

# Implementation of Smart Video Surveillance System Using Motion Detection Technique

Gaurav Verma, Swati Gautam, Rohan Agarwal, Sarthak Saxena and Deval Verma

**Abstract** In today's world, there is a growing need of protection of one's property—security is one major issue which should be taken care of by everyone. There is a lot of increase in the number of thefts or stealing with the advancement of technology and so this directs our attention toward protecting our property. The most popular way of protecting our property would be a video surveillance and constant monitoring. Since technology has both boon and harm, we can use it to develop a system to improve the functionalities of existing video surveillance system, which can track the movements of the suspicious people, generate an alarm, and even record them from the time of suspicious activities. Our main motive is to develop a real-time application for the purpose of video surveillance which will not only detect motion, but also inform the user of any kind of suspicious activity, record from the moment of any kind of activity, and also turn the camera in the direction of motion. We will be doing it with the help of a MATLAB version 2014a.

**Keywords** Video surveillance · Background subtraction · Blob analysis · Arduino · Motor rotation

---

G. Verma (✉) · S. Gautam · R. Agarwal · S. Saxena · D. Verma  
Electronics and Communication Engineering, Jaypee Institute  
of Information Technology, Noida, Uttar Pradesh, India  
e-mail: gaurav.iitkg@gmail.com

S. Gautam  
e-mail: gswati025@gmail.com

R. Agarwal  
e-mail: 23rohan.agarwal@gmail.com

S. Saxena  
e-mail: ssarthak.9424@gmail.com

D. Verma  
e-mail: deval09msc@gmail.com

## 1 Introduction

Representation of an image can be in the form of two-dimensional function and can be written as  $f(x,y)$ , i.e., the function of  $(x,y)$  where  $x$  and  $y$  are coordinates and magnitude/amplitude of this point is called intensity/gray level of image at these coordinates. When the magnitude values of function  $f$  are finite implying discrete values, the image hence formed is a digital image [1]. Digital Image Processing is the manipulation of digital images by the use of computer algorithms. Digital Image Processing is better than analog image processing as it allows a much wider range of computer algorithms to be applied to the input image. Digital Image Processing is a subset of signals and systems and emphasizes only on images [4, 5]. In order to convert analog signals to digital images, we require the use of various computer algorithms. Digital Image Processing is based on the use of such algorithms. They improve clarity, image compression ability across networks for communicating purposes, size extraction, etc., [3]. Digital images can be enhanced in many ways, for example, we can increase the brightness, convert the image to black and white and vice versa, even the colors can be changed for further processing [2]. MATLAB helps us to store images as two-dimensional arrays or matrices. Each element of the matrix represents one pixel in the image. Let us say we have an image of 500 rows and 500 columns. This 500 x 500 will represent a matrix having 10,000 different colored dots or pixels to be more precise. A pixel is combination of RGB, i.e., red, green, and blue components. But in case of RGB, the first plane in the third dimension represents red pixel depth, the second plane is for green pixel depth, and the third plane is for blue pixel depth or intensities [3]. Motion detection is one of the most popular areas which help in developing a security system. The most popular technique used in image processing and computer vision is background subtraction. In this process, we extract the foreground of the image required for various purposes during processing. And hence, it is also known as foreground detection. Since the moving objects like human bodies in the foreground of images are the main things of our concern, background subtraction is widely used approach in “motion detecting” field. The main principle behind such technique is subtracting the current frame from previous frame [6]. The organization of this paper is such that we have an introduction in Sect. I, then we have objective of the paper in Sect. II, system model in Sect. III, acquisition setup in Sect. IV, image processing in Sect. V, experimental results in Sect. VI, applications in Sect. VII, and conclusion in Sect. VIII.

## 2 Objective

The requirement of a smart video surveillance system provoked us to develop this project which will capture any kind of motion and thus will help us prevent any kinds of thefts or stealing at the very initial stage and will also provide us evidence

in the form of video recording. We will be well prepared for future by installing this system [2].

### 3 System Model and Algorithm

MATLAB (MATrix LABoratory) is used for several mathematical operations and finds its use worldwide. It has the ability to solve a lot many problems related to DSP (digital signal processing) or image processing, it has a vast use in communications, used for various educational purposes, research work, and even industrial work. It has user-friendly interface and is developed by MathWorks. MATLAB provides variety of toolboxes for image handling purposes. The toolboxes involved were image acquisition toolbox (for grabbing the image), image processing toolbox, and computer vision system toolbox (for blob analysis).

The algorithm is as follows:

- Start capturing the video.
- Initialize the RGB components of the first frame as zero. Store the components in a temporary matrix.
- Turn the image by 180 degrees. For this purpose, we can use the flip functions. (We are doing this because we need to store the correct image and not the mirror image.)
- Compare two successive frames (means comparing the RGB components of each frame  $\rightarrow$  matrices). Perform background subtraction.
- If motion produced (means if there is nonzero value achieved after subtraction), then recording starts, alarm beeps.
- At the same time, performing median filtering to highlight the major patches produced during subtraction. Median filtering is required to smooth the edges, highlight them, and keep the major patches or “blobs” with which motion is concerned in the differenced image produced after image subtraction.
- RGB to grayscale conversion: A pixel in the 8-bit grayscale image has magnitude or intensity values ordering from 0 to 255. This means a pixel can have any of these 256 values. People confuse gray scale with black and white image. In actual, a gray scale is an image which contains many shades of gray. Since the conversion of RGB to gray scale is as good as moving from 3-D operations to 1-D operations, therefore in order to reduce our load of work in dealing with the three components of RGB, we switch to gray scale, i.e., from an intensity of 24 bit ( $8 \times 3$ ) to an intensity of 8 bit.
- Gray scale to binary image: For converting, we use `im2bw()` function. Suppose we have set a level or threshold, then all the pixels whose intensity values are above that level will be treated as value 1 and all other pixels will be replaced by value 0. Zero corresponds to white color and one corresponds to black color. Thus in this case, further values are reduced to two—0 and 1.

- Blob analysis to find the centroid of the blobs and store the centroids in a matrix.
- Centroid subtraction.
- Pass the differenced pixel values or say manipulated values to Arduino.
- Motor rotation (which in turn rotates camera).

## 4 Acquisition Setup

For acquiring image/video, we require image acquisition toolbox. With the help of this toolbox, we can detect our hardware easily and set properties of the hardware according to our requirements. The basic steps are as follows:

1. Accessing the webcam's video input.
2. Acquiring the image and storing it in the form of a variable in MATLAB.
3. Using the acquired image for further processing [7].

[8] First step for connecting the image acquisition device in MATLAB is creating a video input object. The video input object establishes the connection between image acquisition device and MATLAB. There are several object-related properties that can be used for the purpose of acquisition and controlling its various aspects. A proper procedure has to be followed before creating the object. First of all, we need details about the device to which we are establishing a connection. In order to access the device, the image acquisition toolbox should have details about the name of adaptor (adaptor is the interface which is required for establishing connection between image acquisition devices connected to the system and MATLAB), the device ID, video format, etc. So, an important step is to acquire information about our available hardware on MATLAB. For this purpose, we use the "imaqhwinfo" function which returns information on available hardware. This comes under the image acquisition toolbox. It should be noted that "Start" function should be used to start the object before starting acquisition of data.

## 5 Image Processing

The toolboxes which are used after acquiring the image are image processing toolbox & computer vision systems toolbox having following features: Image analysis including segmentation, image enhancement, filtering, and de-blurring, image transforms, including FFT etc. Provides functions such as "rgb2gray", "imsubtract", "imread", "imwrite" etc. [9–12]. Computer vision system toolbox supports large videos or handles large videos. It helps us to estimate motion using various functions such as BlobAnalysis, VideoPlayer, and ShapeInserter. We required it for various purposes—blob analysis, recording video functions.

## 6 Experimental Results

The following are the results observed at the end of the experiment. We took four screen shots—for moving the camera in the downward, left, right, and upward direction. For example, when the moving body reached the rightmost edge of the frame, manipulations of the centroid sent the command to the Arduino, which made the motor move in the right direction (Figs. 1 and 2).

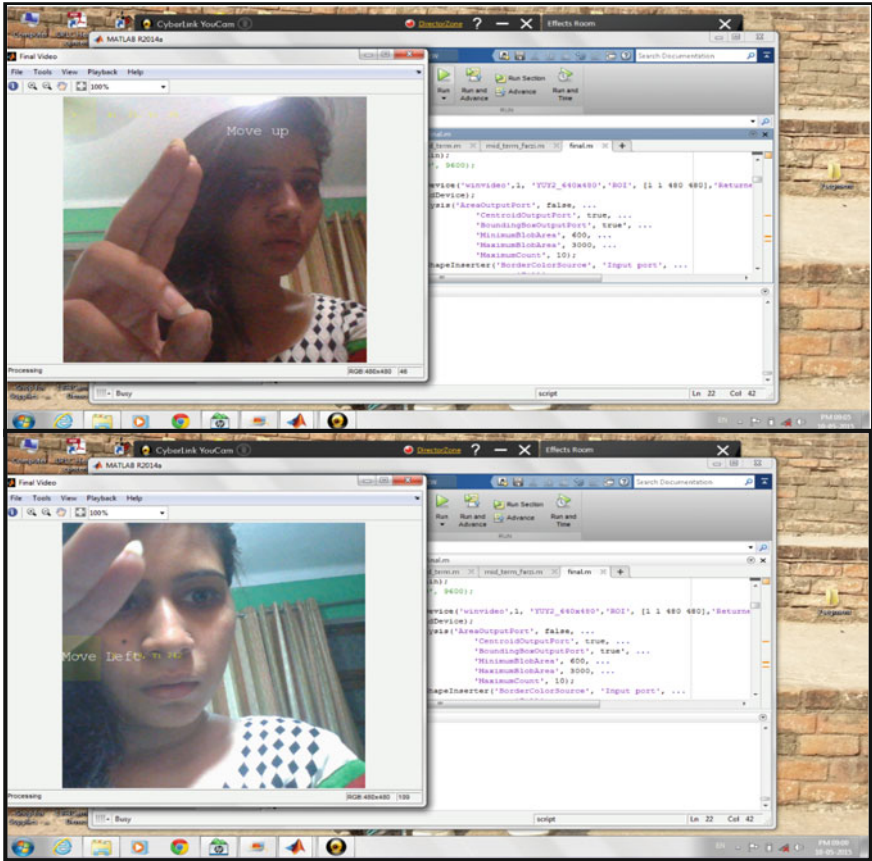


Fig. 1 Showing camera movement in up and left direction

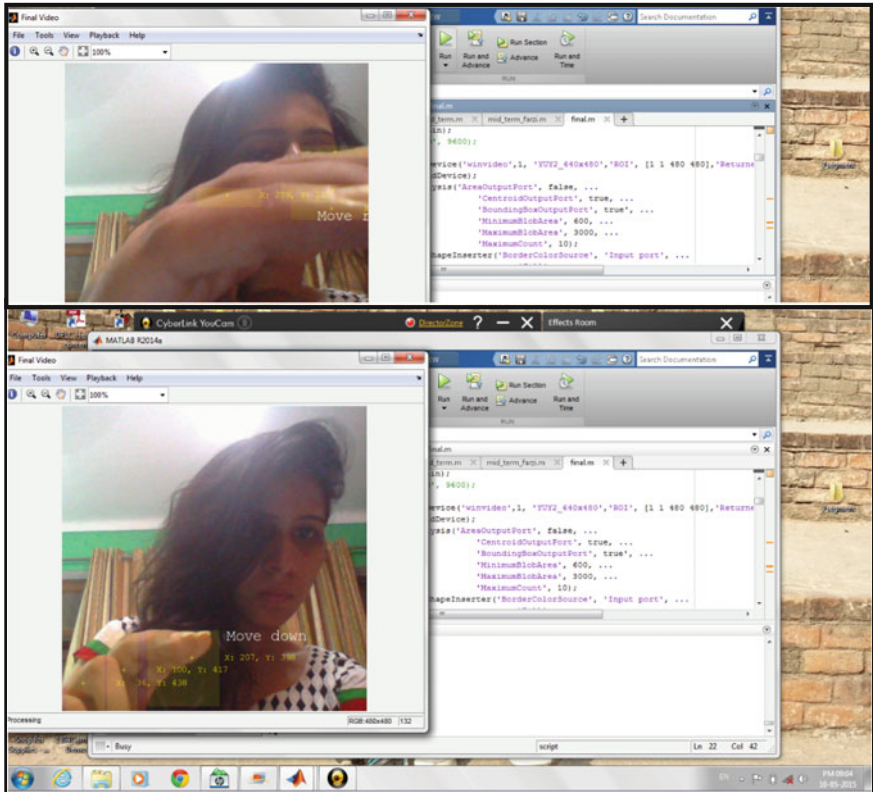


Fig. 2 Showing camera movement in down and right direction

## 7 Applications

Our main objective is developing this smart surveillance system which will not only detect motion, but also move the camera in the direction of motion. Also, if the motion is in the restricted area or if there is mishandling with the camera, then generate an alarm informing the user of any suspicious activity and also record the video for future purposes meant for processing. This can be used in motion sensing games. It can be of great help to the camera man in cricket matches. Since in this project, we are moving the camera according to the motion, it can reduce the manual work of the camera man to follow the ball with such heavy cameras. We can do this by monitoring the camera man's motion to follow the ball. It increases the range of area of camera to be observed (since the camera is rotating).

## 8 Conclusion

We were able to build a video surveillance system for detecting the motion and thus preventing or lessening the amount of thefts or stealing and thus providing a sense of security to people or any organization and also providing the proof of any mishappening in the form of mp4 extension file. The work can be further extended to provide more additional functionalities. These are as follows: The video can be transferred to the Internet to be viewed by everyone on the cloud servers. This underlines the concept of “INTERNET OF THINGS.” The details can be viewed on the smart phones as well. A log-in system can be implemented so that the software can be used only by the authorized people. A proper GUI can be prepared for the layman user so that there is no problem in understanding as to how to run the application. Inbuilt audio recorders, so as to record sound too can be added.

## 9 Consent

We don't have any objection to publishing this paper as a chapter with the images shown above. As the images used are of one of the co-authors of this paper Ms. Swati Gautam.

## References

1. <http://elearning.vtu.ac.in/17/e-Notes/DIP/Unit1-SH.pdf>
2. A. Ansari, T.C. Manjunath (Ph.D., IIT Bombay) and C.Ardil, Implementation of a Motion Detection System, International Journal of Electrical, Computer, Electronics and Communication Engineering 2 9, (2008)
3. [www.mathworks.com](http://www.mathworks.com)
4. <http://www.tutorialspoint.com/dip/>
5. Suresh D.S and Lavanya M.P.: Motion Detection and Tracking using Background Subtraction and Consecutive Frames Difference Method, International Journal of Research Studies in Science, Engineering and Technology, 1(5), 16–22, ISSN 2349–4751 (Print) & ISSN 2349–476X (Online) August (2014)
6. Jeevarajan, Matlab Code for Background Subtraction: <https://www.pantechsolutions.net/blog/matlab-code-for-background-subtraction/#Posted> on December 18, 2013
7. Image Acquisition Toolbox User's Guide by Mathworks, MATLAB
8. Getting Hardware Information: [www.mathworks.com](http://www.mathworks.com), © 1994–2015 The MathWorks, Inc
9. Image Processing Toolbox User's Guide by MathWorks
10. Introduction to image processing by ESA/Hubble: [https://www.spacetelescope.org/static/projects/fits\\_liberator/image\\_processing.pdf](https://www.spacetelescope.org/static/projects/fits_liberator/image_processing.pdf).

11. T. Gupta, G. Verma: “Area & Power Optimization of VPB Peripheral Memory for ARM7TDMI Based Microcontrollers” in International Conference on Cognitive Computing and Information Processing (CCIP-2015) March 3–4, 2015 JSSATEN, Noida, India
12. G. Verma, V. Verma, S. Jhambhulkar, H. Verma, Presented a Paper “Design of a Lead-Lag Compensator for Position Loop Control of a Gimballed Payload” in International Conference on Signal Processing and Integrated Networks (SPIN-2015) February 19–20, 2015 Amity University, Noida, India