An Event-Driven Multiple Objects Surveillance System

Saeed Mina Qaisar
Effat University, Jeddah, KSA
College of Engineering, Department of Electrical and Computer Engineering
Jeddah, Saudi Arabia
sqaisar@effatuniversity.edu.sa

Dija Sidiya
Effat University, Jeddah, KSA
College of Engineering, Department of Electrical and Computer Engineering
Jeddah, Saudi Arabia

Mohammad Akbar
Effat University, Jeddah, KSA
College of Engineering, Department of Electrical and Computer Engineering
Jeddah, Saudi Arabia

Abdulhamit Subasi
Effat University, Jeddah, KSA
College of Engineering, Department of Information Systems
Jeddah, Saudi Arabia

Abstract – Traditional surveillance systems are constrained because of a fixed and preset pattern of monitoring. It can reduce the reliability of the system and cause an increased generation of false alarms. It results in an increased processing activity of the system, which causes an augmented consumption of system resources and power. Within this framework, a human surveillance system is proposed based on the event-driven awakening and self-organization principle. The proposed system overcomes these downsides up to a certain level. It is achieved by intelligently merging an assembly of sensors with two cameras, actuators, a lighting module and cost-effective embedded processors. With the exception of low-power event detectors, all other system modules remain in the sleep mode. These modules are activated only upon detection of an event and as a function of the sensing environment condition. It reduces power consumption and processing activity of the proposed system. An effective combination of a sensor assembly and a robust classifier suppresses generation of false alarms and improves system reliability. An experimental setup is realized in order to verify the functionality of the proposed system. Results confirm proper functionality of the implemented system. A 62.3-fold system memory utilization and bandwidth consumption reduction compared to traditional counterparts is achieved, i.e. a result of the proposed system self-organization and event-driven awakening features. It confirms that the proposed system outperforms its classical counterparts in terms of processing activity, power consumption and usage of resources.

Keywords – Event-driven awakening, embedded processing, face detection, resources utilization, surveillance

1. INTRODUCTION

Recent technological advancements have revolutionized the usage of surveillance. It is used in a variety of applications like detection of abnormal behavior, protection of property, public safety, crime prevention, parking lots, railway line crossing, monitoring of operations, sports, monitoring homes of elderly people, parks, traffic control, warehouses, green houses, prohibited areas, etc. [1]–[12]. Traditional surveillance systems are constrained because of a preset pattern of monitoring. It can result in the generation of false alarms [45], [46], and thus cause a reduction in the reliability of the system [13]–[15]. Furthermore, irrespective of the intended object presence, the system continually remains active [13]–[15]. Therefore, it causes a useless increase in the system processing activity and results in increased consumption of power and utilization of resources by the system [16].
Within this framework, a novel human surveillance system is proposed. It overcomes these downsides of traditional counterparts by intelligently exploiting and integrating event-driven awakening and self-organization features. It is capable of effectively monitoring multiple objects present within the surveillance system range. Works presented in [16]–[19] are sources of inspiration to integrate the event-driven awakening feature into the proposed system. The proposed solution also possesses an interesting feature of self-organization [20], [36]. Its pattern of monitoring and processing is adapted according to activities of humans within the system range. It makes the devised system perform better in comparison to traditional ones in terms of processing activity and reliability. Here, the term processing activity refers to a direct relationship with the utilization of resources and consumption of power. A less processing activity causes a reduced utilization of resources like storage memory, transmission bandwidth, power consumption, etc.

The system employs a smart combination of a sensor assembly and a robust classifier [32], [33]. It suppresses the generation of false alarms and improves system reliability. These interesting features are achieved by smartly integrating event-driven awakening and its self-organization features into the system.

Section 2 describes the devised system principle along with its different modules. The field of the view coverage principle is presented in Section 3. Section 4 discusses the functionality of the proposed system with the help of Algorithmic State Machine (ASM) charts. Section 5 describes system prototype implementation and post-implementation testing results. Finally, Section 6 concludes the article.

2. THE PROPOSED SYSTEM PRINCIPLE

The principle of the proposed system is shown in the block diagram given in Figure 1. It shows that the proposed system front-end is based on two similar modules. The modular design approach simplifies system implementation and debugging by controlling the design and implementation complexity. Different system blocks are described in the following subsections.

![The proposed surveillance system block diagram](image)

2.1. SENSOR ARRAYS

2.1.1. Event detectors

The proposed system is designed for human surveillance. Humans belong to the class of living objects [21] that emit infrared rays [22]. Event detectors always remain in the active mode. They snoop the presence of a living object by revealing infrared rays within the surveillance range [23]. Once an event is sensed, the information is transmitted to the front end embedded processor (FEEP) and the front end central processor (FECP).

2.1.2. Object location detectors

This module is activated by the FEEP according to the event detector output. After activation, it snoops living object locations in the system range [24]. Later on, camera orientations are managed by FEEPs as a function of the location detector output. According to the targeted application, the deployment of location detectors within the system range can be realized in different geometrical configurations. Depending on the output of this module, camera orientations are determined by FEEPs.

2.2. FRONT END EMBEDDED PROCESSOR (FEEP)

When an event occurs, firstly the corresponding FEEP and the FECP are activated. Later on, in coordination with the FECP, the FEEP manages system modes of operation, status observation and control of different system components like the camera, the actuator, sensor arrays and a lightning module.

2.3. CAMERA AND ACTUATOR

Each module contains one camera. It is mounted on an actuator and its orientation can be updated by the FEEP [25], [26]. The actuation commands are generated by the FEEP according to location detector output. It pilots the actuator in the desired pattern and orients the camera towards the targeted objects. The procedure is further discussed in Section 3.

2.4. LIGHTING MODULE

Appropriate luminance required for the smooth functioning of the system is provided by this module [27], [28]. The activation of the lighting module is controlled by the FEEP as a function of the event detector and luminosity sensors integrated in the lighting module outputs [29].

2.5. FRONT END CENTRAL PROCESSOR (FECP)

It manages both front-end modules and acts as a bridge between them and the Central Processing Unit (CPU). It receives information about system status via FEEPs through Inter-Integrated Circuit (I2C) ports [30], and the CPU through the Universal Serial Bus (USB) port [31]. Both FEEPs collect this information through their respective sensor arrays. The FECP employs this information to manage the system functional states (cf. Section 4).
2.6. CENTRAL PROCESSING UNIT (CPU)

This module is connected to the FECP and cameras via USB ports [31]. It coordinates with the FECP in order to update system states. Moreover, after activation, it receives images from cameras. It is composed of two data acquisition buffers, one classifier for face detection, one action decider and one action executor. When the event occurs, the CPU is triggered by the FECP. After activation, the CPU manages the acquisition of images from (an) active camera(s). Later on, these images are analyzed for the purpose of human face detection.

It acquires images from (an) active camera(s) and analyzes them for human face detection. The face detector block is based on the Viola Jones Algorithm [32]–[34]. It provides two outputs for each front-end module, respectively.

The action decider employs outputs of the face detector block. It remains standby along with the action executor, except if one or both face detectors outputs are high. The presence of one or more humans is confirmed by a high logic output of the face detector block. It activates the application executor block, which executes operations in parallel for both front-end modules. It can generate alarms and record images and video clips of the intended period [35]. This module is disengaged when outputs of both face detectors become low.

3. FIELD OF VIEW COVERAGE

In [28] and [37], the surveillance system angle object coverage problem is discussed. In the proposed approach, the overall coverage is $360^\circ$. It is split into different zones of localization. Each location sensor can conceal a specific portion of the complete field of view. In [28] and [37], authors employed multiple stationary cameras in order to cover the overall field of view. However, the suggested solution is composed of only two rotatable cameras. Location sensor output is employed to pilot the actuators. It is done in order to orient cameras towards the active zones. In this way, reliable and real-time surveillance of all active zones is ensured. The procedure is described with the help of Figure 2, where O presents the system origin and L is a horizontal axis.

Figure 2 shows that the system employs two cameras $C_1$ and $C_2$. They are placed on actuators $A_1$ and $A_2$ and are responsible for covering the horizontal angles between $[0^\circ; 180^\circ]$ and between $[180^\circ; 360^\circ]$, respectively. By default, $C_1$ and $C_2$ remain in their home positions and cover $Z_{p1}$ and $Z_{p2}$, respectively. Location sensor outputs are employed to update locations of $C_1$ and $C_2$ towards other active zones.

The system plane coverage view is defined by inequality (1).

$$0 < \frac{1}{2} r^2 \theta < \pi R^2,$$  

(1)

where $r$ is the radius from the origin to the target point and $R$ is the devised system overall coverage radius.

The system also contains a set of event detectors $SE = \{Ei, \text{where } \text{i} \in \{1, 2, 3, \ldots, M\}\}$. Their locations are prearranged.

An event detector location is given by $(r_{ei}, \theta_{ei})$. Here, $rei$ is the ith event detector distance from the origin. $\theta_{ei}$ represents the horizontal placement angle of the $i$th event detector. The ith event detector sensing zone, $Ze_{ei}$ is given by equation (2).

$$Ze_{ei} = \frac{1}{2} (r_{ei})^2 \alpha_{ei}.$$

(2)

In equation 2, $\alpha_{ei}$ is the central angle of the $i$th event detector with respect to its position. It is a function of physical properties and the fabrication process of the employed sensor. $r_{ei}$ is a radial position of the $i$th event detector from the origin $O$. It lies within the range of $0 < r_{ei} < R$, where $R$ is the maximum range of the plane coverage zone from the origin $O$.

The event detection field of view can be divided in a uniform or non-uniform way. It can be done by employing event detectors of different types. The choice should be made according to the sensitivity of different zones in the intended application [16]. A case of uniform distribution of the event detection field, among different event detectors $Ze_{e1}, Ze_{e2}, \ldots, Ze_{eM}$ is shown in Figure 2. In this case, a unique type of event detectors should be employed in the system.

A set of location sensors, $SP = \{Pj, \text{where } j \in \{1, 2, 3, \ldots, N\}\}$, is also integrated in each front-end module of the suggested system. Their locations are prearranged. The location of a position sensor is given by $(r_{pj}, \theta_{pj})$. Here, $r_{pj}$ is the $j$th positioning sensor distance from the origin. $\theta_{pj}$ represents the horizontal placement angle of the $j$th positioning sensor. The $j$th positioning sensor coverage zone, $Z_{pj}$ is defined by equation 3.

$$Z_{pj} = \frac{1}{2} (r_{pj})^2 \alpha_{pj}$$

(3)

In equation 3, $\alpha_{pj}$ is the central angle of the $j$th location sensor with respect to its position. It is a function of physical properties and the fabrication process of the employed positioning sensor. $r_{pj}$ is the radial position of the $j$th location sensor from the origin $O$. It lies within the range of $0 < r_{pj} < R$, where $R$ is the maximum range of the plane coverage zone from the origin $O$.

Event and location detectors can have different sensing radii. However, a case where $r_{pj} = r_{ei}$ is shown in Figure 2. Here, $R$ is the overall coverage radius of the system.

A set of multiple objects, $SH = \{Ha, \text{where } \text{i} \in \{1, 2, 3, \ldots, k\}\}$, can be covered by the proposed system. Locations of intended objects are not static, within the range of surveillance. They are sensed and resolve in a real-time fashion with the help of event detectors, location sensors and FEEP.
4. SYSTEM FUNCTIONALITY

The functionality of the proposed system is described by means of an ASM chart displayed in Figure 3. To improve power efficiency of the system, after initialization all system modules remain in the standby mode except for event detectors. Cameras mounted on actuators remain at their respective home positions.

The current system state becomes not equal to STANDBY on the recognition of an event. It triggers FEEPs and the FECP. Later on, FEEPs and the FECP manage the activation of other system blocks. In coordination with FEECPs and CPU, the FECP monitors the status of location sensors, face detector, cameras, actuators and the lighting blocks. Later on, the FECP updates functional states of the system according to these observations. Different system states are discussed next.

4.1. STANDBY STATE

In this state, except for low-power event detection sensors, all other system modules are kept in the standby mode. It assures a better utilization of system resources. The system functions and processes only relevant information and conserves its assets. Upon event detection by one of the front-end event detector array, the state of the system is updated to SNOOPING.

4.2. SNOOPING STATE

In this case, in coordination with FEEPs and the CPU, the FECP keeps observing the status of both front-end location sensors. This state is split into three different cases as a function of location sensor output.

Case 1: If no location sensors have found the intended object, then the state of the system is modified to STANDBY_MODE_ACTIVATION.

Case 2: If one location sensor has sensed the intended object, then the location of the target is first decided by the concerned FEEP(s). It is attained by using the target sensor predefined location table. Later on, the concerned camera(s) is oriented towards the respective active zone. Finally, the state of the system is updated to ACQUIRING.

Case 3: If more than one location sensors have sensed the intended objects among one or both front-end sensor arrays, then the state of the system is modified to VARIOUS_OBJECTS.

4.3. ACQUIRING STATE

In this state, images are captured and analyzed for face detection. When at least one face is recognized, alarms are generated and videos are recorded for the corresponding cameras for a predefined period. Later on, the state of the system is modified to SNOOPING. On the other hand, if no face is recognized on the acquired image, then the state of the system deprived of alarming and recording of a video is modified to SNOOPING.
4.6. STANDBY MODE ACTIVATION STATE

Cameras are oriented towards their respective home positions in this state. Later on, the state of the system is modified to STANDBY.

5. IMPLEMENTATION AND RESULTS

A system prototype is realized in order to exhibit the appealing features of the proposed system. Commercially available appropriate components are employed to achieve an expeditious prototype development. System costs, size and power consumption could be further optimized by employing a specific system design and development in place of the standard evaluation card based implementation.

The desired field of view of the developed system is between \([0 \degree; 360 \degree]\). The surveillance field is divided into six different positioning zones (cf. Figure 5). It is done in accordance with the employed cameras and positioning sensor characteristics, [16], [42]. It allows to properly cover the system overall field of view between \([0 \degree; 360 \degree]\). The first front-end module, \(M_1\), covers the surveillance field horizontal angle between \([0 \degree; 180 \degree]\). It is composed of zones \([Z_{p1}, Z_{p2}\) and \(Z_{p3}\)]. The second front-end module, \(M_2\), covers the remaining surveillance field horizontal angle between \([180 \degree; 360 \degree]\). It is composed of zones \([Z_{p4}, Z_{p5}\) and \(Z_{p6}\)]. The overall system coverage radius \(R=4m\) is selected. In this case, the system coverage view can be defined by inequality 4.

\[
0 < \frac{1}{2} r^2 \theta < 16 \pi
\]

where \(r\) is the radius from the origin to the targeted object and \(R=4m\) is the overall coverage radius of the developed system.

The FECP and FEEPs are implemented with Arduino UNO [38], [39]. A specific C-based development is realized for the front-end embedded blocks.

The lightning block is implemented with power efficient light-emitting diodes (LEDs) and light dependent resistors (LDRs) [16]. The CPU is implemented on a personal computer (PC). The classifier, application decoder and application executor blocks are implemented with MATLAB. These blocks run on the CPU. The liaison between the FECP and FEEPs is achieved via I2C ports. The CPU is connected to the FECP via the UART-USB bridge [47]. The images are transferred from cameras to the CPU through USB ports.

Figure 5 shows that the realized system is composed of two cameras \(C_1\) and \(C_2\), mounted on actuators \(A_1\) and \(A_2\), respectively. LifeCam HD-3000 with a central angle of 70° are employed as cameras [16], [40], and S3003 servo motors are employed as actuators. In order to orient camera lenses towards the active zones, actuators are piloted by FEEPs as a function of sensor array output. Rotation angles of actuators are adjusted in such a way that the \(C_i\) lens remains at 30°, 90° and 150° for \(Z_{p1}, Z_{p2}\) and \(Z_{p3}\) respectively. In the same way, the \(C_j\) lens remains at 210°, 270° and 330° for \(Z_{p4}, Z_{p5}\) and \(Z_{p6}\), respectively.

The system also employs two event detectors, \(E_1\) and \(E_2\). These are implemented with passive infrared (PIR) sensors and have a central angle of 180°; [16], [41]. \(E_1\) covers the system detection field between \([0 \degree; 180 \degree]\) and \(E_2\) covers the system detection field between \([180 \degree; 360 \degree]\). \(E_1\) and \(E_2\) locations are predetermined at \(r_{e1}=0.1cm, \theta_{e1}=90\degree\) and \(r_{e2}=0.1cm, \theta_{e2}=270\degree\), respectively. They have equal sensing radii of \(r_{e1}=r_{e2}=(4-0.1)m\), along with equal respective central angles of \(\alpha_{e1}=\alpha_{e2}=180\degree\) (cf. Figure 5). It results in two equal event detection sensing zones:

\[
Z_{e1} = Z_{e2} = \frac{1}{2}(3.9)^2 \pi.
\]

The specific coverage zone for each event detector is given by equations 6 and 7, respectively:

\[
Z_{e1} = \frac{1}{2}(3.9)^2 \theta, \text{ where } 0 \leq \theta \leq \pi.
\]

\[
Z_{e2} = \frac{1}{2}(3.9)^2 \theta, \text{ where } \pi \leq \theta \leq 2\pi.
\]

A set of six positioning sensors, \(SP = [P_1, P_2, P_3, P_4, P_5, P_6]\), is employed in the realized system. These are realized with six ultrasonic modules HC-SR04 and have a central angle of 60°, [16], [42]. Positioning sensor locations are predetermined at \(r_{p1}=0.2cm, \theta_{p1}=30\degree\), \(r_{p2}=0.2cm, \theta_{p2}=90\degree\), \(r_{p3}=0.2cm, \theta_{p3}=150\degree\), \(r_{p4}=0.2cm, \theta_{p4}=210\degree\), \(r_{p5}=0.2cm, \theta_{p5}=270\degree\) and \(r_{p6}=0.2cm, \theta_{p6}=330\degree\), respectively. They have equal sensing radii of \(r_{p1}=r_{p2}=r_{p3}=r_{p4}=r_{p5}=r_{p6}=(4-0.2)m\) along with equal respective central angles of \(\alpha_{p1}=\alpha_{p2}=\alpha_{p3}=\alpha_{p4}=\alpha_{p5}=\alpha_{p6}=60\degree\). This results in uniform target positioning zones, expressed mathematically by equation 8.

\[
Z_{pj} = \frac{1}{2}(3.8)^2 \frac{\pi}{3}, \text{ where } j(e(1,2,\ldots,6)).
\]

By default, cameras remain at their respective home positions and cover \(Z_{p1}\) and \(Z_{p3}\). Their orientations are updated to other active zones according to positioning sensor output. This prototype can cover multiple humans present within the system range. In the case of the presence of multiple humans within the field of view of \(C_j\), the \(A_j\) rotation is prioritized as \(Z_{p6}\) then \(Z_{p5}\) and lastly \(Z_{p4}\). Similarly, for the field of view of \(C_i\), the \(A_i\) rotation is prioritized as \(Z_{p2}\) then \(Z_{p3}\) and lastly \(Z_{p1}\).

The developed system is positioned at the center of a 5x5m² laboratory. The setup is shown in Figure 6. System functionality is tested for a simultaneous surveillance of two humans who enter simultaneously the coverage fields of \(C_1\) and \(C_2\). When an event is detected, human presence is firstly confirmed via the face detector. Later on, depending on the face detector output, alarms are generated and video clips are recorded.
By analyzing and correlating these videos and alarm stamps, it is noted that actions, alarming and video recording, are executed only when an event of interest occurs. False alarms, generated by event detectors, are suppressed by the face detection classifier. Moreover, cameras successfully follow the intended objects as a function of their occurrence in different locations. The lighting block is triggered only when the luminosity of a sensing environment is lower than a predefined threshold. It remains disabled otherwise. It adds to system power consumption efficiency when compared to its counterparts [27], [28], [37].

System performance is studied in terms of memory utilization. It is accomplished with a three-hour experiment. Throughout this period two persons simultaneously entered the surveillance fields of both front-end modules thirteen times. They did it after random time intervals and at different positioning zones. Each time, they entered the range of the system for a duration of less than two minutes. When an event and the face are detected, a two-second video clip is recorded for the concerned camera. As a result, twenty-six minute videos are recorded, for both $C_1$ and $C_2$, for the three-hour experiment. The process is further clear from results, as shown in Figure 7.

In equation 9, the Frame Rate represents the number of frames per second. The frame size can be computed by equation 10.

$$\text{Frame size (Bytes)} = \frac{\text{Camera Resolution} \times \text{Colour Depth}}{8}$$  \hspace{1cm} (10)

On the right-hand side of equation 10, a constant denominator is employed to calculate the Frame size in bytes. By replacing the employed camera resolution and color depth from [49], the Frame size of 900KBytes is obtained.

Calculations based on equations 9 and 10 result in a memory utilization of 528MBytes/minute. Therefore, for the proposed system the total video size obtained for each camera is 13.4GB. Therefore, the overall video size obtained for both cameras is 26.8GB.

These results are compared with the counter classical solution. In this case, cameras remain static. Therefore, the surveillance pattern and the field of view of each camera remain fixed. To study the traditional surveillance approach, a system is developed based on the same LifeCam HD-3000 cameras. Each of these cameras can cover a planner angle of 70 degrees. Therefore, 6 cameras are employed for a proper coverage of 360 degrees of the planner surveillance angle. These cameras are connected to the CPU via two USB hubs [48]. All system modules function continuously irrespective of the existence of intended objects in the system range. Moreover, the recording of videos is performed at a rate of 30frames per second. It results in a memory utilization rate of 1584MBytes/minute. The process is clear from Figure 7. The recorded video duration remains one 180 minutes for each camera and at a rate of 15frames per second it delivers into a video size of
278.4GB for each employed camera. Therefore, the overall memory utilization for all cameras is 1670.4GB.

It shows a 20.8 times system memory utilization reduction per camera and an overall 62.3 times system memory utilization reduction achieved by the proposed system when compared to its counterparts [37]. It promises a gain of more than one order degree in terms of power consumption and data processing activity for the proposed system when compared to the traditional one. It is accomplished by efficiently merging the appealing features of event-driven awakening, soft decimation and self-organization techniques.

In order to improve system memory utilization the acquired videos are compacted with the MPEG4 video compressor [50]. This results in a memory utilization rate of 1.4Mbits/second for the proposed solution and of 12.2Mbits/second for the classical one. An appropriate network bandwidth is required for a proper transmission of the surveillance data towards a remote observer. Within this framework, a study is performed in terms of network bandwidth requirements. The key factors which should be taken into account while doing these calculations are the resolution of video recording, the frame rate per second and the employed video compression technique. For real-time transmission a data rate of 1.4Mbits/second is required for the proposed solution. The bandwidth requirement for real-time transmission in the classical case is 12.2Mbits/second. It shows a 9 times bandwidth consumption reduction per second achieved by the proposed solution compared to the traditional one. It is attained by reducing the frame rate and the number of required cameras in the devised system compared to the classical one. Furthermore, in the classical case, the system will continuously consume the network bandwidth at a fixed rate. It results in a bandwidth consumption of 16.1GBytes for the classical case for the overall experiment duration. On the other hand, because of its event-driven and self-organization features, the proposed system does not consume the network bandwidth in a continuous fashion. Bandwidth utilization will be correlated with the intended activity because the proposed system only records and transmits relevant information. The process is clear from Figure 7. Therefore, for overall experiment duration of 3 hours, the proposed system will consume a total bandwidth of 264MBytes. It shows that the devised system achieves an overall bandwidth consumption reduction of 62.3 times with respect to its counterparts.

6. CONCLUSION

An inventive event-driven human surveillance system is proposed in this article. The traditional surveillance systems are constrained because of a prefixed pattern of monitoring. Consequently, false alarms can be triggered and this can result in a reduction of system reliability. Furthermore, all system modules function continuously irrespective or the existence of target in the range of the system. This causes an augmented processing activity of the system and leads towards an unserviceable utilization of resources and an unwanted consumption of power. With experimental results, it is demonstrated that the proposed system overcomes these downsides up to a certain extent by smartly integrating the event-driven awakening and self-organization features in the proposed solution.

Results have shown that the developed system is capable of monitoring multiple objects. Furthermore, for the case under study, the total 62.3 times reduction of the system memory usage and of the bandwidth consumption is attained by the devised solution in comparison to its counterparts [27], [28], [37]. It tends to achieve a gain of more than one order degree in terms of power consumption of the proposed system when compared to some other ones. It is attained by intelligently integrating event-driven awakening, the soft decimation and the self-organization techniques in the proposed system.

The proposed system power consumption measurement and its comparison with the classical one is in progress. Addition of wireless and mobile vision nodes is also under study. The devised system prototyping and testing with an extended number of front-end modules will be done in the future. A study on system performance in the crowded and real environment is also a future task. The devised system possesses a capability to suppress false alarms. It is attained by intelligently using a robust classifier with front-end sensor arrays. System performance evaluation in terms of false alarm rejection is another future task. An automated alteration of the video recording frame rate as a function of the system environment status is a prospect. It could be realized by integrating machine learning tools in the system. Another prospect is to realize a completely embedded and cost-effective implementation of the devised system.

7. ACKNOWLEDGEMENT

Authors are thankful to anonymous reviewers for their valuable feedback. Authors are also thankful to Eng. S. Ammari for her help during system prototyping and experimentation. The project is funded by the Effat University, Jeddah, KSA.

8. REFERENCES:


