A Surveillance System Based on Motion Detection and Motion Estimation using Optical Flow

Muhammad Kamal Hossen
Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology,
Chittagong-4349, Chittagong, Bangladesh.
mkhossen@cuet.ac.bd

Sabrina Hoque Tuli
Department of Computer Science and Engineering,
Chittagong University of Engineering and Technology,
Chittagong-4349, Chittagong, Bangladesh.
sabrina.tuli07@gmail.com

Abstract— **In today's world Surveillance system is playing an important role in the field of security. Moving object detection has been widely used in video surveillance system. As well as motion estimation is an important part of surveillance video processing such as video filtering and compression from video frames. This paper proposes a simple and efficient surveillance system based on motion detection with motion vector estimation from surveillance video frames. Motion is detected with a new approach- edge region determination which makes detection faster. The surveillance video is then processed for motion estimation using optical flow with Horn-Schunck algorithm for estimating motion vector for its reasonable performance and simplicity. This method is computationally faster without requiring any special hardware for image processing. So it can be more applicable to embedded systems.**

Keywords—motion detection; video surveillance; motion estimation; optical flow

I. INTRODUCTION

In recent years surveillance system have become more popular in the field of computer vision. Traditional surveillance systems only provide analog services in hardware. Security guards must stay at security room and look at arrays of CCTV (Closed Circuit Television) or play back the videotapes sequentially to find out the surveillance events. This demanding task is very inefficient. Real time moving object detection is core of surveillance applications. One of the main challenges in these applications is to detect moving objects competently. Moving object detection judges the change in images, captured by a camera and detects whether any moving object present or not, if there any, extracts the object as soon as possible. Although a number of well known methods are exist, however problem becomes more difficult to solve in presence of noise, illumination changes, complex body motion and in real time environment. Background modeling, often required in background subtraction, is very time consuming and complex. Some texture based boundary evaluation methods also exist but their computational costs are relatively high hence they are incompatible for real time purposes.

In this paper, we proposed a surveillance system where motion detection scheme is an edge based approach. Edges are robust against illumination changes and noise. The rest of this paper is categorized as follows. Section 2 gives a brief description of related research. Section 3 describes the whole surveillance system and motion estimation from surveillance video frames. Section 4 shows the experimental results, analysis and computational cost of our proposed system. Section 5 concludes this paper. The last section 6 includes references.

II. RELATED RESEARCH

The demand of surveillance application is gradually increasing day by day. Everyone wants to be as much as secure as possible. Ensuring security for a specific place like home, offices in both personal and corporate life is a burning issue in recent time. In corporate life, the need is more. Moving object detection is the core and fundamental task of every surveillance system. Many methods have been proposed for detecting moving objects in video sequences. Background subtraction is a well known approach to extract moving object. The methods [1]-[3] proposed a number of background subtraction techniques. Each pixel in a new frame is compared against a background model and is considered as moving object if it differs significantly from that model. Unfortunately, the derivation of model is complex, time consuming and computationally expensive. Kim and Hwang method [4], describes an edge based method, which is sensitive to intensity image and requires higher computational cost. Optical flow methods make use of the flow vectors of moving objects over time to detect moving regions in an image. Optical flow is more accurate and gives more perfect result in different illumination even in moving camera condition. Horn-Schunck algorithm [5] is a prominent algorithm for determining optical flow for its simplicity and reasonable performance. Here we have used this algorithm for determining flow vectors from surveillance video frames.

III. ARCHITECTURE

A. Overview of the Surveillance System

A total glance of the proposed system is given in the figure 1. Brief description of each of the cited phases in the figure is discussed in the following sections.

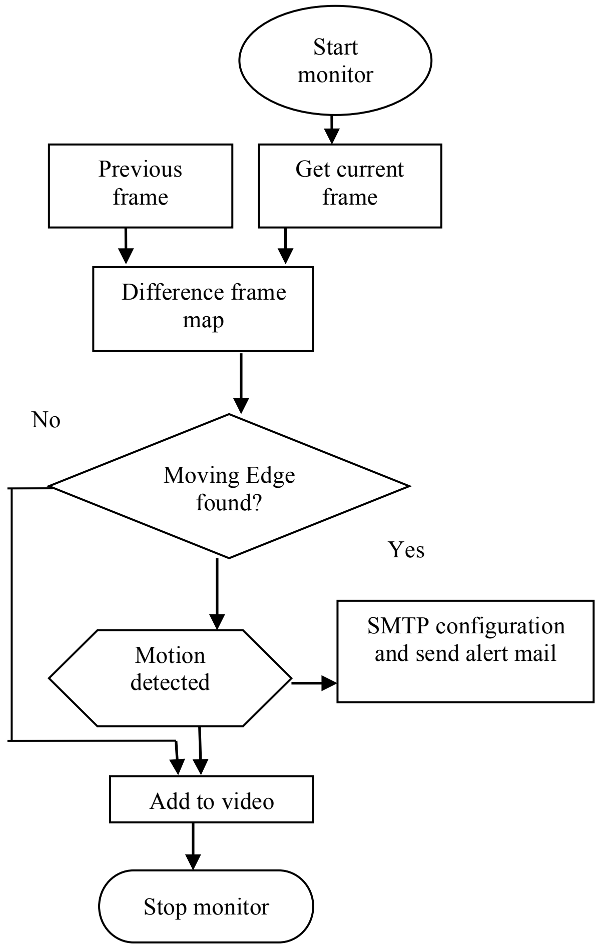


Figure 1. Surveillance System Architecture

Motion detection is the process of detecting a change in position of an object relative to its surroundings or the change in the surroundings relative to an object. When user commands the system to start monitor using GUI, then image acquisition device (here webcam) starts to acquire snapshot of the current scenario frequently one after another. Then it compares the current image with previously taken image and if any change is found between these successive two images, then it immediately shows warning dialogue that motions detected and send an alert message to the predefined e-mail address. For the purpose of motion detection between consecutive frames, first of all we take the grayscale image of current snapshot as I_n and previous image as I_{n-1}. Then absolute difference of these two images $|I_n - I_{n-1}|$ with a threshold value is determined for further calculations. The difference edge map DE_n is derived using Sobel edge detector to detect if there is a motion occurred otherwise no difference edge map is found. All motion detected frame and non-motion detected frame are recorded to a video file for further operation.

Edge model is defined $\boldsymbol{DEn= \{e1, e2, e3, \ldots\ldots, ek\}}$ as a set of all edge points detected by the sobel operator from the difference of current frame $\boldsymbol{n}$ and previous frame $\boldsymbol{n-1}$. Another edge model $\boldsymbol{DEb= \{a1, a2, a3, \ldots\ldots, ak\}}$ as a set of all edge points detected by the sobel operator from the difference of current frame $\boldsymbol{n}$ and background frame $\boldsymbol{b}$. Moving edges can be determined by finding those pixel positions where both difference edge map has edge points, i.e. where $\boldsymbol{e == a}$. Equation (1) represents the moving edge map of current frame $\boldsymbol{n}$ as follows:

$$\boldsymbol{MEn=\{ x\varepsilon MEn : x\varepsilon\ (DEn \cap DEb)\}} \qquad (1)$$

Now the region where moving edges exist in the current frame $\boldsymbol{n}$ is needed to be determined. Again, $\boldsymbol{Row_n= \{x1, x2, x3, \ldots., xk\}}$ as a set of all X-axis values of all moving edge pixels and $\boldsymbol{Col_n= \{y1, y2, y3, \ldots., yk\}}$ as a set of all Y-axis values of all moving edge pixels of the moving edge map $\boldsymbol{ME_n}$. Then the minimum (x,y) and maximum (x,y) values from all edge points is founded, i.e.

$$\boldsymbol{Row_{min}= \{x\varepsilon Row_n \mid min||x||\}} \qquad (2)$$

$$\boldsymbol{Row_{max}= \{x\varepsilon Row_n \mid max||x||\}} \qquad (3)$$

$$\boldsymbol{Col_{min}= \{y\varepsilon Col_n \mid min||y||\}} \qquad (4)$$

$$\boldsymbol{Col_{max}= \{y\varepsilon Col_n \mid max||y||\}} \qquad (5)$$

From the value of $\boldsymbol{Row_{min}}$, $\boldsymbol{Col_{min}}$, and $\boldsymbol{Row_{max}}$, $\boldsymbol{Col_{max}}$ defined in equation (2), (3), (4), (5) the moving edge region $\boldsymbol{MEReg_n}$ of current frame $\boldsymbol{n}$ can be determined. Finally the moving object region of current frame $\boldsymbol{n}$ is extracted which is defined in the equation (6).

$$\boldsymbol{MObjReg_n = I_n(x,y)} \begin{cases} \boldsymbol{where, Row_{min} \le x \le Row_{max}} \\ \boldsymbol{where, Col_{min} \le y \le Col_{max}} \end{cases} \qquad (6)$$

where $\boldsymbol{I_n}$ is the current frame, $\boldsymbol{MObjReg_n}$ is the moving object region of current frame, in which region all moving objects are found. It requires much less calculation and computation time than other morphological operations for extraction of single or multiple View Object Plane (VOP). It is much simpler, faster and more accurate to extract single or multiple moving objects within their range in the image frame.

B. SMTP (Simple Mail Transfer Protocol) Configuration

Simple Mail Transfer Protocol (SMTP) is a protocol for sending electronic mail (email) messages between servers. SMTP is generally used to send email messages from a mail client to a mail server. The SMTP design architecture is shown as follows in figure. 2:

Figure 2: SMTP design structure

When an SMTP client has a message to transmit, it establishes a two-way transmission channel to an SMTP server. The responsibility of an SMTP client is to transfer mail messages to one or more SMTP servers, or report its failure to do so. Here we have used a Gmail account for sending mail. The Gmail SMTP server settings for sending mail is given below:

- Gmail SMTP server address: smtp.gmail.com
- Gmail SMTP user name: full Gmail address
- Gmail SMTP password: Gmail password
- Gmail SMTP port: 465
- Gmail SMTP SSL(Secure Sockets Layer) required: yes
- Gmail SMTP authentication required: yes

C. Motion Estimation from Surveillance Video Frames

Optical flow is a most common and efficient technique for motion estimation. Here we have used optical flow for measuring relative motion between frames of surveillance videos. Optical flow is the apparent motion of image brightness pattern of objects in a visual scene caused by the relative motion between an observer and the scene. The optical flow methods try to calculate the motion between two image frames which are taken at times t and $t + \Delta t$ at every pixel position. We have used *Horn-Schunck* algorithm for measuring optical flow. The *Horn-Schunk* algorithm (HS) is one of the classical algorithms in optical flow due to its reasonable performance and simplicity of the algorithm. The HS algorithm is a technique used to identify the image velocity or motion vector based on Spatial Temporal Gradient Technique which computes the image velocity from spatiotemporal derivatives of image intensity. Figure. 3 shows each step of motion estimation process with Horn-Schunck algorithm.

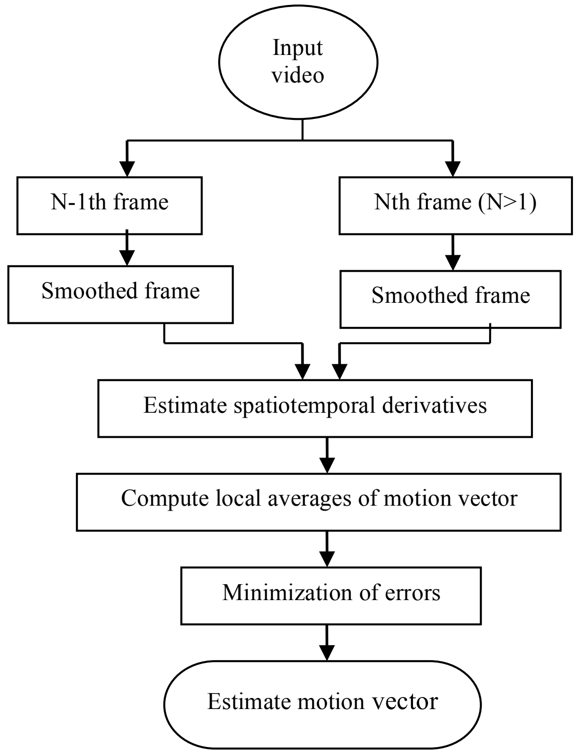


Figure 3: Block diagram of motion estimation

1) Gaussian Smoothing: Every frame is smoothed for further processing. In smoothing, the pixels of an image are modified so individual points (presumably because of noise) are reduced, and pixels that are lower than the adjacent pixels are increased leading to a smoother image. Here we used Gaussian smoothing. A Gaussian smoothing is the result of blurring an image by a Gaussian function. Gaussian function in two dimensions,

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} exp^{-\frac{x^2+y^2}{2\sigma^2}}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

Here we have used 3x3 Gaussian filter for smoothing :

1/16	1	2	1
	2	4	2
	1	2	1

2) Horn-Schunck Algorithm : A sequence of 2D images is mathematically described as a function $I(x, y, t)$, where I is the image intensity at time t and at position (x, y). The total derivative of change of brightness is given by

$$\frac{DI}{Dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} \tag{7}$$

where, $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ can be computed directly from a pair of images *I(x,y,t)* and *I(x,y,t+δt)*; $\frac{dx}{dt}$ and $\frac{dy}{dt}$ are the *u* and *v* components of velocity, respectively. Suppose that, when the pattern moves, the intensity *I(x, y, t)* of a particular point in the pattern is conserved, so that $\frac{DI}{Dt} = 0$ in expression (7). Considering this constraint, $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$ and $I_t = \frac{\partial I}{\partial t}$, equation (1) can be rewritten as:

$$\frac{DI}{Dt} = I_x u + I_y v + I_t = 0 \qquad (8)$$

We must observe that this is a single equation with two unknowns variables u and v. Besides, no smoothness constraints and prior knowledge about the solution have been considered yet. All these problems can be addressed by introducing a regularization term and a variational formulation for the problem. Therefore, Horn and Schunck compute *u* and *v* for each pixel of the image by minimizing the functional

$$E^2 = \int_\Omega \left[\alpha^2 E_c^2 + E_b^2\right] d\Omega \qquad (9)$$

where α is a regularization parameter; E_c^2 and E_b^2 are, respectively, the data and regularization terms, given by:

$$E_b^2 = (I_x u + I_y v + I_t)^2 \qquad (10)$$
$$E_c^2 = (u_x)^2 + (u_y)^2 + (v_x)^2 + (v_y)^2 = \|\nabla v\|_2^2 \qquad (11)$$

where *v = (u, v)*. By minimizing the functional (9), we obtain the Euler equations:

$$I_x^2 u + I_x I_y v = \alpha^2 \nabla^2 u - I_x I_t$$
$$I_y^2 v + I_x I_y u = \alpha^2 \nabla^2 v - I_y I_t \qquad (12)$$

The Laplacian ∇u and ∇v in equation (12) are approximated by :

$$\nabla^2 u = (\bar{u} - u)$$
$$\nabla^2 v = (\bar{v} - v) \qquad (13)$$

where, $\bar{u}$ and $\bar{v}$ are local averages values between the pixels:

$$\overline{u_{i,j,k}} = 1/6 \{u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k}\} + 1/12 \{u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k}\},$$

$$\overline{v_{i,j,k}} = 1/6 \{v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k}\} + 1/12 \{v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j+1,k} + v_{i+1,j-1,k}\} \qquad (14)$$

By replacing expression (12) in equation (13), we can obtain the system

$$(\alpha^2 + I_x^2)u + I_x I_y v = \alpha^2 \bar{u} - I_x I_t ,$$
$$(\alpha^2 + I_y^2)v + I_x I_y u = \alpha^2 \bar{v} - I_y I_t \qquad (15)$$

whose solution is given by,

$$u = \frac{(\alpha^2 + I_y^2)\bar{u} - I_x I_y \bar{v} - I_x I_t}{\alpha^2 + {I_x}^2 + {I_y}^2}$$
$$v = \frac{(\alpha^2 + I_x^2)\bar{v} - I_x I_y \bar{u} - I_y I_t}{\alpha^2 + {I_x}^2 + {I_y}^2} \qquad (16)$$

The system (15) can be iteratively solved through the scheme:

$$u^{k+1} = \overline{u^k} - \frac{I_x\left[I_x \overline{u^K} + I_y \overline{v^k} + I_t\right]}{\alpha^2 + {I_x}^2 + {I_y}^2}$$
$$v^{k+1} = \overline{v^k} - \frac{I_x\left[I_x \overline{u^K} + I_y \overline{v^k} + I_t\right]}{\alpha^2 + {I_x}^2 + {I_y}^2} \qquad (17)$$

where k indicates the current iteration of the algorithm, and the partial derivatives I_x, I_y and I_t are computed by a forward finite difference method :

$$I_x = 0.25(I_{i,j+1,k} - I_{i,j,k} + I_{i+1,j+1,k} - I_{i+1,j,k} + I_{i,j+1,k+1} - I_{i,j,k+1} + I_{i+1,j+1,k+1} - I_{i+1,j,k+1}),$$

$$I_y = 0.25(I_{i+1,j,k} - I_{i,j,k} + I_{i+1,j+1,k} - I_{i,j+1,k} + I_{i+1,j,k+1} - I_{i,j,k+1} + I_{i+1,j+1,k+1} - I_{i,j+1,k+1}),$$

$$I_t = 0.25(I_{i,j,k+1} - I_{i,j,k} + I_{i+1,j,k+1} - I_{i+1,j,k} + I_{i,j+1,k+1} - I_{i,j+1,k} + I_{i+1,j+1,k+1} - I_{i+1,j+1,k}). \qquad (18)$$

IV. RESULTS & ANALYSIS

This section gives a brief description of the experiments performed on surveillance videos recorded by a low end camera, comparative analysis of the computational cost and different parameters of motion estimation.

A. Experimental Results

For implementing our surveillance system and moving object extraction from surveillance video sequences, we used a laptop running Windows 7 Ultimate (32-bit operating system). The system has the Intel Core i3 processor 2.13 GHz, 2 GB RAM. No other application was running in the background while experiments were performed. When user push Start Monitor button to the right of the preview, monitoring gets started and every snapshot is captured and recorded as a video. While monitoring, if any motion is found immediately it is detected and a warning dialogue box is displayed on the screen that shows in how many frames motion is detected.

(a) (b)

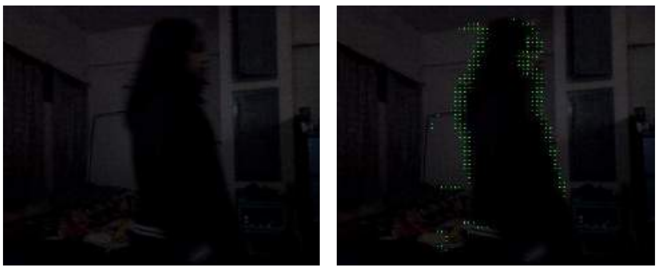
(c)

Figure 4: Motion detection of record.avi (frame-37): (a) no motion; (b) motion detected; (b) warning dialog

Estimation of motion vector among image sequences of surveillance videos is performed applying Horn-Schunck optical method and motion vectors on video frames is drawn to view relative motion between immediate frames. Here is some output of surveillance videos with different illumination condition including input video frames and output frames with motion vector.

(a) (b)

Figure 5: Motion estimation of record.avi (frame-46) (semi-indoor): (a) original; (b) motion vector



(a) (b)

Figure 6: Motion estimation of record2.avi (frame-40) (indoor-night): (a) original; (b) motion vector

(a) (b)

Figure 7: Motion estimation of record1.avi (frame-34) (indoor-day): (a) original; (b) motion vector

(a) (b)

Figure 8: Motion estimation of record3.avi (frame-102) (outdoor-sunny): (a) original; (b) motion vector

(a) (b)

Figure 9: Motion estimation of record7.avi (frame-16) (outdoor-foggy): (a) original; (b) motion vector

B. Performance Analysis

We have calculated computation cost for each step of motion detection while monitoring and motion estimation using optical flow for the purpose of performance evaluation and we also calculated mean velocity error and mean angular error for motion estimation with optical flow. We have evaluated the performance of motion estimation using Horn-Schunck algorithm. The first one, namely Err_{vel}, is the mean velocity error is computed by:

$$Err_{vel} = \frac{\sum_{i=1}^{nv} \left| \|v_{est\_i}\| - \|v_{gt\_i}\| \right|}{nv} \qquad (19)$$

Where, v_{est} is the estimated motion vector and v_{gt} is the ground truth vector of each frame and nv is the total number of frames of a video.

The second error metrics is the mean-angular error is defined by:

$$Err_{ang} = \frac{\sum_{i=1}^{nv} arccos\left(\frac{v_{est\_i} . v_{gt\_i}}{\|v_{est\_i}\| \|v_{gt\_i}\|}\right)}{nv} \qquad (20)$$

For, test cases we have used tolerance level for iteration is 0.0001 and global smoothness weight α is 1.

We have calculated mean velocity error and mean angular error for surveillance videos in various illumination conditions like indoor, outdoor, semi-indoor, sunny, foggy etc. Table 1 shows the mean velocity error and mean angular error for different surveillance videos with different illumination condition:

TABLE 1: MEAN VELOCITY ERROR AND MEAN ANGULAR ERROR FOR SURVEILLANCE VIDEOS

Video Title	Global Smoothness (α)	Iteration Number	Mean Velocity Error Err_{vel}	Mean Angular Error Err_{ang}
record.avi (semi-indoor sunlight)	1.0	100	0.0961	1.3788
record1.avi (indoor-day)	1.0	100	0.0983	1.5603
record2.avi (indoor-night)	1.0	100	0.0994	1.5184
record3.avi (outdoor-sunny)	1.0	100	0.0822	1.4311
record7.avi (outdoor-foggy)	1.0	100	0.1195	1.4486

Therefore, it shows the mean velocity error Err_{vel} and mean angular error for different illumination condition surveillance videos and finally we got the average Err_{vel} is 0.4363 and Err_{ang} is 1.4674. Here, we have performed a comparative analysis with Kim and Hwang method based on computational cost.

TABLE 2. MEAN PROCESSING TIME OF THE PROPOSED SYSTEM COMPARED WITH KIM AND HWANG METHOD

Processing Steps(Proposed System)	Mean Time (ms)	Processing Steps(Kim and Hwang)	Mean time (ms)
Difference map image calculation	4	Difference map images calculation	4
Edge map generation from difference images	18	Edge maps calculation	72
Estimate spatiotemporal derivatives	19	Moving edge map calculation	5
Minimization of errors	34	Morphological operations	10
Motion vector draw	9	Moving object extraction	7
Total time required	82	Total time required	98

Table 2 shows that the total time required to process an image of 352x288 size is about 82 ms. Therefore our system can process about 12 frames per second to detect and estimate motion from surveillance video frames. By increasing the CPU speed and changing platform this time can be improved.

V. CONCLUSION

This paper presents a surveillance system based on new motion detection approach and motion estimation from surveillance video sequences. The motion detection scheme in surveillance system is computationally faster and motion detection accuracy is about 98.6%. Motion estimation approach is done by Horn-Schunck optical flow algorithm. This provides high density of motion vector and simplicity in implementation. So, it needs less computational cost. Besides, no morphological operations are performed which makes our system swift. It is free from illumination changes like indoor, outdoor, sunny, foggy, night, day etc. and moving camera condition. In our future work, we will work for 3D motion estimation with tracking, recognition and classification of moving object and solving limitations in future study.

VI. REFERENCES

[1] R. Zhang, J. Ding, “Object Tracking and Detecting Based on Adaptive Background Subtraction”, International Workshop on Information and Electronics Engineering (IWIEE), 2012.

[2] D. K. Yadav, “Efficient method for moving object detection in cluttered background using Gaussian Mixture Model”, IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2014.

[3] D. K. Yadav, L. Sharma, S. K. Bharti, “Moving object detection in real-time visual surveillance using background subtraction technique”, IEEE International Conference on Hybrid Intelligent System (HIS), 2014.

[4] C. Kim, J. N. Hwang, “Fast and automatic video object segmentations and tracking for content-based applications”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, No. 2, February, 2002.

[5] B. K. P. Horn and B. G. Schunck, “Determining Optical Flow”, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A, 17(1981), pp. 185-203.

[6] D. Fortune, P. Bouthemy, C. Kervrann “Optical flow modeling and computation: A survey”, ELSEVIER Computer Vision and Image Understanding, Volume 134, pp 1-21, May 2015.

[7] S. Kermani, K. Jamshidi, A. Monadjemi, “Independent Motion Detection using Optical Flow on Stereo Images”, 5th Symposium of Advances in Science & Technology, 2011.

[8] A. A. Shafie, F. Hafiz, and M. H. Ali, “Motion Detection Techniques Using Optical Flow”, World Academy of Science Engineering and Technology, 2010.