Dynamic Probe Window Based Optimization for Surveillance in Home Security System

Bhaskar Kapoor[1] and Anamika Chhabra[2]

[1]Department of Information Technology, MAIT, New Delhi INDIA

bhaskarkapoor@gmail.com

[2]Department of Information Technology, MAIT, New Delhi INDIA

anamika.chhabra@gmail.com

ABSTRACT

Tracking of a moving object is very important for video surveillance in a real time scenario. The proposed algorithm uses dynamic probe window based approach & combines the conventional edge based and frame differencing approach to achieve better algorithmic time complexity as well as improved results. First it computes the edge map of two consecutive frames with the help of first order differential sobel operator due to its noise resistant attributes and applies the frame differencing method between the two consecutive edge maps. Apart from the above optimization, our method doesn't differentiate between the scenario when motion occurs and when it doesn't, that is, almost same computation overhead is required even if motion is not there so it reduces the time complexity of the algorithm when no motion is detected. The effectiveness of the proposed motion detection algorithm is demonstrated in a real time environment and the evaluation results are reported.

KEYWORDS

Motion Detection in real time scenario, Edge Detection, Frame Differencing, Dynamic Probe Window

1. INTRODUCTION

Real-time motion detection has attracted a great interest from many computer vision researchers due to its wide application scenarios, such as home security surveillance systems, surveillance in mining & hazard zones [1], traffic monitoring & many more application areas. However, the main point of concern had been the large computation complexity or time involved in processing the motion detection algorithms and obtaining accurate results. Currently, the main motion detection algorithms include: 1) Frame Difference Method / Temporal Differencing 2) Background Subtraction Method 3) Optical Flow Method 4) Statistical Learning Method

Optical flow method [4] [11] is the most complex algorithm. It spends more time than other methods, and statistical learning method needs many training samples and also has much computational complexity. These two methods are not suitable for real-time processing. The Background [4][11] subtraction method is extremely sensitive to the changes of light. Frame difference method [4][5][10]is simple and easy to implement, but the results are not accurate enough, because the changes taking place in the background brightness cause misjudgment. All the above methods are computationally expensive and therefore make's some tradeoff between the speed & the accuracy of detection.

In this paper, detection method & frame differencing method [5][6] is presented along with a dynamic probe window optimization. One of the perceptual user interface that we tend to exploited in motion detection and surveillance system [4][5] is the human body movement. Perceptual user interfaces are ones in which the computer is given the ability to sense and produce analogs of the human senses ,such as allowing computers to perceive and produce localized sound and speech ,giving computers a sense of touch and force feedback and in our case giving the computers an ability to detect motion in continuous moving frames.

In our paper, firstly, Edge detection methods are used to generate the edge maps of two consecutive frames in a video sequence. As the edges of moving objects are almost not changed with light, this method is not sensitive to the changes of light [5] [6] [7]**.**Secondly, the frame differencing method is used to detect the motion object areas in the edge maps obtained by the first method, by comparing the no. of non zero pixels in the difference image with a threshold value. The obtained motion areas are then mapped to the original image and appropriate edge pixels are highlighted. Lastly a dynamic probe window based optimization is applied so to reduce the computational complexity of detection process ,making it suitable for various applications where computing requirements are limited such as home security systems .

2. EDGE DETECTION

2.1 Edge Detection method

Edge Detection [6][7]is a process of identifying and locating sharp discontinuities in an image. The discontinuities are abrupt changes in pixel intensity which characterize boundaries of objects in a scene. Classical methods of edge detection involve convolving the image with an operator (a 2-D filter), which is constructed to be sensitive to large gradients in the image while returning values of zero in uniform regions. Among the edge detection methods proposed so far, the canny edge detector [9] is the most strictly defined operator and is widely used. Its optimality in terms of the three criteria: 1. Good detection 2. Good localization, 3. Single response to an edge has made it popular.

In Canny edge detection algorithm [8][9],edge detection is basically performed by: Smoothing, Differentiating and Tresholding. The computation of the gradient of an image has been performed by obtaining the partial derivatives in x and y directions by means of *Roberts, Prewitt* and *Sobel* operators [11] as a standard manner. Although mathematically the gradient of a function of two variables could be approximated by using numeric differentiation techniques. The gradient operator generally introduces noise in image, which is fundamental problem with gradient based applications. In our algorithm we have used the sobel operator [12][13] because of its filtering attributes.

2.2 Gradient of image

The most common method of differentiation in image processing application is the gradient operator [11][12] as the gradient vector points in the direction of maximum rate of change of f at (x,y) is the basis for various approaches in image differentiation. For a function *f(x, y),* the gradient of *f* at coordinates (x, y) is defined as the vector

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial x} \end{bmatrix} \quad \text{The magnitude of this vector} \nabla f = mag(\nabla \mathrm{f}) = \left[G_x^2 + G_y^2\right]^{1/2}$$

gives maximum rate of change of f at (x,y) while the angle with respect to the x axis $\alpha(x, y) = tan^{-1}(G_y / G_x)$ gives the direction of change of f at (x,y). The computation of the gradient of an image is based on obtaining the partial derivatives $\partial f / \partial x$ and $\partial f / \partial y$ at every pixel location. These partial derivatives may be implemented in digital form in several ways: *Roberts, Prewitt* and *Sobel* [12][13] whose smoothing effect is a particularly attractive for gradients as derivative enhances noise. Figure 1 (a) and (b) show *Sobel* operators for G_x and G_y respectively

-1	*-2*	*-1*
	x,y	
1	*2*	*1*

(a)

-1		*1*
-2	*x,y*	*2*
-1		*1*

(b)

Figure 1: Sobel operator for: a) G_x b) G_y

Sobel operator is more sensitive to diagonal edges than vertical and horizontal edges but having good filtering attributes, so we have used it for our application part.

3. FRAME DIFFERENCING METHOD BASED ON EDGE DETECTION

Frame differencing method attempts to detect moving regions by making use of the difference of consecutive frames (two or three) in a video sequence. This method is highly adaptive to static environment, so frame differencing is good at providing initial course motion areas [5].But the frame differencing method is prone to noise or the change in the illumination level of the scene.So, the method is still prone to false detection due to change in background of the image or large change in the background illumination level.

To overcome this shortcoming of frame differencing we use two edge maps to compute the difference image. In frame differencing, the unchanged part is eliminated while the changed part remains. This change might be caused by noise also but this edge based differencing technique is resistant to noise and change in scean illumination since the edge has no relation with brightness. The basic steps for finding the difference of two edge maps are as follows:

1) A Simple Method for Motion Detection is the Subtraction of two or more images in a given sequence.
2) Now, we call this Difference Image as D(x,y)

Where any non zero value will indicate the areas of motion.

3) Let the two edge maps be $EDGE_{k-1}(x,y)$ & $EDGE_k(x,y)$ for two consecutive frames
   $D(x, y) = EDGE_{k-1}(x,y) - EDGE_k(x,y)$
   $D(x,y) = 0$ (If $EDGE_{k-1}(x,y) = EDGE_k(x,y)$) and = 1 (Otherwise)
4) Calculate the no. of pixels in D(x, y) with binary 1 & compare with the threshold value E. If | D[x, y] | >= E then “Trigger a Motion Change“ Else “Ignore the change “
5)

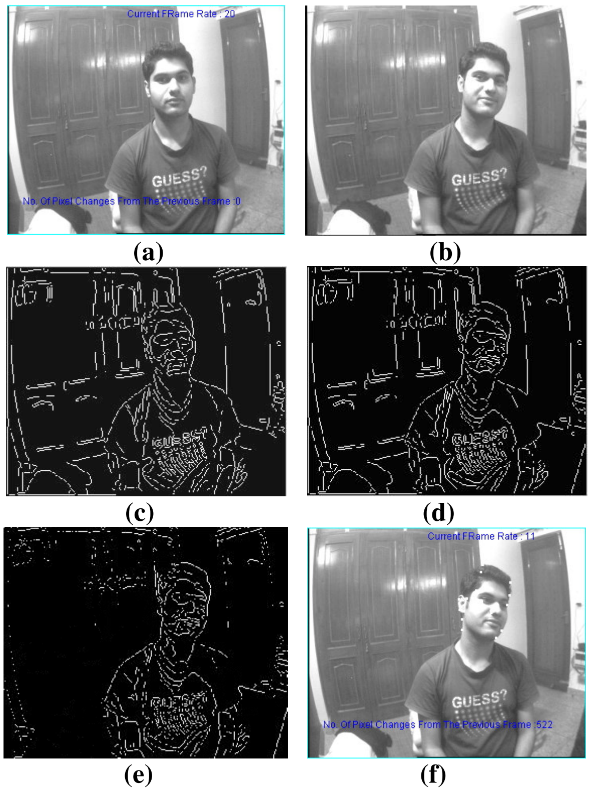

Current FRame Rate : 20
No. Of Pixel Changes From The Previous Frame :0
(a)
(b)
(c)
(d)
(e)
Current FRame Rate : 11
No. Of Pixel Changes From The Previous Frame :522
(f)

Figure 2:

(a) Frame(k-1) taken at time t
(b) Frame(k) taken at time t+1
(c) Edge map($EDGE_{k-1}$) of Frame(k-1)
(d) Edge map($EDGE_k$) of Frame(k)
(e) Edge difference image D(x,y) highlighting the difference of Edge maps of Frame(k-1) and Frame(k)
(f) Obtained motion areas are then mapped to the original image and appropriate edge pixels (white pixels) are highlighted.

Figure 2(e) shows the difference of two edge maps of Figure 2(c) and Figure 2(d) .This difference image is then mapped to the original image (Figure 2(b)) highlighting the pixels along the edges which have changed from their initial position . Thus we get the image in Figure 2(f) which highlights the change pixels above the threshold value .

From the above result we can see that the frame differencing based on the edge detection is a simple method for detecting for moving objects and gives better results.

4. DYNAMIC PROBE WINDOW (DPW) BASED APPROACH

As we have seen, edge detection along with frame differencing reduces the computational complexity of comparing the current and previous frames i.e. computing the pixels that are only on the edges thereby reducing the no. of comparisons. Further more, this algorithm apart from the above optimization doesn't consider the difference between the scenario's when motion occurs and when motion doesn't occurs i.e. same amount of computations(CPU utilization)is done even if the motion is not there.

So, we consider a scenario of Home Surveillance System [2][3] where we assume that motion in the area of surveillance doesn't occurs for the long period of time. During this scenario also, the improved method performs the same procedure-i.e. computing the edge and then comparing only edge pixels.

Now, we propose an approach 'Dynamic Probe Window' based on robust statistics (Robust statistics are those that tends to ignore the data far away from the region of interest) which assumes a optimistic view, that most of the time motion is not encountered. Upon analysis, the method has minimum CPU utilization and simultaneously maintaining higher detection accuracy. The steps of the new algorithm presented in this paper are as follow:

(1). Initialize the probe window to default video size i.e. as AREA (W, H)

(2).Compute the edge map $EDGE_{k-1}$ and $EDGE_k$ of two continuous frame $(k-1)^{th}$ and frame $(k)^{th}$ frame using the first order differential sobel operator of AREA(W,H).

(3). The difference image D(x, y) is computed by taking difference between the two computed edge maps.(i.e. $D(x, y) = EDGEk - EDGE_{k-1}$) this giving us the course motion areas

(4). No. of change pixels (non zero pixels) $| D(x, y) |$ is compared with a threshold value. If the no. of changed pixels is greater than the threshold value 'E', a motion alarm is triggered.

(5). Now considering an optimistic view that motion doesn't occurs for most of the time or the value $| D(x, y) | < E$ (Threshold) → No motion change detected.

(6). We defined the probe window as the area of the video analyzed by the algorithm. In case of no motion, shrink (reduce) the probe window AREA (W, H),

Width by $W = \frac{W}{FPS}$ and Height by $H = \frac{H}{FPS}$ where FPS is the current video frame rate.[Figure 3 shows the shrinking probe window in case of no motion]

(7). So, the probe window keeps on gradually reducing taking an optimistic view that no motion is encountered and In case a motion is encountered within the probe area, the probe window is reset to the default video size and the algorithm proceeds in the usual way. To decrease the probability of non detection due to reduction in the probe window, the threshold is also made dynamic i.e. it decreases with the decrease in the probe window size, making it more sensitive to change in scene.

The threshold [4] can be viewed as $E = \mu \pm 3\sigma$ where μ is the mean f the difference image D(x,y) within the probe window and σ is the standard deviation.

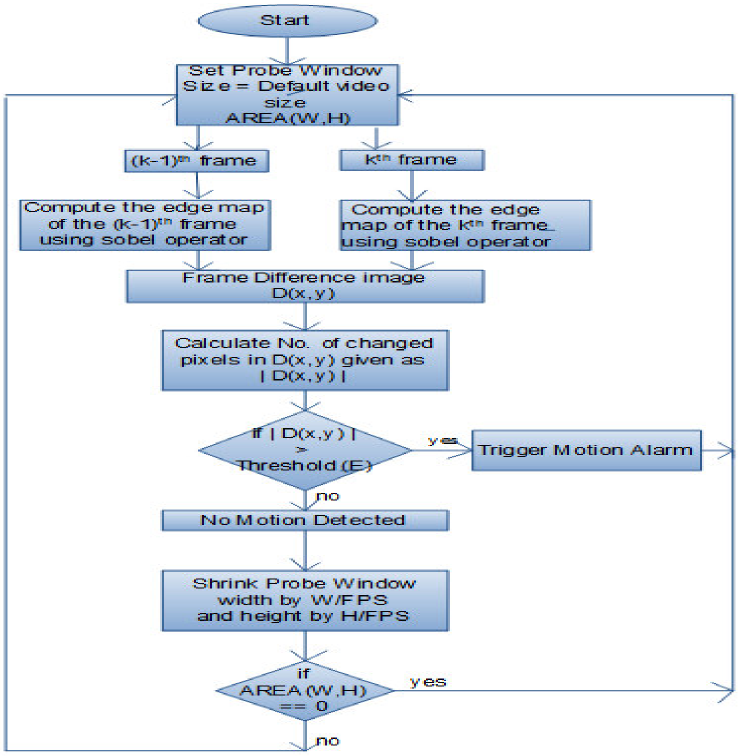
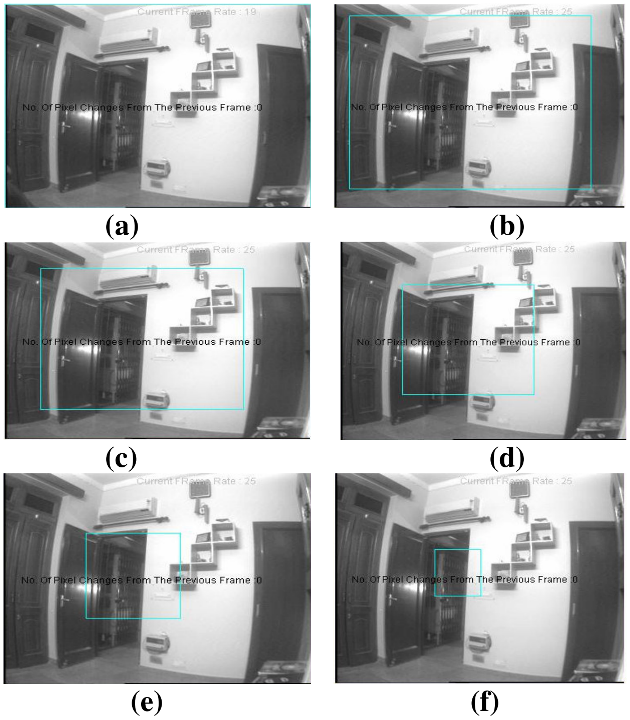

Start
Set Probe Window Size = Default video size AREA(W,H)
(k-1)th frame
Kth frame
Compute the edge map of the (k-1)th frame using sobel operator
Compute the edge map of the kth frame using sobel operator
Frame Difference image D(x,y)
Calculate No. of changed pixels in D(x,y) given as | D(x,y) |
if | D(x,y) | > Threshold (E)
yes
Trigger Motion Alarm
no
No Motion Detected
Shrink Probe Window width by W/FPS and height by H/FPS
if AREA (W,H) == 0
yes
no

Figure 3: Flow chart for the improved algorithm based on Dynamic Probe Window Optimization



(a) (b)

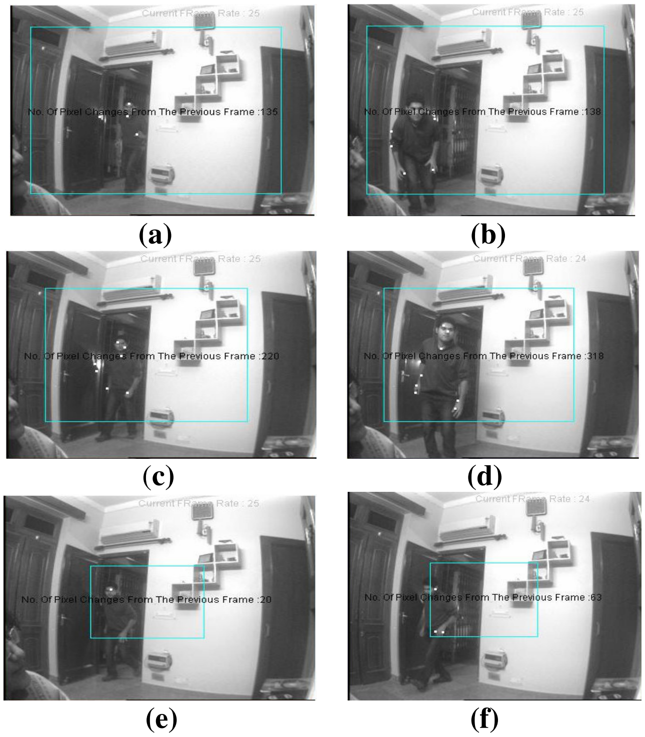
(c) (d)

(e) (f)

Figure 4: Images from (a-g) showing the gradually reducing size of Probe Window Area (Reduction by a factor of W/FPS for Width and H/FPS for Height) when no motion is detected.

5. EXPERIMENTAL RESULTS AND ANALYSIS

In this paper, an improved moving object detection algorithm based on Dynamic Probe Window based optimization is presented. It was tested in a real environment on a 2392.20 MHz PC with 64 MB graphic memory without any dedicated GPU. Results are as follows:


Current Frame Rate : 25
No. Of Pixel Changes From The Previous Frame :135
Current Frame Rate : 25
No. Of Pixel Changes From The Previous Frame :138
Current Frame Rate : 25
No. Of Pixel Changes From The Previous Frame :220
Current Frame Rate : 24
No. Of Pixel Changes From The Previous Frame :318
Current Frame Rate : 25
No. Of Pixel Changes From The Previous Frame :20
Current Frame Rate : 24
No. Of Pixel Changes From The Previous Frame :63

(a) (b)

(c) (d)

(e) (f)

Figure 6: Results with dynamic probe window method with moving object

Images from 6(a-b) showing motion detection even in case of shrinking probe window (where white pixels denotes the moving edges).Images (c-d) showing Reduced Probe window Area and still motion is detected. Images (e-f) showing the further reduction in Probe window area and still motion is detected.

No Motion Analysis- Figure 5 shows an average CPU Utilization of 87.16 % with Edge & Frame Differencing method, even when no motion is there & Figure 9 shows an average CPU Utilization of 46.64 % with dynamic probe window optimization.

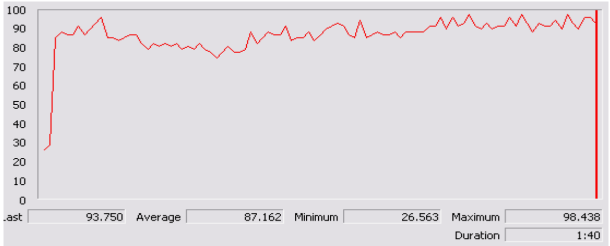


Figure 7: CPU Utilization in case of no motion in Edge + Frame Differencing Based Algorithm (Assuming a video frame rate of 30 FPS)

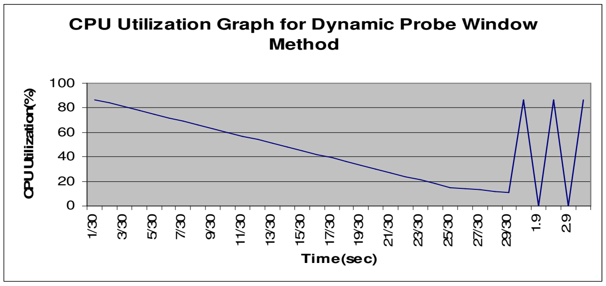


Figure 8: CPU Utilization in case of no motion using Adaptive Probe Window with Edge + Frame Differencing Based Algorithm (Assuming a video frame rate of 30 FPS)

6. CONCLUSION

This paper presented an improved motion detection algorithm based on frame differencing and edge detection along with dynamic probe window based optimization. Experimental results showed that the algorithm can detect moving objects precisely in read time (30 frames per second) along with less computation complexity.

The probe window size gradually decreases when no motion is detected and hence the probability of detecting a new motion gradually decreases, but this change is as fast as 1/30th of a second .So even with this method, the motion does not goes undetected in most of the cases .

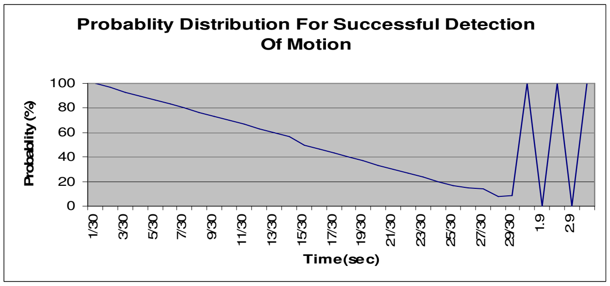


Figure 9: Probability of detecting an object during the shrinking phase of probe window

In case of Dynamic Probe window with adaptive threshold, the probability of detection of motion is little bit improved due to decrease in threshold (so increased sensitivity) even if the size of the probe window keeps on decreasing gradually .On the other hand in case of motion, the probe window size never decreases, so the probability of detecting the change is always one.

7. REFERENCES

[1] Video Motion Detection for Real-Time Hazard Warnings in Surface Mines. Edward L. McHugh Spokane Research Laboratory, National Institute for Occupational Safety and Health, Spokane, Washington, USA

[2] C. Bregler: Learning and recognizing human dynamics in video sequences. In Proc. *IEEE Int. Conf. CVPR'97*, pages 568-574, 1997.

[3] Fujiyoshi, and R. S. Patil: Moving target classification and tracking from real-time video. In Proc.*IEEE Workshop Application of Computer Vision*, pages 8-14, 1998.

[4] Nan Lu, Jihong Wang and Q.H. Wu and Li Yang , “An Improved Motion Detection Method for Real-Time Surveillance”, IAENG International Journal of Computer Science, 35:1, IJCS_35_1_16, Advance online publication: 19 February 2008

[5] A Moving Detection Algorithm Based on Space-Time

Background Difference Mei Xiao1, Lei Zhang2, and Chong-Zhao Han1. D.S. Huang, X.-P. Zhang, G.-B. Huang (Eds.): ICIC 2005, Part I, LNCS 3644, pp. 146 – 154, 2005.

[6] M. Julius Hossain, M. Ali Akber Dewan, and Oksam Chae,” Suitability of Edge Segment Based Moving Object Detection for Real Time Video Surveillance”, B. Apolloni et al. (Eds.): KES 2007/WIRN 2007, Part I, LNAI 4692, 2007.© Springer-Verlag Berlin Heidelberg 2007, pp.526-533

[7] Ahn, K.O., Hwang, H.J., Chae, O.S.: Design and Implementation of Edge Class for Image Analysis Algorithm Development based on Standard Edge. In: Proc. of KISS Autumn Conference,2003, pp. 589–591

[8] Gang Liu and R.M. Haralick. Two practical issues in Canny’s edge detector implementation *Proceedings of 15th International Conference on Pattern Recognition*. Volume 3, 3-7, 2000,676–678 Sept.

[9] J. Canny. A Computational approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 1986.

[10] Ren Mingwu, and Sun Han, “A Practical Method for Moving Target Detection Under Complex Background”. Computer Engineering, Oct 2005, pp. 33-34.

[11] J. Guo, D. Rajan and E.S. Chng, “Motion detection with adaptive background and dynamic thresholds,” *2005 Fifth International Conference on Information, Communications and Signal Processing,* 06-09 Dec. 2005, pp. 41-45

[12] J. Lopez, M. Markel, N. Siddiqi, G. Gebert and J. Evers, “Performance of passive ranging from image flow,” IEEE International Conference on Image Processing, vol. 1, 2003, pp. 929-932.

[13] Design of an image edge detection filter using the Sobel operator Kanopoulos, N.; Vasanthavada, N.; Baker, R.L. Solid-State Circuits, IEEE Journal of Volume 23, Issue 2, Apr 1988 Page(s):358 – 367

Bhaskar Kapoor is working as Lecturer in Maharaja Agarsen Institute of Technology, NewDelhi,He has done ME(CTA) from Delhi College of Engineering. He is Lifetime Member of ISTE.He Teaches Courses for B.Tech,BCA and MCA. His research areas of interest include Computer Graphics Vision & Multimedia, Human Computer Interaction,and Network Security. He has presented and published various research papers in national and international Journals and conferences.

Anamika Chhabra has done M.Tech(IT) from Guru Gobind Singh Indraprastha University,Delhi.She is a gold Medalist in B.Tech(IT) and a silver Medalist in M.Tech(IT), and is also GATE qualified. Her areas of interest are Computer graphics and Object Oriented Software Engineering. She has presented and published various research papers in national and international conferences.