Y, Song Y, Zhang D, Li J, Yang Z (2008) Human attention model for semantic scene analysis in movies. In: IEEE international conference on multimedia and expo. Hannover, Germany, pp 1473–1476
15. Ma YF, Lu L, Zhang HJ, Li M (2002) A user attention model for video summarization. In: ACM international conference on multimedia, pp 533–542
16. Menezes P, Barreto JC, Dias J (2004) Face tracking based on haar-like features and eigenfaces. In: The 5th symposium on intelligent autonomous vehicles, pp 5–7
17. Peters C, O'Sullivan C (2003) Attention-driven eye gaze and blinking for virtual humans. In: ACM SIGGRAPH 2003 sketches & applications. San Diego, USA, pp 1–1
18. Radke RJ, Andra S, Al-Kofahi O, Roysam B (2005) Image change detection algorithms: a systematic survey. IEEE Trans Image Process 14(3):294–307
19. Reinders M (1997) Eye tracking by template matching using an automatic codebook generation scheme. In: Third annual conference of the advanced school for computing and imaging. Heijen, The Netherlands, pp 85–91
20. Rowe LA, Jain R (2005) ACM SIGMM retreat report on future directions in multimedia research. ACM Trans Multimed Comput Commun Appl 1(1):3–13
21. Savas Z (2005) Real-time detection and tracking of human eyes in video sequences. MSc thesis, Middle East Technical University, Ankara, Turkey
22. Savas Z (2008) Trackeye: real-time tracking of human eyes using a webcam. http://www.codeproject.com/KB/cpp/TrackEye.aspx

23. Smith P, Shah M, da Vitoria Lobo N (2000) Monitoring head/eye motion for driver alertness with one camera. In: IEEE international conference on pattern recognition. Barcelona, Spain, pp 636–642
24. Stauffer C, Grimson WEL (1999) Adaptive background mixture models for real-time tracking. In: IEEE Computer Society conference on computer vision and pattern recognition, vol 2. Ft. Collins, CO, USA, pp 252–258
25. Taylor JG, Fragopanagos N (2004) Modelling human attention and emotions. In: IEEE international joint conference on neural networks, vol 1. Budapest, Hungary, pp 501–506
26. Vaiapury K, Kankanhalli M (2008) Finding interesting images in albums using attention. J Multimedia 3(4):1–12
27. Vilaplana V, Marques F (2008) Region-based mean shift tracking: application to face tracking. In: The 15th IEEE international conference on image processing. San Diego, CA, pp 2712–2715
28. Vural U, Akgul YS (2009) Eye-gaze based real-time surveillance video synopsis. Pattern Recogn Lett 30:1151–1159
29. Wallace E, Diffey C (1988) CCTV control room ergonomics. Tech. rep., Police Scientific Development Branch, UK Home Office
30. Wang J, Kankanhalli MS, Yan W, Jain R (2003) Experiential sampling for video surveillance. In: First ACM international workshop on video surveillance. Berkeley, California, USA, pp 77–86
31. Wu C, Lin Y, Zhang WJ (2005) Human attention modeling in a human-machine interface based on the incorporation of contextual features in a Bayesian network. In: IEEE international conference on systems, man and cybernetics, vol 1. San Antonio, USA, pp 760–766
32. Wu J, Trivedi MM (2010) An eye localization, tracking and blink pattern recognition system: algorithm and evaluation. ACM Trans Multimed Comput Commun Appl 6(2):1–23

Pradeep K. Atrey is an Assistant Professor in the Department of Applied Computer Science at The University of Winnipeg, Canada. He received his B.Tech. (Computer Science and Engineering) and M.Sc. (Software Systems) from India; and Ph.D. in Computer Science from National University of Singapore. He was a postdoctoral researcher at the Multimedia Communications Research Laboratory, University of Ottawa, Canada. His current research interests are in the area of Multimedia Computing with a focus on Surveillance Security and Privacy, Smart Environment, and Web. He is an Editor for the ETRI Journal and was recipient of "ETRI Journal Best Reviewer (2009)" award. Dr. Atrey has been actively involved in the research community and has served as a member of the organizing and technical program committees of several international conferences.

Abdulmotaleb El Saddik (F'IEEE-09) is University Research Chair and Professor, SITE, University of Ottawa and recipient of the Professional of the Year Award (2008), the Friedrich Wilhelm Bessel Research Award from Germany's Alexander von Humboldt Foundation (2007), the Premier's Research Excellence Award (PREA 2004), and the National Capital Institute of Telecommunications (NCIT) New Professorship Incentive Award (2004). He is the Director of the Multimedia Communications Research Laboratory (MCRLab). He is a theme co-leader in the LORNET NSERC Research Network. He is Associate Editor of the ACM Transactions on Multimedia Computing, Communications and Applications (ACM TOMCCAP) and IEEE Transactions on Computational Intelligence and AI in Games (IEEE TCIAIG) and Guest Editor for several IEEE Transactions and Journals. Dr. El Saddik has been serving on several technical program committees of numerous IEEE and ACM events. He has been the General Chair and/or Technical Program Chair of more than 20 international conferences, symposia and workshops on collaborative haptoaudio-visual environments, multimedia communications and instrumentation and measurement. He was the General Co-Chair of ACM MM 2008. He is leading researcher in haptics, service-oriented architectures, collaborative environments and ambient interactive media and communications. He has authored and co-authored two books and more than 200 publications. He has received research grants and contracts totaling more than $10 million and has supervised more than 90 researchers. His research has been selected for the BEST Paper Award three times. Dr. El Saddik is a Senior Member of ACM, and is an IEEE Distinguished Lecturer.

Mohan S. Kankanhalli obtained his B.Tech. (Electrical Engineering) from the Indian Institute of Technology, Kharagpur and his M.Sc./Ph.D. (Computer and Systems Engineering) from the Rensselaer Polytechnic Institute. He is a Professor at the School of Computing at the National

University of Singapore. He is on the editorial boards of several journals including the ACM Transactions on Multimedia Computing, Communications, and Applications, IEEE Transactions on Multimedia, Springer Multimedia Systems Journal, Pattern Recognition Journal and the LNCS Transactions on Data Hiding and Multimedia Security. His current research interests are in Multimedia Systems (content processing, retrieval) and Multimedia Security (surveillance, authentication and digital rights management).