A PROJECT REPORT

ON

OBJECT TRACKING USING MODIFIED BACKGROUND SUBTRACTION

Submitted by

M.HARISHA (2451-12-735-064)

SAMALA RISHIKA (2451-12-735-074)

In the partial fulfillment for the award of the degree

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the guidance of

Ms. K.V.B.L. DEEPTHI

(Assistant Professor, Department of ECE)

M.V.S.R. ENGINEERING COLLEGE

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

M.V.S.R. ENGINEERING COLLEGE

Sponsored by Matrusri Education Society, Estd. 1980
Affiliated to. Osmania University & Recognized by AICTE.
Nadergul, Hyderabad - 501 510.
Saroornagar Mandal, R.R. District, Telangana.
2015-16

CERTIFICATE

This is to certify that the project work entitled **"OBJECT TRACKING USING MODIFIED BACKGROUND SUBTRACTION"** is a bonafide work carried out by

M.HARISHA (2451-12-735-064)

SAMALA RISHIKA (2451-12-735-074)

in partial fulfillment of the requirements for the award of degree of **BACHELOR OF ENGINEERING** in **ELECTRONICS AND COMMUNICATION ENGINEERING** by the **OSMANIA UNIVERSITY,** Hyderabad, under our guidance and supervision during the academic year 2015-16.

Ms. K.V.B.L. DEEPTHI
Assistant Professor, ECE
(Internal Guide)

Dr. V. CHANDRA SEKHAR
Head of the Department, ECE

(EXTERNAL EXAMINER)

CERTIFICATE

This is to certify that the major project work entitled **"OBJECT TRACKING USING MODIFIED BACKGROUND SUBTRACTION"** is a bonafide work carried out by the following students of M.V.S.R Engineering College(affiliated to Osmania University) in the partial fulfillment for the award of degree of Bachelor Of Engineering(B.E.) during the academic year 2015-16

M.HARISHA (2451-12-735-064)

SAMALA RISHIKA (2451-12-735-074)

This project work is guided by the following officers of Lasers and Optics lab, Design And Engineering Division, BHARAT DYNAMCS LIMITED, Hyderabad.

(SAGAR HARNOL)	**(R.SUDHAKAR RAO)**
Deputy Manager	**Senior Manager**

DECLARATION

This is to certify that the work reported in the present project entitled **"OBJECT TRACKING USING MODIFIED BACKGROUND SUBTRACTION"** is a record of work done by us in the Department of Electronics and Communication Engineering, M.V.S.R. Engineering College. The reports are based on the project work carried out by us and not copied from any other source.

M.HARISHA	**(2451-12-735-064)**
SAMALA RISHIKA	**(2451-12-735-074)**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

We express our profound gratitude to the management of **Bharat Dynamics Limited**, Hyderabad for giving us the opportunity to do this project.

We are indebted to our project guide **Mr. SAGAR HARNOL**, Deputy Manager (Lasers and Optics lab – D&E), B.D.L. for his unabridged support, encouragement, continuous guidance for the successful completion of the project on time.

We extend our sincere gratitude towards **Dr. V. CHANDRA SEKHAR**, Head of the department of Electronics and Communication Engineering for giving us his valuable knowledge and wonderful technical guidance.

We are greatly thankful to our internal project guide **Ms. K.V.B.L. DEEPTHI**, Assistant Professor, ECE, for her constant guidance, encouragement and interest throughout the course of this project.

We are also thankful to **Ms. B.SARALA**, Associate Professor, ECE, **Mr. K.NARASIMHA RAO,** Assistant Professor, ECE and, **Ms. K.V.B.L. DEEPTHI**, Assistant Professor, ECE, for their reviews and valuable suggestions throughout the project.

We are also thankful to our project coordinators **Mr. G.SRAVAN KUMAR and Ms. K.V.B.L. DEEPTHI**, Assistant Professors of ECE Dept., for their valuable suggestions and administrative help.

Finally we would like to take this opportunity to thank our families for their support through the work. We sincerely acknowledge and thank all those who directly and indirectly have given their support in completion of this work.

M.HARISHA (2451-12-735-064)

SAMALA RISHIKA (2451-12-735-074)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Tracking is a common procedure in application domains such as surveillance and visual servoing. Its aim is to match entities that appear in different frames of a sequence. The first step in the tracking includes capturing a video of a moving object using a camera. It is followed by sampling the video into frames which are sent into image processing block, where various methods are used for dynamic object tracking and detection. Among them “BACKGROUND SUBTRACTION” method is widely used for static backgrounds. It is a simple technique of subtracting the background model from the current frame to obtain difference image and by thresholding the difference frame, a mask of moving object in the current frame is obtained. The entire process is done in a pixel-by-pixel fashion. This method makes the assumption that every frame in a sequence is made up of a fixed background with moving objects on it and the pixels of the foreground object have different gray scale intensities from those of background pixels. However, the major drawback of this traditional background subtraction method is that it only works for static background and is susceptible to environmental changes. Therefore background model update is required. Hence in this project we propose a method for dynamic background scenes and in cases of gradual or sudden illumination changes. This is followed by morphological processing to remove noises and jitters and finally the position and shape of the object are detected.

This method is best suitable for object tracking within small areas and large field of view. It is of low cost and high efficiency. This is suitable for the real time surveillance system because it has fast computation and it is robust against environmental disturbances. By locating and tracking moving objects in a video sequence in real time, we can develop real time alert system to enhance current surveillance system.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO TRACKING SYSTEMS

Tracking can be defined as the mechanism of continuously maintaining the path of the object moving and keeping track of its motion, orientation, occlusion etc. in order to extract useful information. The block diagram of tracking system is shown in the fig.1.1. Generally a camera may be a CCD or a CMOS camera is used to record the moving object path. From the camera the video recorded is given to a high speed frame grabber which is an electronic device that captures (i.e., grabs) individual, digital still frames from an analog video signal or a digital video stream. It is usually employed as a component of a computer vision system, in which video frames are captured in digital form and then displayed, stored or transmitted.

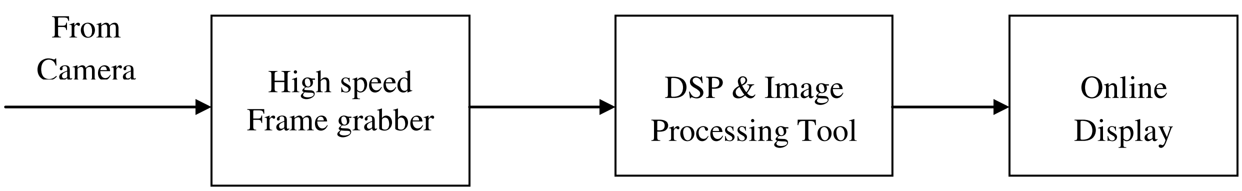


Fig 1.1- Block Diagram of Tracking System

After grabbing the recorded output is transmitted to a computer. Image processing and analysis techniques like correlation, segmentation, edge detection and the entire image processing functions are used for processing the image for extracting the moving object information.

This project is to be implemented using MATLAB. Moving objects may be any vehicle, person or any other object. With the help of images captured and through modern image processing tools it is possible to compute the position of an object in a given frame by applying various tracking techniques.

Object tracking is required in many vision applications such as human-computer interfaces, video communication/compression, road traffic control, security and surveillance systems. Often the goal is to obtain a record of the trajectory of the moving single or multiple targets over time and space, by processing information from distributed sensors.

Object tracking in video sequences requires on-line processing of a large amount of data and is time-expensive. Additionally, most of the problems encountered in visual tracking are nonlinear, non Gaussian, multi-modal or any combination of these. Different techniques are available in the literature for solving tracking tasks in vision and can be divided in general into two groups:

i) Classical applications, where targets do not interact much with each other, behave independently such as aircrafts that do not cross their paths.
ii) Applications in which targets do not behave independently (ants, bees, robots, people), their identity is not always very well distinguishable. Tracking multiple identical targets has its own challenges when the targets pass close to each other or merge.

1.2 LITERATURE SURVEY

The word object tracking is creating a lot of interest to research people as it is nowadays used in a wide variety of applications mainly related to security, surveillance, missile tracking and enemy target detection in the navy. Its wide range of applications has created interest in us for choosing this project. Firstly we have gone through many research papers having different algorithms used for object tracking. The following papers we surveyed would give a better idea of different techniques used for object tracking.

A SURVEY ON OBJECT DETECTION AND TRACKING METHODS

Firstly, one need to know about what is tracking and its preceding steps of object detection and object classification. A basic idea of different methods and algorithms existing need to be known before proposing any new algorithm. This paper presents a brief survey of different object detection, object classification and object tracking algorithms available in the literature including analysis and comparative study of different techniques used for various stages of tracking.

A REAL-TIME METHOD TO DETECT AND TRACK MOVING OBJECTS (DATMO) FROM UNMANNED AERIAL VEHICLES (UAVS) USING A SINGLE CAMERA

This paper proposes a new technique of tracking moving objects called **optical flow method**. In this paper they developed a real-time method to detect and track moving objects (DATMO) from unmanned aerial vehicles (UAVs) using a single camera. To address the challenging characteristics of these vehicles, such as continuous unrestricted pose variation and low-frequency vibrations, new approaches must be developed. The main concept proposed in this paper is to create an artificial optical flow field by estimating the camera motion between two subsequent video frames. The core of the methodology consists of comparing this artificial flow with the real optical flow directly calculated from the video feed. The motion of the UAV between frames is estimated with available parallel tracking and mapping techniques that identify good static features in the images and follow them between frames. By comparing the two optical flows, a list of dynamic pixels is obtained and then grouped into dynamic objects. Tracking these dynamic objects through time and space provides a filtering procedure to eliminate spurious events and misdetections.

MOTION-BASED SEGMENTATION AND REGION TRACKING IN IMAGE SEQUENCES

This paper gives another method for tracking called **temporal differencing**. This paper presents an algorithm for segmenting and tracking moving objects in a scene. Temporal information provided by a region tracking strategy is integrated for improving frame-to-frame motion segmentation. The method has been applied to a traffic monitoring system and it provides facilities such as estimating trajectories of vehicles, detecting stopped vehicles, counting vehicles and estimating the mean velocity of the traffic.

AUTOMATIC DETECTION AND TRACKING OF MOVING OBJECTS IN COMPLEX ENVIRONMENTS FOR VIDEO SURVEILLANCE APPLICATIONS

Autonomous video surveillance and monitoring has a rich history. In this paper a new method for detecting multiple moving objects based on **background subtraction** model and for tracking, based on feature based approach are proposed. Then identified moving objects are also counted, by indexing individually. The proposed algorithm is automatic and efficient in intelligent surveillance applications like vehicles monitoring, event recognition, and crime prevention, etc. The proposed model has proved to be robust in various environments (including indoor and outdoor scenes) and different types of background scenes. Experiments on real scenes show that the algorithm is effective for object detection and tracking.

OBJECT TRACKING INITIALIZATION USING AUTOMATIC MOVING OBJECT DETECTION

In this paper, new methods for object tracking initialization using automated moving object detection based on background subtraction are presented. The proposed new background model updating method and adaptive thresholding are used to produce a foreground object mask for object tracking initialization.

Traditional background subtraction method detects moving objects by subtracting the background model from the current image. Compared to other common moving object detection algorithms, background subtraction segments foreground objects more accurately and detect foreground objects even if they are motionless. How-ever, one drawback of traditional background subtraction is that it is susceptible to environmental changes, for example, gradual or sudden illumination changes. The reason of this drawback is that it assumes a static back-ground, and hence a background model update is required for dynamic backgrounds. The major challenges are how to update the background model, and how to determine the threshold for classification of foreground and background pixels. This proposed method determines the threshold automatically and dynamically depending on the intensities of the pixels in the current frame and a method to update the background model with learning rate depending on the differences of the pixels in the background model and the previous frame.

BACKGROUND SUBTRACTION USING A PIXEL-WISE ADAPTIVE LEARNING RATE

This paper presents a new method for object tracking initialization using background subtraction. It proposes an effective scheme for updating a background model adaptively in dynamic scenes. Unlike the traditional methods that use the same “learning rate” for the entire frame or sequence, this method assigns a learning rate for each pixel according to two parameters. The first parameter depends on the difference between the pixel intensities of the background model and the current frame. The second parameter depends on the duration of the pixel being classified as a background pixel. It also introduces a method to detect sudden illumination changes and segment moving objects during these changes. Experimental results show significant improvement in moving object detection in dynamic scenes such as waving tree leaves and sudden illumination change, and it has a much lower computational cost compared to Gaussian mixture model.

WALLFLOWER: PRINCIPLES AND PRACTICE OF BACKGROUND MAINTENANCE

Background maintenance is a frequent element of video surveillance systems. In this paper they develop Wallflower, a three component system for background maintenance: a) the pixel level component performs Wiener filtering to make probabilistic predictions of the expected background; b) the region-level component fills in homogeneous regions of foreground objects; c) the frame-level component detects sudden, global changes in the image and swaps in better approximations of the background. This paper compares developed system with 8 other background subtraction algorithms. Wallflower is shown to outperform previous algorithms by handling a greater set of the difficult situations that can occur. Finally, it analyzes the experimental results and propose normative principles for background maintenance.

ROBUST BACKGROUND SUBTRACTION TO GLOBAL ILLUMINATION CHANGES VIA MULTIPLE FEATURES-BASED ONLINE ROBUST PRINCIPAL COMPONENTS ANALYSIS WITH MARKOV RANDOM FIELD

Background subtraction is an important task for various computer vision applications. The task becomes more critical when the background scene contains more variations, such as swaying trees and abruptly changing lighting conditions. Recently, robust principal component analysis

(RPCA) has been shown to be a very efficient framework for moving-object detection. However, due to its batch optimization process, high dimensional data need to be processed. As a result, computational complexity, lack of features, weak performance, real-time processing, and memory issues arise in traditional RPCA-based approaches. To handle these, a background subtraction algorithm robust against global illumination changes via online robust PCA (OR-PCA) using multiple features together with continuous constraints, such as Markov random field (MRF), is presented in this paper. OR-PCA with automatic parameter estimation using multiple features improves the background subtraction accuracy and computation time, making it attractive for real-time systems. Moreover, the application of MRF to the foreground mask exploits structural information to improve the segmentation results. In addition, global illumination changes in scenes are tackled by using sum of the difference of similarity measure among features, followed by a parameter update process using a low-rank, multiple features model. Evaluation using challenging datasets demonstrated that the proposed scheme in this paper is a top performer for a wide range of complex background scenes.

1.3 ORGANISATION OF THE REPORT

Chapter 1 includes the introduction to tracking systems their applications and corresponding literature survey.

Chapter 2 is about the basics of video acquisition system, its parts, types of cameras and parameters of cameras which helps in understanding of how to capture a video and what are the parameters involved in it.

Chapter 3 includes all the basics of image processing system, its need, applications and different steps involved in processing.

Chapter 4 is about steps involved in tracking, different methods of tracking, our proposed method flowchart and its description.

Chapter 5 deals with the MATLAB software, how go get started with it and introduction to image processing toolbox.

Chapter 6 includes implementation of our proposed code in MATLAB and the results of each step of our proposed flowchart.

Chapter 7 is about Conclusion, Future scope and the references we studied.

Finally the **Appendix** gives the code of our proposed tracking method.

CHAPTER 2
VIDEO ACQUISITION SYSTEM

Video is an electronic medium for recording, copying, playback, broadcasting, and display of moving visual media.

Video systems vary greatly in the resolution of the display, how they are refreshed, and the rate of refreshed, and 3D video systems exist. They can also be carried on a variety of media, including radio broadcast, tapes, DVDs, computer files etc.

2.1 HISTORY OF VIDEO ACQUISITION SYSTEM

Video technology was first developed for mechanical television systems, which were quickly replaced by cathode ray tube(CRT) television systems, but several new technologies for video display devices have since been invented. Charles Ginsburg led an Ampex research team developing one of the first practical video tape recorders (VTR). In 1951 the first video tape recorder captured live images from television cameras by converting the camera's electrical impulses and saving the information onto magnetic video tape.

The use of digital techniques in video created digital video, which allowed higher quality and, eventually, much lower cost than earlier analog technology. Advances in computer technology allowed even inexpensive personal computers to capture, store, edit and transmit digital video, further reducing the cost of video production, allowing program-makers and broadcasters to move to tapeless production. The advent of digital broadcasting and the subsequent digital television transition is in the process of relegating analog video to the status of a legacy technology in most parts of the world.

As of 2015, with the increasing use of high-resolution video cameras with improved dynamic range and color gamut’s, and high-dynamic-range digital intermediate data formats with improved color depth, modern digital video technology is slowly converging with digital film technology.

2.2 BLOCK DIAGRAM FOR VIDEO ACQUISITION, PROCESSING AND SENDING TO COMPUTER

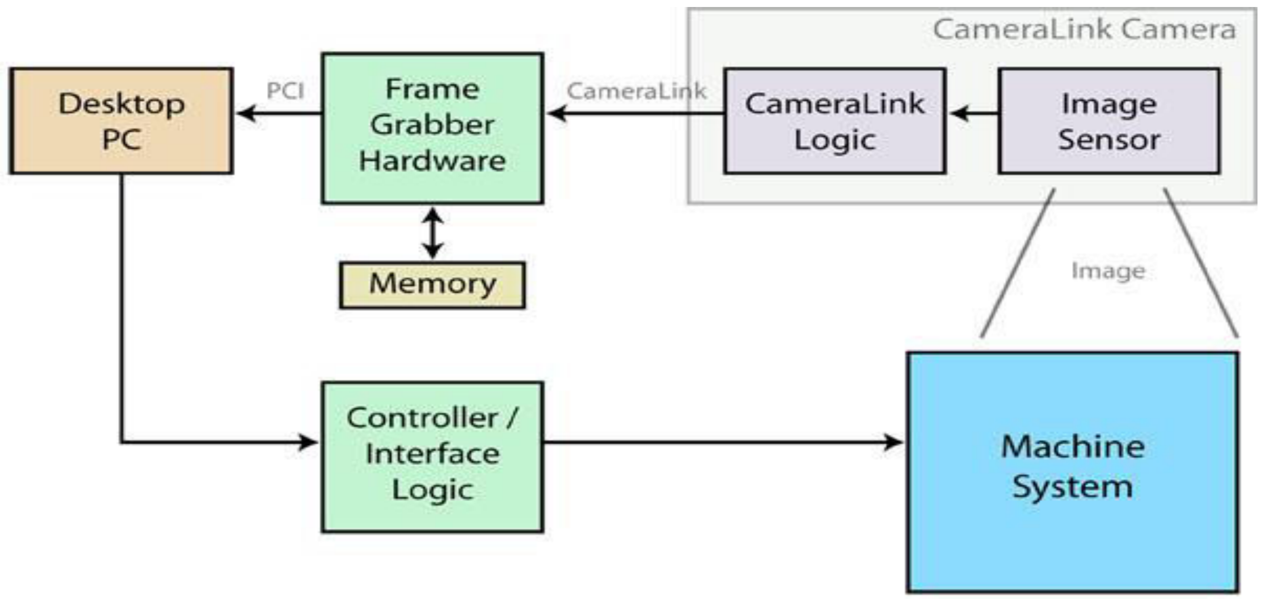


Fig 2.1 – Block Diagram of Video Acquisition System

2.3 DIFFERENT PARTS OF VIDEO ACQUISITION SYSTEM

2.3.1 CAMERA LENS

Camera lens (also known as photographic lens or photographic objective) is an optical lens or assembly of lenses used in conjunction with a camera body and mechanism to make images of objects either on photographic film or on other media capable of storing an image chemically or electronically.

2.3.2 IMAGE SENSOR

The image sensor is basically a micro-chip with a width of about 10mm. The chip consists of array of sensors, which can convert the light into electrical charges. Though both CMOS and CCD are very common, CMOS chips are known to be cheaper. But for higher pixel range and costly cameras mostly CCD technology is used.

An image sensor or imaging sensor is a sensor that detects and conveys the information that constitutes an image. It does so by converting the variable attenuation of waves (as they pass through or reflect off objects) into signals, the small bursts of current that convey the information. The waves can be light or other electromagnetic radiation. Image sensors are used in electronic imaging devices of both analog and digital types, which include digital cameras, camera modules, medical imaging equipment, night vision equipment such as thermal imaging devices, radar, sonar, and others. As technology changes, digital imaging tends to replace analog imaging.

Early analog sensors for visible light were video camera tubes. Currently, used types are semiconductor charge-coupled devices (CCD) or active pixel sensors in complementary metal–oxide–semiconductor (CMOS) or N-type metal-oxide-semiconductor (NMOS, Live MOS) technologies. Analog sensors for invisible radiation tend to involve vacuum tubes of various kinds. Digital sensors include flat panel detectors.

CCD VS CMOS SENSORS:

Today, most digital still cameras use a CMOS sensor because CMOS sensor technology in recent years has leapfrogged CCDs. CCD is still in use for cheap low entry cameras, but weak in burst mode. Both types of sensor accomplish the same task of capturing light and converting it into electrical signals.

Each cell of a CCD image sensor is an analog device. When light strikes the chip it is held as a small electrical charge in each photo sensor. The charges are converted to voltage one pixel at a time as they are read from the chip. Additional circuitry in the camera converts the voltage into digital information.

A CMOS imaging chip is a type of active pixel sensor made using the CMOS semiconductor process. Extra circuitry next to each photo sensor converts the light energy to a voltage. Additional circuitry on the chip may be included to convert the voltage to digital data.

Neither technology has a clear advantage in image quality On one hand, CCD sensors are more susceptible to vertical smear from bright light sources when the sensor is overloaded; high-end CMOS sensors in turn do not suffer from this problem. On the other hand, cheaper CMOS sensors are susceptible to undesired effects that come as a result of rolling shutter.

CMOS sensors can potentially be implemented with fewer components, use less power, and/or provide faster readout than CCD sensors. CCD is a more mature technology and is in most respects the equal of CMOS. CMOS sensors are less expensive to manufacture than CCD sensors.

Another hybrid CCD/CMOS architecture, sold under the name "CMOS," consists of CMOS readout integrated circuits (ROICs) that are bump bonded to a CCD imaging substrate – a technology that was developed for infrared staring arrays and now adapted to silicon-based detector technology. Another approach is to utilize the very fine dimensions available in modern CMOS technology to implement a CCD like structure entirely in CMOS technology. This can be achieved by separating individual poly-silicon gates by a very small gap. These hybrid sensors are still in the research phase and can potentially harness the benefits of both CCD and CMOS imagers.

The newer sensor technology is Back-side illuminated CMOS (BSI-CMOS) which uses less electricity than traditional CMOS with better performance than CCD, so lower-end cameras still use CCD sensors such as those implemented by Fujifilm in its Bridge cameras. CCD sensors are rarely used in new models, except for very high pixel count, big sensor cameras which still use CCDs.

The principal advantages of CMOS over the CCD for space instrumentation are compactness, low mass, low power and radiation hardness. The CCD remains unchallenged in dynamic range and photometric accuracy. The effects of radiation in CMOS and CCD sensors are similar in that both suffer from ionizing radiation and displacement damage. However, the key advantage of the CMOS APS is that there is no degradation of CTE.

APPLICATIONS OF CMOS:

1. **Security**

Surveillance systems enable a human operator to remotely monitor activity over large areas. Such systems are usually equipped with a number of video cameras, communication devices and computer software or some kind of DSP for real-time video analysis. Such analysis can include scene understanding, attention based alarming, colour analysis, tracking, motion detection, windows of interest extraction etc. With recent progress in CMOS image sensor technology and embedded processing, some of the mentioned functions and many others can be implemented in dedicated hardware, minimizing system cost and power consumption

2. **Medical applications**

Artificial vision is an example of CMOS image sensors implementation in medical applications. Almost all medical and near medical areas benefit from these image sensors utilization. These sensors are used for patients' observation and drug production, inside the dentist offices and during surgeries.

3. Quantum efficiency

The quest for a scientific CMOS sensor is today still in its infancy. The quantum efficiency of front-illuminated CMOS sensors is compromised by the in-pixel electronics and aluminum bus tracks reducing the “fill factor”, a measure of the fraction of the pixel’s area that is actually sensitive to light. The losses arise from the reflection of light from the aluminum bus lines and photon-induced electrons in the substrate being absorbed and lost within the in-pixel transistor electronics. An obvious solution to the problem is to thin and back-illuminate the sensor and several research groups have now demonstrated back-illuminated CMOS sensors that achieve quantum efficiencies in line with their CCD counterparts.

4. Readout noise and dynamic range

A second area of research is to minimize readout noise and maximize charge storage capacity and linearity. A limitation is the linear voltage swing that can be obtained outside the transistor threshold regions in modern low-voltage CMOS processes. The linear dynamic range of today’s best CMOS sensors is $\approx$ 5000, considerably less than a CCD. Several approaches to overcoming the problem are being investigated including the concept of pixels that deliberately behave in a nonlinear fashion and sensors that allow individual pixels to have varying exposure periods. More complex five- and six-transistor pixels are also being investigated in the pursuit of increased dynamic range. We can anticipate further progress in the future as researchers adapt to exploiting the advantages of CMOS technology rather than attempting to emulate the elegance of the CCD.

5. CMOS in space

CMOS sensors are already used in space, having applications in satellite bus instrumentation such as star trackers and inspection cameras. CMOS is yet to have a significant impact in scientific payloads for which the CCD remains dominant.

APPLICATIONS OF CCD:

1. In astronomy and space

The CCD sensors are becoming the integral components in astronomy and satellite mapping. The CCD sensors are used in Hubble Space Technology (HST) because of its high performance.

2. In Computers

The CCD’s are used to read the barcodes which consist of parallel vertical lines or bars used to assign a unique identification to an item. The CCD’s in this application read the data at a rate ranging from 100 to 250fps.

3. **In Image Processing**

CCDs are used as contact image sensors for data acquisition and optical character recognition. CCDs are used for data compression and signal integration analysis.

2.3.3 FRAME GRABBER

Frame grabber is an electronic device that captures (i.e., "grabs") individual, digital still frames from an analog video signal or a digital video stream. It is usually employed as a component of a computer vision system, in which video frames are captured in digital form and then displayed, stored or transmitted. Early frame grabbers typically had only enough memory to store a single digitized video frame, hence the singular "frame" in the device name, whereas modern frame grabbers often can store multiple frames.

Historically, frame grabber expansion cards were the predominant way to interface cameras to PC's. Other interface methods have emerged since then, with frame grabbers (and in many case, cameras themselves) connecting to computers via interfaces such as USB, Ethernet and IEEE 1394 ("FireWire").

Modern frame grabber devices often perform functions beyond capturing a single video input. For example, some devices can capture audio in addition to video, and some provide multiple video inputs that are captured concurrently. Other operations may be performed as well, such as de interlacing, text or graphics overlay, image transformations (e.g., resizing, rotation, mirroring), and real time compression using algorithms such as MPEG2 and JPEG. Also, technological demands in fields such as radar acquisition, manufacturing and remote guidance have led to the development of frame grabbers that can capture images at high frame rates and resolutions.

In our project we use a very high speed CCD camera to record the video. This is given to PC using a frame grabber. The recording speed of video is very fast and when given to PC it runs at the speed of processor in the PC, thus a frame grabber is used to adjust the frames per second in our PCs.

2.4 TYPES OF CAMERA

1. Ordinary camera
2. High speed camera

2.4.1 Ordinary Camera:

An Ordinary digital camera has lens/lenses which are used to focus the light that is to be projected and created. This light is made to focus on an image sensor which converts the light signals into electric signals. The light hits the image sensor as soon as the photographer hits the shutter button. As soon as the shutter opens the pixels are illuminated by the light in different intensities. Thus an electric signal is generated. This electric signal is then further broke down to digital data and stored in a computer.

2.4.2 High Speed Camera:

A high-speed camera is a device capable of image exposures in excess of 1/1,000 or frame rates in excess of 250 frames per second. It is used for recording fast-moving objects as a photographic image(s) onto a storage medium. After recording, the images stored on the medium can be played back in slow motion. Early high-speed cameras used film to record the high-speed events, but today high-speed cameras are entirely electronic using either a charge coupled device (CCD) or a CMOS active pixel sensor, recording typically over 1,000 frames per second into DRAM and playing images back slowly to study the motion for scientific study of transient phenomena.

A high-speed camera can be classified as:

- A high-speed film camera which records to film.
- A high-speed video camera which records to electronic memory.
- A high-speed framing camera which records images on multiple image planes or multiple locations on the same image plane (generally film or a network of CCD cameras),
- A high-speed streak camera records a series of line-sized images to film or electronic memory.

A normal motion picture film is played back at 24 frames per second, while television uses 25 frames/s (PAL) or 29.97 frames/s (NTSC). High-speed film cameras can film up to a quarter of a million frames per second by running the film over a rotating prism or mirror instead of using a shutter, thus reducing the need for stopping and starting the film behind a shutter which would tear the film stock at such speeds. Using this technique one can stretch one second to more than ten minutes of playback time (super slow motion). High-speed video cameras are widely used for scientific research, military test and evaluation, and industry.

Examples of industrial applications are

- Filming a manufacturing line to better tune the machine,
- In the car industry the crash testing to better document the crash and what happens to the automobile and passengers during a crash.

Today, the digital high-speed camera has replaced the film camera used for Vehicle Impact Testing.

2.5 PARAMETERS OF CAMERA:

Frame rate, also known as frame frequency, is the frequency (rate) at which an imaging device displays consecutive images called frames. The term applies equally to film and video cameras, computer graphics, and motion capture systems. Frame rate is expressed in frames per second (FPS).

Resolution is the capability of the sensor to observe or measure the smallest object clearly with distinct boundaries. There is a difference between the resolution and a pixel. A pixel is actually a unit of the digital image. Resolution depends upon the size of the pixel. Usually, with any given lens setting, the smaller the size of the pixel, the higher the resolution will be and the clearer the object in the image will be. Images having smaller pixel sizes might consist of more pixels. The number of pixels correlates to the amount of information within the image.

An image of N pixels height by M pixels wide can have any resolution less than N lines per picture height, or N TV lines. But when the pixel counts are referred to as resolution, the convention is to describe the pixel resolution with the set of two positive integer numbers, where the first number is the number of pixel columns (width) and the second is the number of pixel rows (height), for example as 7680 by 6876.

Another popular convention is to cite resolution as the total number of pixels in the image, typically given as number of megapixels, which can be calculated by multiplying pixel columns by pixel rows and dividing by one million. Other conventions include describing pixels per length unit or pixels per area unit, such as pixels per inch or per square inch.

CHAPTER 3

FUNDAMENTALS OF DIGITAL IMAGE PROCESSING

3.1 INTRODUCTION

The field of digital image processing refers to processing digital images by means of a digital computer. A digital image is composed of a finite number of elements, each of which has a particular location and these elements are referred to as picture elements, image elements, pels and pixels. Pixel is the term most widely used to denote the elements of a digital image.

Thus digital image processing refers to the processes of acquiring an image of the area containing the text, preprocessing that image, extracting (segmenting) the individual characters, describing the characters in a form suitable for computer processing, and recognizing those individual characters.

There are no clear-cut boundaries in the continuum from image processing at one end to computer vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, and high-level processes.

Low-level processes involve primitive operations such as image preprocessing to reduce noise, contrast enhancement, and image sharpening. A low-level process is characterized by the fact that both its inputs and outputs are images.

Mid-level processing on images involves tasks such as segmentation (partitioning an image into regions or objects), description of those objects to reduce them to a form suitable for computer processing, and classification (recognition) of individual objects. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images (e.g., edges, contours, and the identity of individual objects).

Finally, higher-level processing involves “making sense” of an ensemble of recognized objects, as in image analysis, and, at the far end of the continuum, performing the cognitive functions normally associated with vision.

Image:

An image refers to a 2D light intensity function f(x,y), where (x,y) denote spatial coordinates and the value of f at any point (x,y) is proportional to the brightness or gray levels of the image at that point.

Digital Image:

A digital image is an image f(x,y) that has been discretized both in spatial coordinates and brightness. The elements of such a digital array are called image elements or pixels.

An image may be defined as a two-dimensional function, f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image.

Representation of digital images:

As the CCD sensor acquires the images, the main objective is to generate digital images from the sensed data. The output of most sensors is a continuous voltage waveform whose amplitude and spatial behavior are related to the physical phenomenon being sensed. To create a digital image, we need to convert the continuously sensed data to digital form. This involves two processes: Sampling and Quantization.

If f(x,y) is a continuous image with respect to x-coordinate and y-coordinate and also in amplitude. To convert it to digital form we have to sample the function in both coordinates and in amplitude. Digitizing the amplitude value is called Quantization.

The result of sampling and quantization is a matrix of real numbers. If a 2D continuous image f(x,y) is sampled so that the resulting digital image has M rows and N columns of (x,y), it now becomes the discrete quantity. For notational convenience, integer values are used for these discrete quantities. The intersection of a row and a column is termed as pixel. The values assigned to integer coordinates [m,n] with {m=0,1,2,3,4,….M-1} and {n=0,1,,2,3,…N-1}.

The digitization process requires decisions about values for M,N and for the number L, of discrete grey levels allowed for each pixel. There are no requirements on M,N other than they have to be positive numbers. However due to processing, storage and sampling hardware considerations, the number of gray levels typically is an integer power of 2 i.e., L=2 power of k. Assume that discrete levels are equally spaced and that they are integers in the interval [0,L-1]. Sometimes the gray scale is called the dynamic range of an image.

The values assigned to every pixel rounded to the nearest integer values. Images whose gray levels span a significant portion of the gray scale have a high dynamic range. When an appreciable number of pixels exhibit this property, the image will have contrast. Conversely, an image with low dynamic range tends to have a low contrast. The number of bits required to store digitized image is b=M*N*K. When an image can have 2 powers of K gray levels, it is referred as “K-bit image”.

3.2 NEED FOR PROCESSING

Images are produced by a variety of physical devices, including still and video cameras, X-ray devices, electron microscope, radar and ultra sound and used for variety of purposes, including entertainment, medical, business, industrial, military, civil, security and scientific fields. The goal in each case for observer, human or machine is to extract useful information about the scene being imaged. Often the raw image is not directly suitable for this purpose and must be processed in some way. Such processing is called image enhancement; processing by an observer to extract information is called image analysis. Enhancement and analysis are distinguished by their output, image vs. scene information and the challenges faced and methods employed. Images enhancement has been done by chemical, optical and electronics means, while analysis has been done mostly by humans.

3.3 FUNDAMENTAL STEPS IN IMAGE PROCESSING

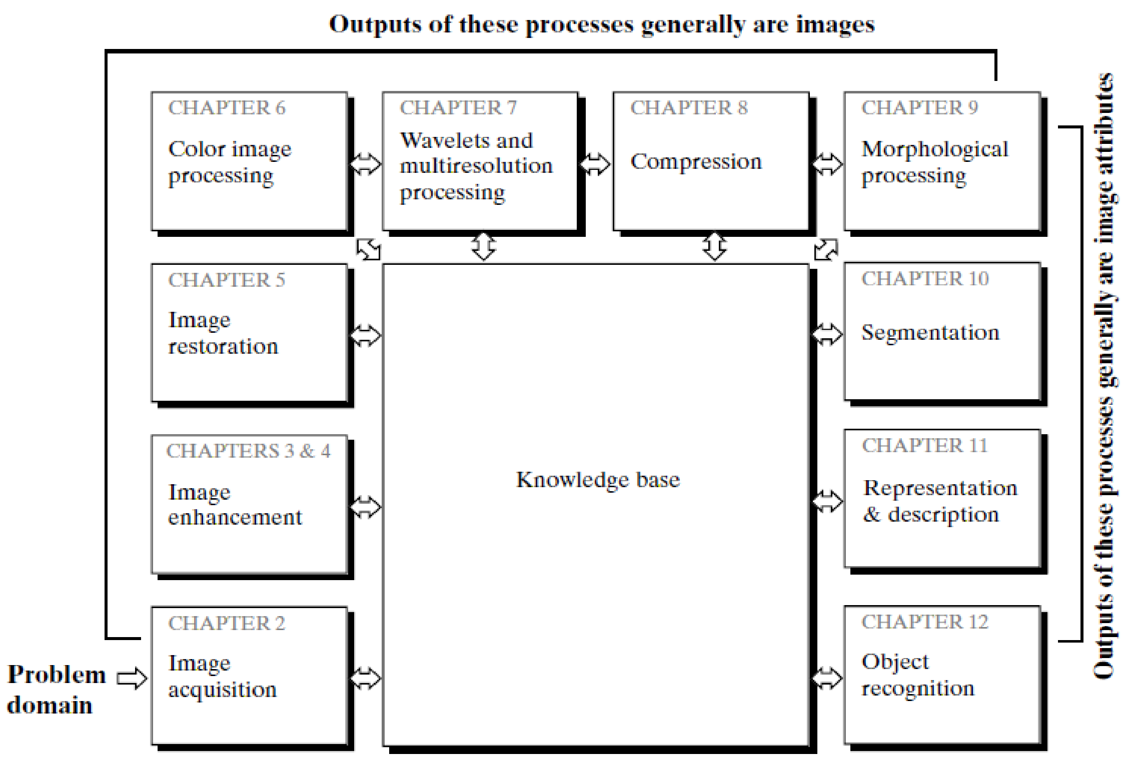


Fig 3.1- Different Steps in Image Processing

Image acquisition is the first process as shown in figure. Acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves preprocessing, such as scaling.

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. A familiar example of enhancement is when we increase the contrast of an image because "it looks better." It is important to keep in mind that enhancement is a very subjective area of image processing.

Image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Color image processing is an area that has been gaining in importance because of the significant increase in the use of digital images over the Internet.

Wavelets are the foundation for representing images in various degrees of resolution.

Compression, as the name implies, deals with techniques for reducing the storage required to save an image, or the bandwidth required to transmit it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar (perhaps inadvertently) to most users of computers in the form of image file extensions, such as the jpg file extension used in the JPEG (Joint Photographic Experts Group) image compression standard.

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape.

Segmentation procedures partition an image into its constituent parts or objects. The more accurate the segmentation, the more likely recognition is to succeed.

Representation and Description: Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another. Recognition is the process that assigns a label (e.g., "vehicle") to an object based on its descriptors.

Knowledge about a problem domain is coded into an image processing system in the form of a **knowledge database**. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules.

3.4 DIGITAL IMAGE PROCESSING SYSTEM:

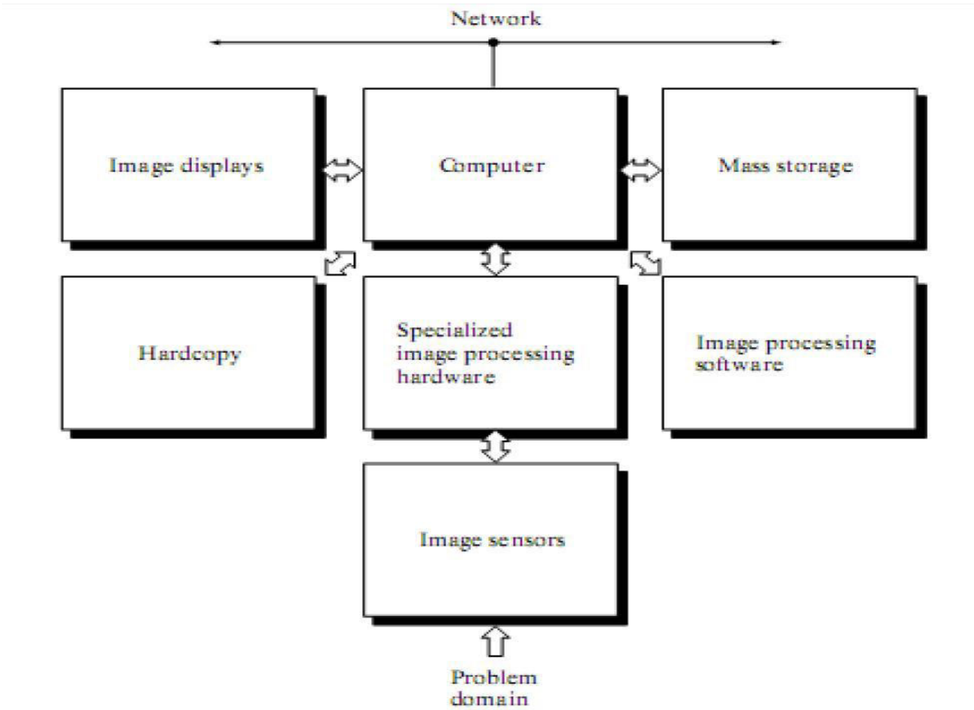


Fig 3.2- Image Processing System

Image Sensors

With reference to sensing, two elements are required to acquire digital images. The first is a physical device that is sensitive to the energy radiated by the object we wish to image. The second, called a digitizer, is a device for converting the output of the physical sensing device into digital form.

Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), which performs arithmetic and logical operations in parallel on entire images. This type of hardware sometimes is called a front-end subsystem, and its most distinguishing characteristic is speed.

The **computer** in an image processing system is a general-purpose computer and can range from a PC to a supercomputer. In dedicated applications, sometimes specially designed computers are used to achieve a required level of performance.

Software for image processing consists of specialized modules that perform specific tasks. A well-designed package also includes the capability for the user to write code that, as a minimum, utilizes the specialized modules.

Mass storage capability is a must in image processing applications. Digital storage for image processing applications falls into three principal categories: (1) short term storage for use during processing, (2) on-line storage for relatively fast recall, and (3) archival storage, characterized by infrequent access.

One method of providing short-term storage is computer memory. Another is by specialized boards, called frame buffers that store one or more images and can be accessed rapidly, usually at video rates. Online storage generally takes the form of magnetic disks or optical-media storage. Finally, archival storage is characterized by massive storage requirements but infrequent need for access. Magnetic tapes and optical disks housed in "jukeboxes" are the usual media for archival applications.

Image displays in use today are mainly color (preferably flat screen) TV monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system.

Hardcopy devices for recording images include laser printers, film cameras, heat-sensitive devices, inkjet units, and digital units, such as optical and CD-ROM disks.

3.5 APPLICATIONS OF IMAGE PROCESSING:

Digital image processing and analysis techniques are used today in variety of problems.

The following are a few major application areas:

Military applications: Missile guidance and detection, target identification, navigation of pilot's vehicle and range finding etc.

Criminology: Finger print identification, human face identification and matching.

Information Technology: Facsimile image transmission, video text, video conferencing and videophones etc.

Scientific application: High energy physics, bubble chamber and other forms of tracking analysis etc.

Astronomy and Space application: Restoration of images suffering from geometric and photometric distortions, computing close-up picture of planter surface etc.

Bio-Medical: ECG, EEG, EMG analysis, cytological, histological and stereological applications, automated radiology and pathology-ray image analysis, mass screening.

CHAPTER 4

TRACKING METHODS AND OUR PROPOSED METHOD

4.1 STEPS IN TRACKING

The basic steps involved in tracking are as follows:

Object Detection:

Object Detection is to identify objects of interest in the video sequence and to cluster pixels of these objects. Object detection can be done by various techniques such as frame differencing, Optical flow and Background subtraction.

Object Classification:

Object can be classified as vehicles, birds, floating clouds, swaying tree and other moving objects. The approaches to classify the objects are Shape-based classification, Motion-based classification, Color based classification and texture based classification.

Object Tracking:

Tracking can be defined as the problem of approximating the path of an object in the image plane as it moves around a scene. The approaches to track the objects are point tracking, kernel tracking and silhouette.

Following are some of the challenges that should be taken care in object tracking:

1. Loss of evidence caused by estimate of the 3D realm on a 2D image,

2. Noise in an image,

3. Difficult object motion,

4. Imperfect and entire object occlusions,

5. Complex objects structures.

4.2 DIFFERENT METHODS

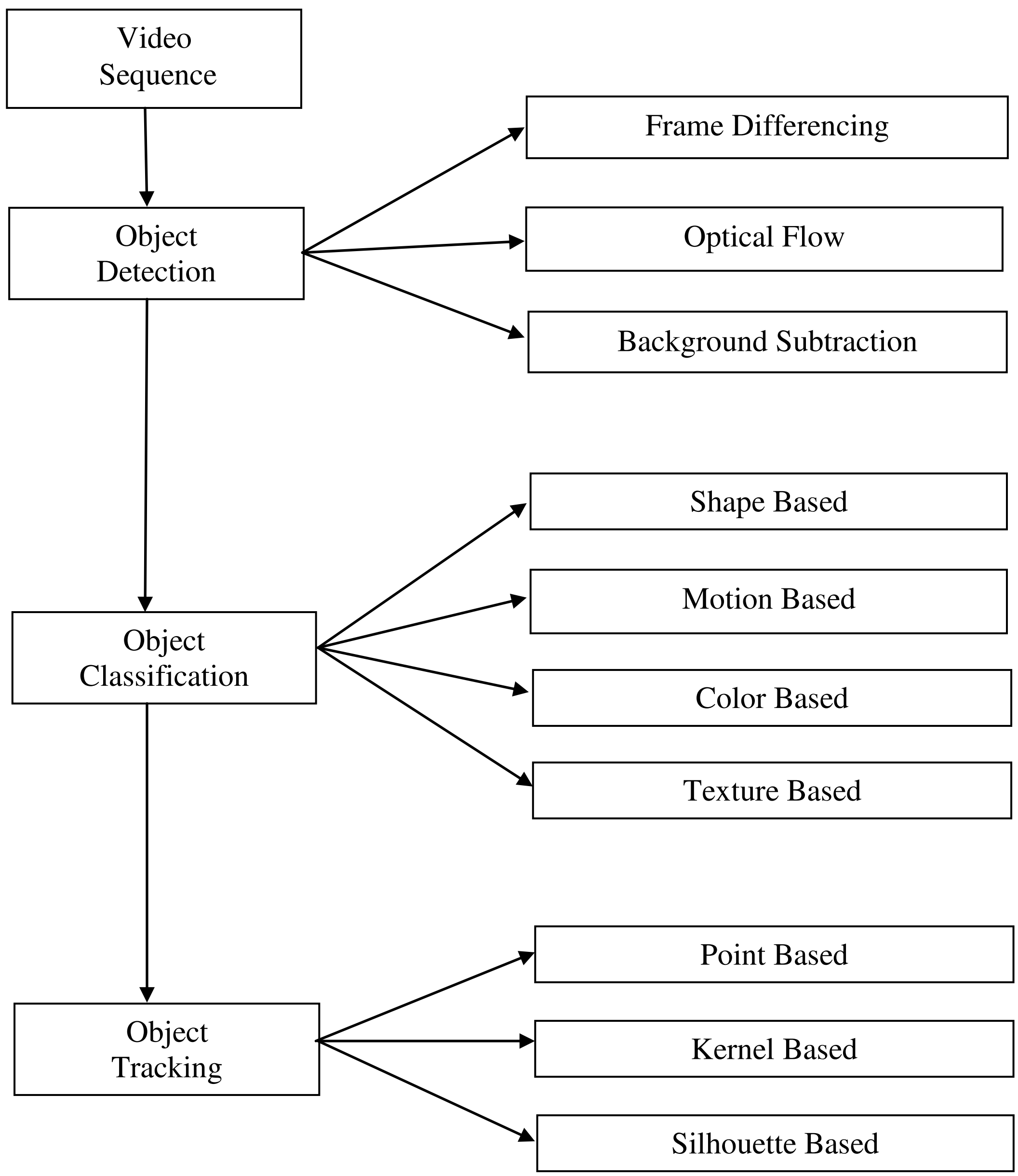


Fig 4.1- Different Steps and Methods in Tracking

OBJECT DETECTION METHODS:

A. Frame Differencing:

The presence of moving objects is determined by calculating the difference between two consecutive images. Its calculation is simple and easy to implement. For a variety of dynamic environments, it has a strong adaptability, but it is generally difficult to obtain complete outline of moving object, responsible to appear the empty phenomenon, as a result the detection of moving object is not accurate.

B. Optical Flow:

Optical flow method is to calculate the image optical flow field, and do clustering processing according to the optical flow distribution characteristics of image. This method can get the complete movement information and detect the moving object from the background better, however, a large quantity of calculation, sensitivity to noise, poor anti noise performance, make it not suitable for real-time demanding occasions.

C. Background subtraction:

First step for background subtraction is background modeling. It is the core of background subtraction algorithm. Background Modeling must sensitive enough to recognize moving objects. Background Modeling is to yield reference model. This reference model is used in background subtraction in which each video sequence is compared against the reference model to determine possible Variation. The variations between current video frames to that of the reference frame in terms of pixels signify existence of moving objects. The background subtraction method is to use the difference method of the current image and background image to detect moving objects, with simple algorithm, but very sensitive to the changes in the external environment and has poor anti-interference ability. However, it can provide the most complete object information in the case background is known.

OBJECT CLASSIFICATION METHODS:

A. Shape-based classification:

Different descriptions of shape information of motion regions such as representations of points, box and blob are available for classifying moving objects. Input features to the network is mixture of image-based and scene-based object parameters such as image blob area, apparent aspect ratio of blob bounding box and camera zoom. Classification is performed on each blob at every frame and results are kept in histogram.

B. Motion-based classification:

Non-rigid articulated object motion shows a periodic property, so this has been used as a strong cue for moving object classification. Optical flow is also very useful for object classification. Residual flow can be used to analyze rigidity and periodicity of moving entities. It is expected that rigid objects would present little residual flow where as a non rigid moving object such as human being had higher average residual flow and even displayed a periodic component.

C. Color-based classification:

Unlike many other image features (e.g. shape) color is relatively constant under viewpoint changes and it is easy to be acquired. Although color is not always appropriate as the sole means of detecting and tracking objects, but the low computational cost of the algorithms proposed makes color a desirable feature to exploit when appropriate. To detect and track vehicles or pedestrians in real-time color histogram based technique is used. According to a Gaussian Mixture Model is created to describe the color distribution within the sequence of images and to segment the image into background and objects.

D. Texture-based classification:

Texture based technique counts the occurrences of gradient orientation in localized portions of an image, is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

OBJECT TRACKING METHODS:

A. Point Tracking:

In an image structure, moving objects are represented by their feature points during tracking. Point tracking is a complex problem particularly in the incidence of occlusions, false detections of object. Recognition can be done relatively simple, by thresholding, at of identification of these points.

B. Kernel Based Tracking:

Kernel tracking is usually performed by computing the moving object, which is represented by a embryonic object region, from one frame to the next. The object motion is usually in the form of parametric motion such as translation, conformal, affine, etc. These algorithms diverge in terms of the presence representation used, the number of objects tracked, and the method used for approximation the object motion. In real-time, illustration of object using geometric shape is common. But one of the restrictions is that parts of the objects may be left outside of the defined shape while portions of the background may exist inside. This can be detected in rigid and non-rigid objects.

C. Silhouette Based Tracking Approach:

Some object will have complex shape such as hand, fingers, shoulders that cannot be well defined by simple geometric shapes. Silhouette based methods afford an accurate shape description for the objects. The aim of a silhouette-based object tracking is to find the object region in every frame by means of an object model generated by the previous frames. Capable of dealing with variety of object shapes, Occlusion and object split and merge.

4.3 OUR PROPOSED METHOD

We have chosen to implement the simple object tracking algorithm using background subtraction method with shape based classification and blob analysis techniques. Our algorithm is a modified method for traditional background subtraction method by considering the sudden illumination changes in the background. Our modifications include considering the background model to have the average of all intensity values of all the frames in the given video and also by performing morphological operations on the difference image obtained.

4.4 FLOWCHART

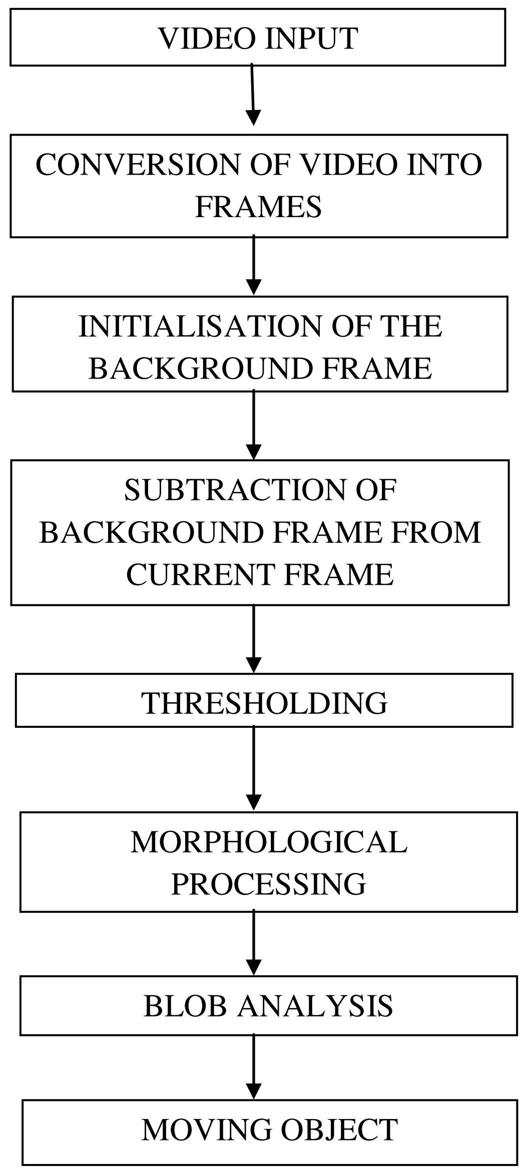


Fig 4.2 Flow Chart of the Project

4.5 DESCRIPTION OF FLOWCHART:

Step-1: Video Input

Videos are image sequences over the time. The camera will continuously capture the images. In this we take a video of any format as input to our system which has already been recorded at some frames per second by a high speed camera. This video input is provided for next step of processing.

Step-2: Conversion of Video into Frames

Directly processing on the video is not easy. So the conversion of the video into frames is done, i.e., video is now cut into different number of frames based on the speed of the processor in the computer system and the speed at which the video is recorded. This will give the frames at different time period which are used for further processing.

Step-3: Initialization of the background frame

In this step the background frame or the reference frame is initialized. The first frame can be itself considered itself as the background frame directly, but this is the traditional method and not suitable for sudden illumination changes in the background. So a new frame is formed by taking the mean of each corresponding pixels in all the frames and that frame is considered as the background frame. This method can encounter the tracking defects which may be due to sudden illumination changes in the background, as it considers the mean value of all the intensity values in all the frames while calculating the background intensity values.

Step-4: Subtraction of Background frame from current frame

In this step all the frames of the video are subtracted from the chosen background frame in a sequential order and the corresponding difference image frames are formed. The subtraction will be done in a pixel by pixel fashion between the current and the reference (background) frames.

Step-5: Thresholding

In this step a threshold value is chosen and applying that threshold value to the difference images formed, segmentation of all pixels in all frames is done. Each frame now has only two kind of pixels one foreground and the other background pixels. Foreground pixels are those which actually correspond to the moving objects where as background pixel refers to static objects and static backgrounds.

Step-6: Morphological Processing

Morphological processing or filtering refers to removing of small regions probably created by noise and camera jitter. It also fills up unnecessary cavities, smoothens boundaries, extract edges**.** The fundamental morphological operations are erosion and dilation.

Erosion: To erode an image, each object pixel that is touching a background pixel is changed into a back-ground pixel, erosion removes isolated background pixel. The definition of erosion is

f(x, y) = min{ I(x,y) and its neighboring pixels }

where f(x,y) is the eroded image at pixel (x,y), and I(x,y) is the pixel intensity. 8-point neighborhood is used to define neighboring pixels.

Dilation: Dilation is the dual of erosion. Each background pixel that is touching an object pixel is changed into an object pixel when it is dilated. Dilation adds pixels to the boundary of the object and closes isolated background pixel (fills up the holes of the object). The definition of dilation is

g(x, y) = max{ I(x,y) and its neighboring pixels }

where g(x,y) is the dilated image at pixel (x,y).

The other two functions are:

Opening: Opening is erosion followed by a dilation operation. Opening removes small islands and thin filaments of object pixels. It is used to remove fine object, separate a target from fine points, and smooth a boundary without changing the area or shape. The definition of opening is

Opening(M) = Dilation(Erosion(M))

Closing: Closing is a dilation operation followed by an erosion operation. Closing removes islands and thin filaments of background pixels. It fills small holes within objects, and joins the neighboring objects without changing the area or shape.

Closing(M) = Erosion(Dilation(M))

These techniques are useful for processing noisy images where some pixels have the wrong classification results. Successive operations of opening and closing are used to improve these defects.

Step-7: Blob Analysis

Blob analysis is a block used in image processing toolbox to calculate statistics for labeled regions in a binary image. The block returns quantities such as centroid, bounding box, label matrix, blob count, area, connectivity, major axis, minor axis, orientation etc. Some of the quantities and their functions are described below.

Table 4.1- Gives different properties and their corresponding functions described by blob analysis block of image processing tool box.

PROPERTY	FUNCTION
Centroid	This outputs an *M*-by-2 matrix of [x y] centroid coordinates. The rows represent the coordinates of the centroid of each region, where *M* represents the number of blobs.
Bounding Box	This output an *M*-by-4 matrix of [x y width height] bounding boxes. The rows represent the coordinates of each bounding box, where *M* represents the number of blobs.
Connectivity	Define which pixels connect to each other. If we want to connect pixels located on the top, bottom, left, and right, select 4. If you want to connect pixels to the other pixels on the top, bottom, left, right, and diagonally, select 8.
Orientation	This outputs a vector that represents the angles between the major axes of the ellipses and the *x*-axis. The angle values are in radians and range between: -pi/2 and +pi/2.
Major Axis Length	It represents the lengths of major axes of ellipses.
Minor Axis Length	It represents the lengths of minor axes of ellipses.

Step-8: Moving Object

After performing morphological filtering and blob analysis a clear box around the moving objects is formed in each frame and thus the moving object is tracked using the above algorithm.

CHAPTER 5

INTRODUCTION TO MATLAB AND IMAGE PROCESSING TOOLBOX

5.1 MATLAB INTRODUCTION

MATLAB is a programming language developed by Math Works. It started out as a matrix programming language where linear algebra programming was simple. It can be run both under interactive sessions and as a batch job.

The name MATLAB stands for MATrix LABoratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research.

MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide. It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.

5.2 ADVANTAGES AND DISADVANTAGES OF MATLAB

Advantages:

- Ease of use
- Platform independence
- Predefined functions
- Plotting

Disadvantages:

- Can be slow
- Expensive

5.3 GETTING STARTED WITH MATLAB

Firstly after installing MATLAB software and when we open it we find many windows.

MATLAB includes a variety of different windows for displaying different types of information and performing specific tasks. Each window can generally be opened/closed, docked in the main window or popped out, and repositioned/resized depending on current needs/preferences. The Window menu helps us navigate between the currently open windows, while the Desktop menu lets you open new windows (which can also be done from the command window).

Command Window: The window where we type commands and non-graphic output is displayed. A '>>' prompt shows us the system is ready for input. The lower left hand corner of the main window also displays 'Ready' or 'Busy' when the system is waiting or calculating. Previous commands can be accessed using the up arrow to save typing and reduce errors. Typing a few characters restricts this function to commands beginning with those characters.

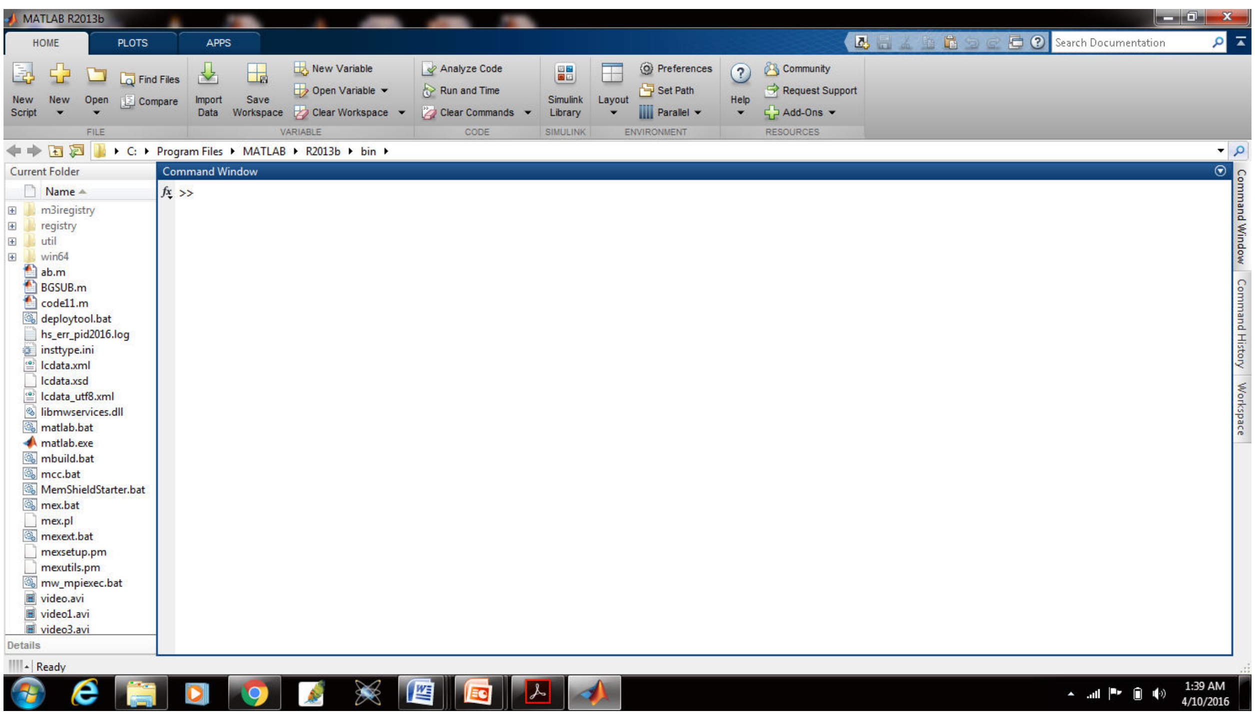


Fig 5.1 Command Window in MATLAB Tool

Command History: It records commands given that session and recent sessions. It can be used for reference or to copy and paste commands.

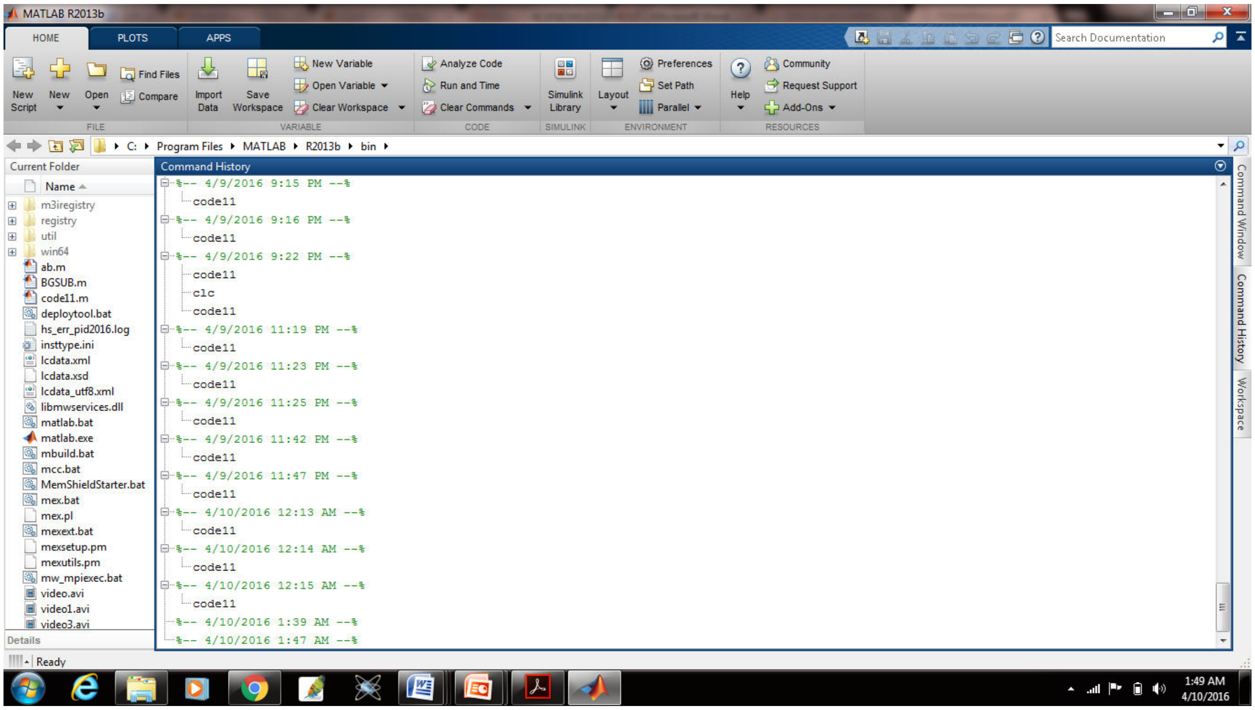


Fig 5.2- Command History Window in MATLAB

Workspace: It shows the all the variables that we have currently defined and some basic information about each one, including its dimensions, minimum, and maximum values. The icons at the top of the window allow us to perform various basic tasks on variables, creating, saving, deleting, plotting, etc. Double-clicking on a variable opens it in the Variable or Array Editor. All the variables that we've defined can be saved from one session to another using File>Save Workspace As (Ctrl-S). The extension for a workspace file is .mat.

Current Directory: The directory (folder) that MATLAB is currently working in. This is where anything we save will go by default, and it will also influence what files MATLAB can see. You won't be able to run a script that you saved that you saved in a different directory (unless you give the full directory path), but you can run one that's in a sub-directory. The Current Directory bar at the top centre of the main window lets you change directory in the usual fashion — you can also use the UNIX commands cd and pwd to navigate through directories. The Current Directory window shows a list of all the files in the current directory.

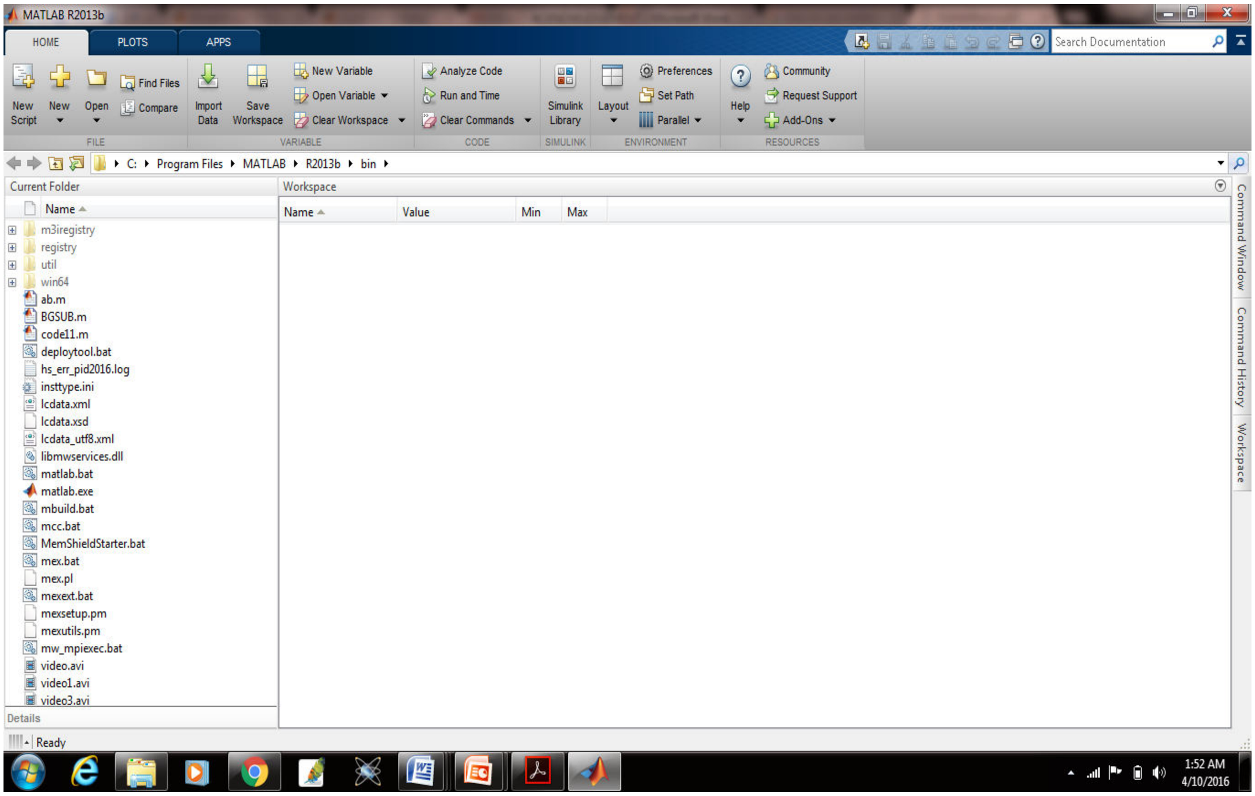


Fig 5.3- Workspace Window in MATLAB

Editor: The window where we edit m-files — the files that hold scripts and functions that we've defined or are editing — and includes most standard word-processing options and keyboard shortcuts. It can be opened by typing edit in the Command Window. Typing edit myfile will open myfile.m for editing. Multiple files are generally opened as tabs in the same editor window, but they can also be tiled for side by side comparison. Orange warnings and red errors appear as underlining and as bars in the margin. Hovering over them provides more information; clicking on the bar takes you to the relevant bit of text. Also remember that MATLAB runs the last saved version of a file, so you have to save before any changes take effect.

The editor window can be opened by opening NEW SCRIPT in the top of the MATLAB home page.

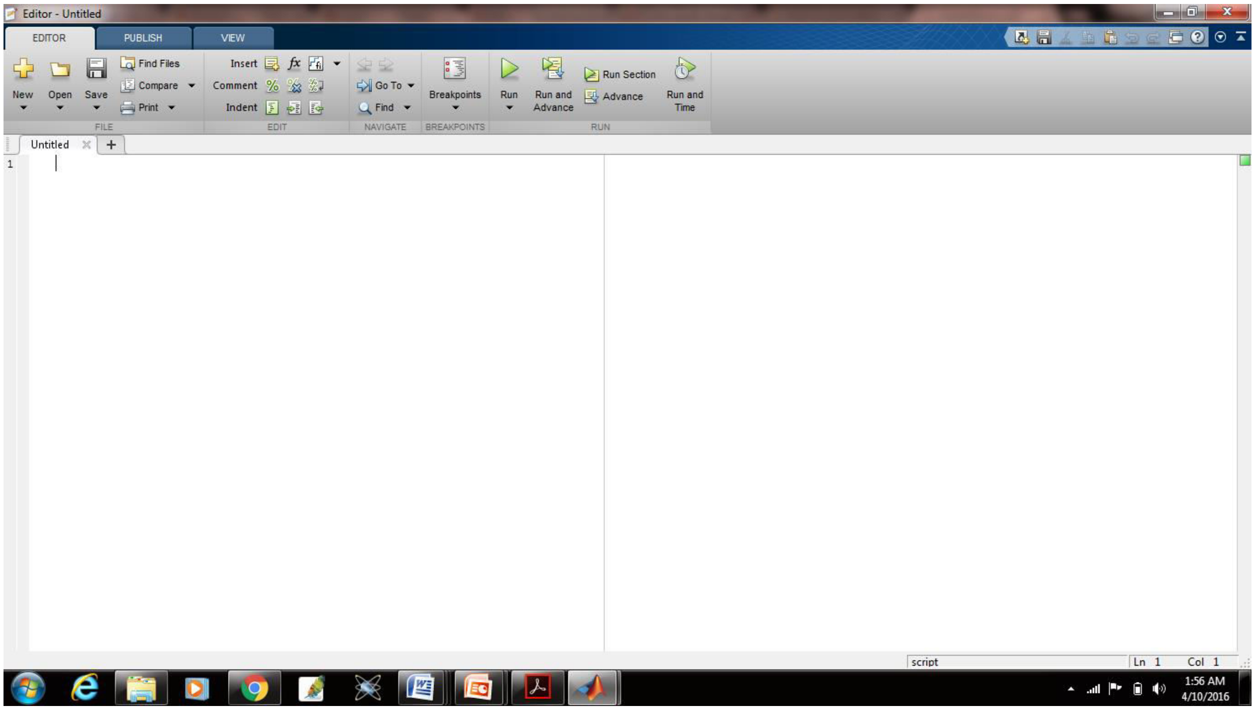


Fig 5.4- Editor Window in MATLAB

In the above scrip page we can write the codes and functions.

Help: Again in the home page by pressing the **question mark** on the top of the page, we can get aid of many tool boxes and different functions that can be used in MATLAB.

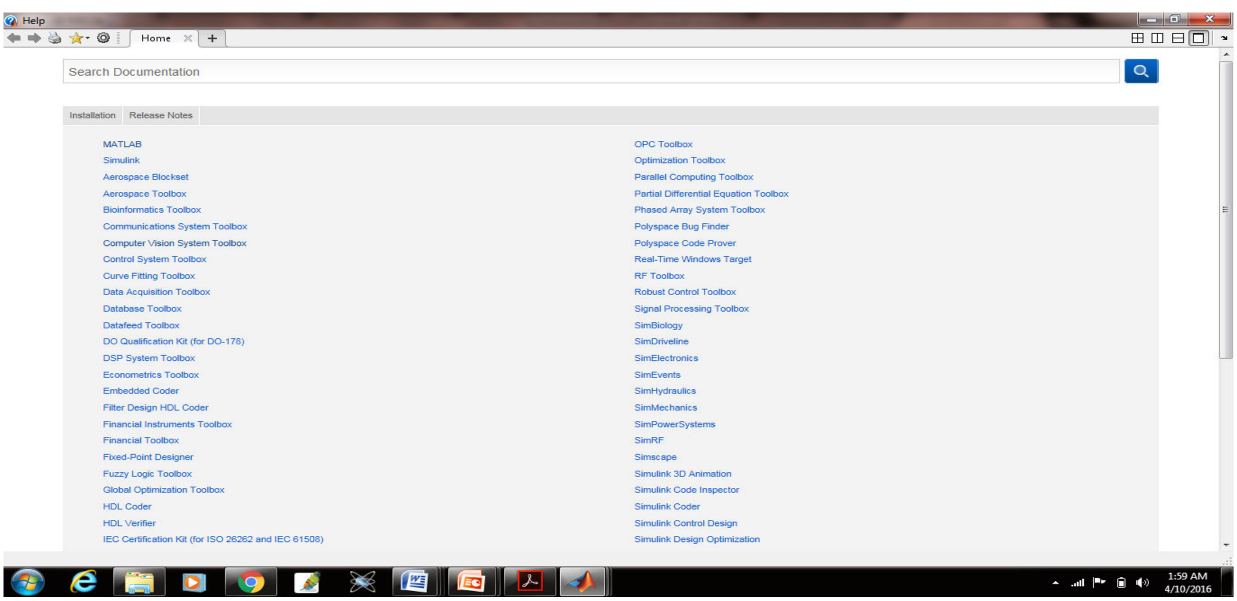


Fig 5.5- Help Window in MATLAB

For our project of Object Tracking, we need image processing tool box.

5.4 IMAGE PROCESSING TOOBOX

The Image Processing Toolbox is a collection of functions that extend the capability of the MATLAB® numeric computing environment. The toolbox supports a wide range of image processing operations, including:

- Spatial image transformations
- Morphological operations
- Neighborhood and block operations
- Linear filtering and filter design
- Transforms
- Image analysis and enhancement
- Image registration
- Deblurring
- Region of interest operations

Many of the toolbox functions are MATLAB M-files, a series of MATLAB statements that implement specialized image processing algorithms.

We can view the MATLAB code for these functions using the statement

type function_name

We can extend the capabilities of the Image Processing Toolbox by writing our own M-files, or by using the toolbox in combination with other toolboxes, such as the Signal Processing Toolbox and the Wavelet Toolbox.

BASIC FUNCTIONS IN IMAGE PROCESSING TOOLBOX:

Table 5.1- Basic Functions of Image Processing Toolbox

FUNCTION NAME	FUNCTION DESCRIPTION
mmreader	Create multimedia reader object for reading video files
immovie	Make movie from multiframe image
implay	Play movies, videos, or image sequences

imshow	Display image
im2bw	Convert image to binary image, based on threshold
im2double	Convert image to double precision
gray2ind	Convert grayscale or binary image to indexed image
rgb2gray	Convert RGB image or colormap to grayscale
imcontrast	Adjust Contrast tool
checkerboard	Create checkerboard image
imcrop	Crop image
imresize	Resize image
imrotate	Rotate image
bwboundaries	Trace region boundaries in binary image
bwtraceboundary	Trace object in binary image
entropy	Entropy of grayscale image
imcontour	Create contour plot of image data
regionprops	Measure properties of image regions

imhist	Display histogram of image data
imabsdiff	Absolute difference of two images
imadd	Add two images or add constant to image
imsubtract	Subtract one image from another or subtract constant from image
imcomplement	Complement image
imadjust	Adjust image intensity values or colormap
medfilt2	2-D median filtering
imfilter	N-D filtering of multidimensional images
imdilate	Dilate image
imerode	Erode image
imfill	Fill image regions and holes
bwareaopen	Morphologically open binary image (remove small objects)
bwconncomp	Find connected components in binary image

CHAPTER 6

RESULTS AND DISCUSSIONS

6.1 INTRODUCTION:

We followed the flowchart above and wrote the code using MATLAB functions and image processing Tool Box. The code wholly given in the appendix is written in the editor window of the MATLAB.

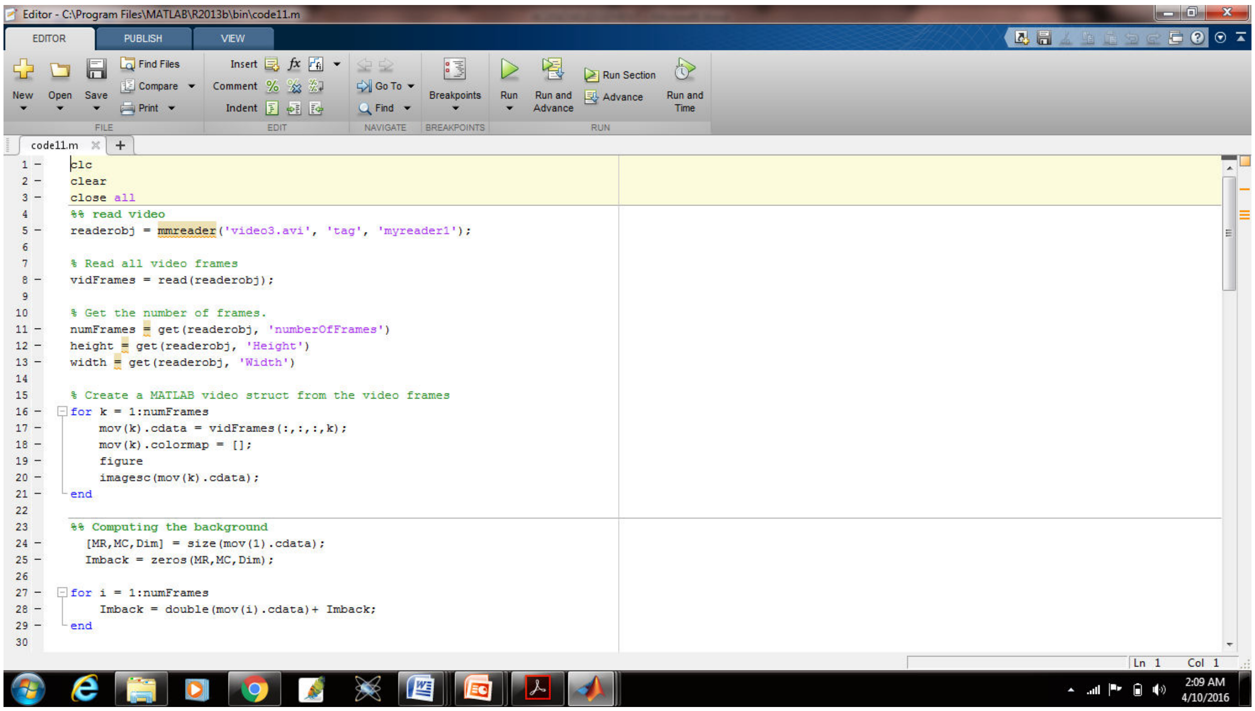


Fig 6.1- Code Written in editor window of MATLAB

6.2 DISCUSSIONS AND RESULTS

On running the code, we get the outputs at each step of the flowchart. The results are given below.

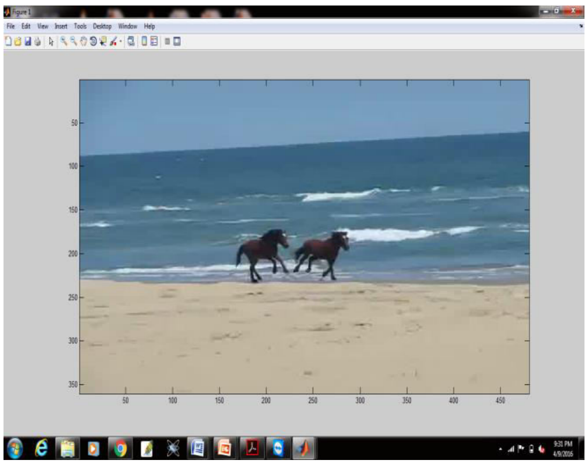
Step-1: Video Input

The video named video3 in '.avi' format is chosen as the input video which is added to the path of the MATLAB file. The video consists of horses running at the seashore. Our aim is to track the moving objects in that video. The video is given as input by a function called mmreader.

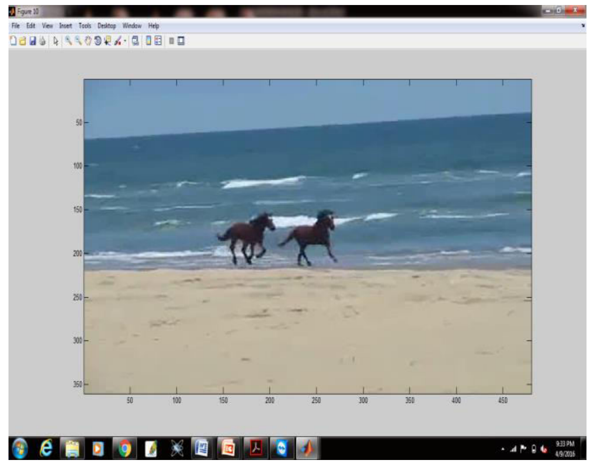
Step-2: Conversion of Video into Frames

All the video frames in the video are read through a function called 'read' and the number of frames, height and width of each frame is known as output in the command window. A video structure file is created from the frames read and all the frames are seen as output. Our video has been cut into 273 frames. Different frames from our chosen video3 are shown.

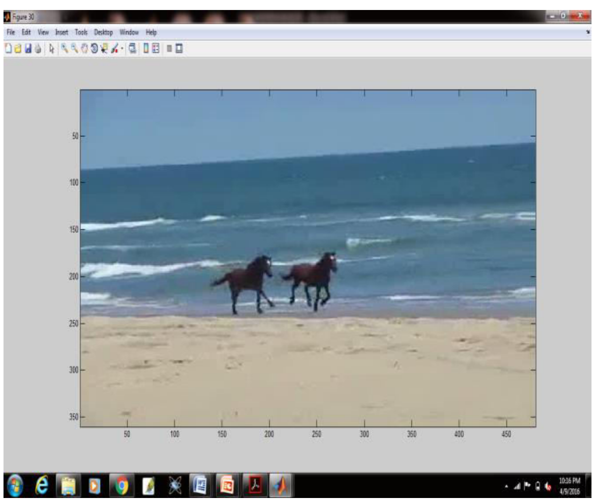
Fig Box 6.2- Results Of conversion of video into frames



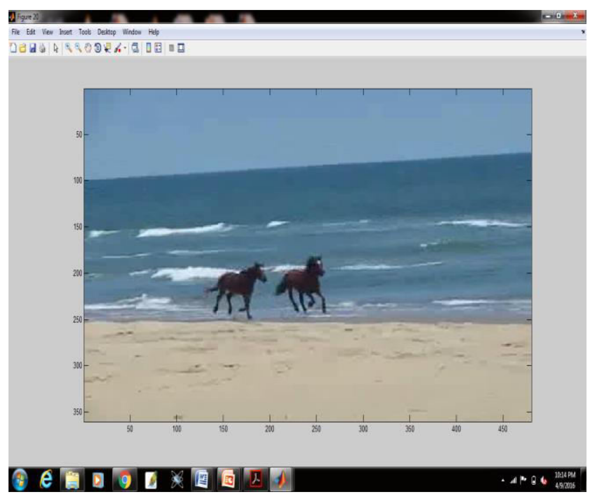
Frame1



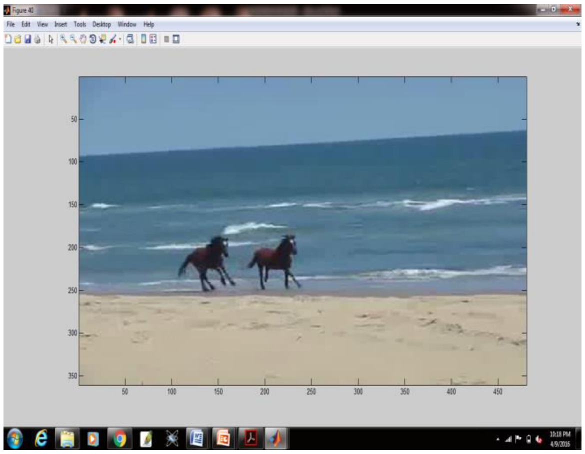
Frame 10



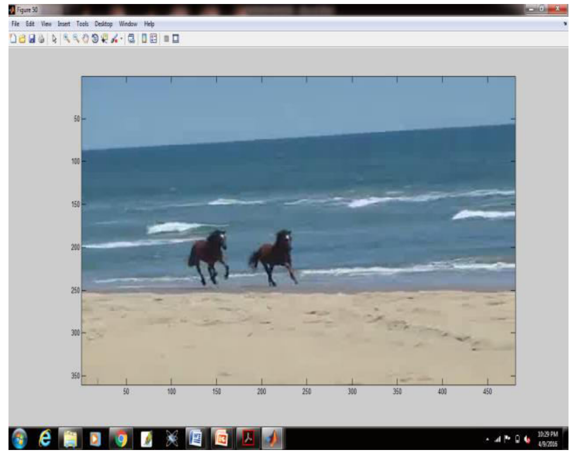
Frame 20



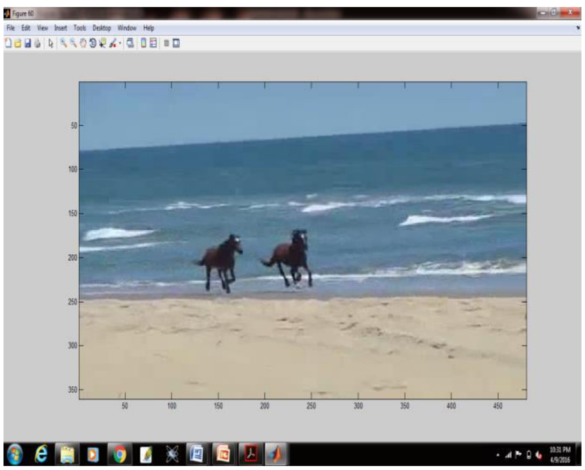
Frame 30



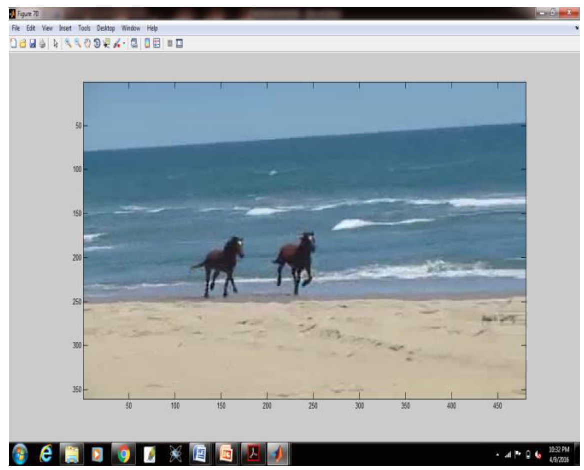
Frame 40



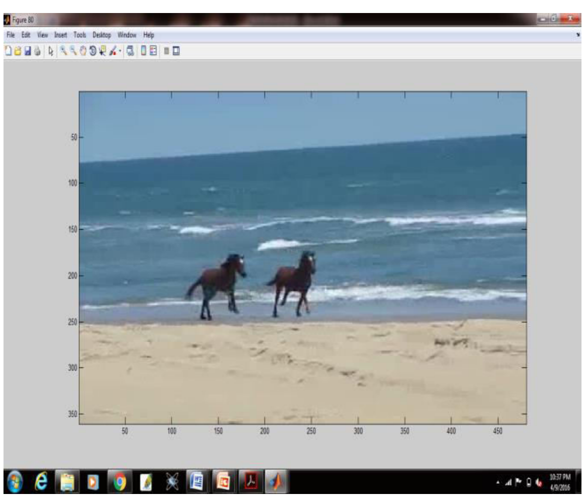
Frame 50



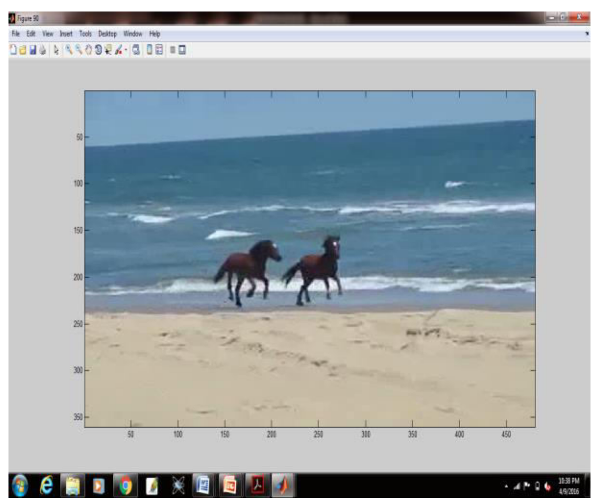
Frame 60



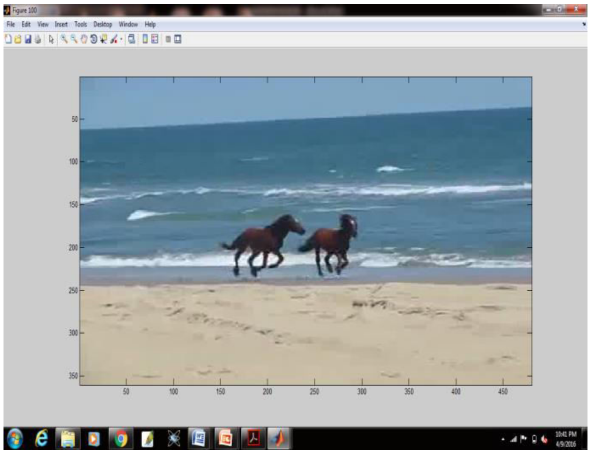
Frame 70



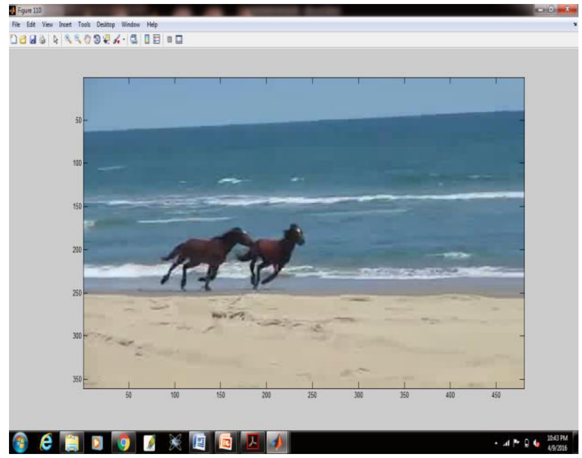
Frame 80



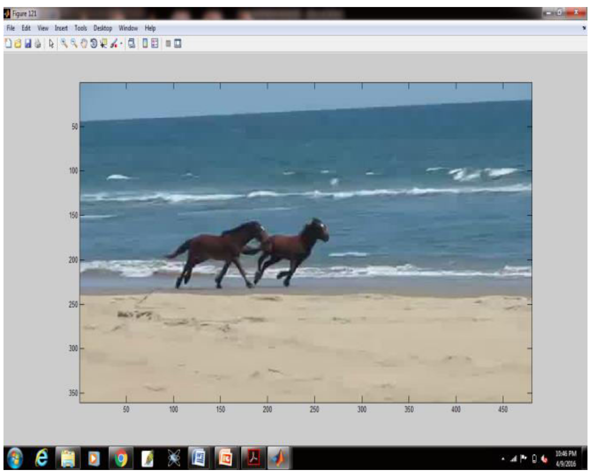
Frame 90



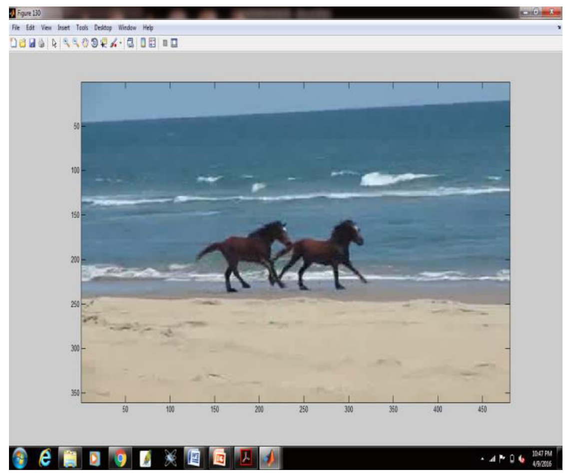
Frame 100



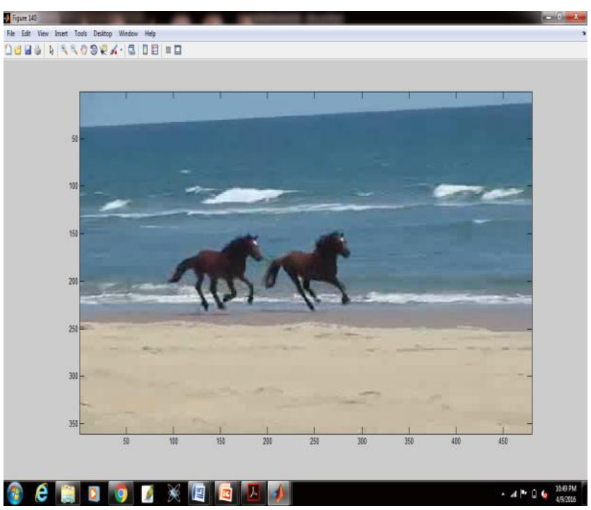
Frame 110



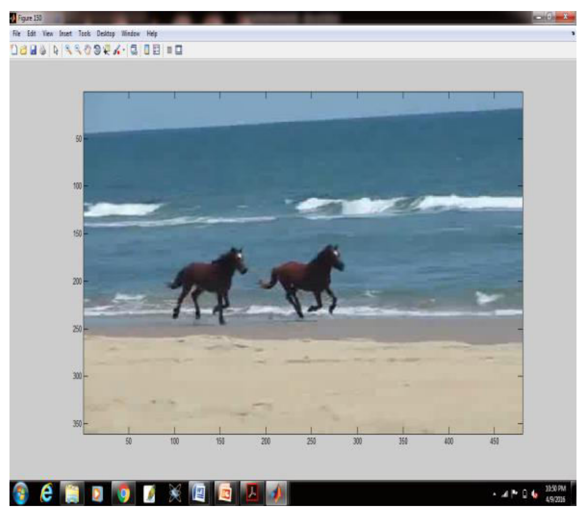
Frame 120



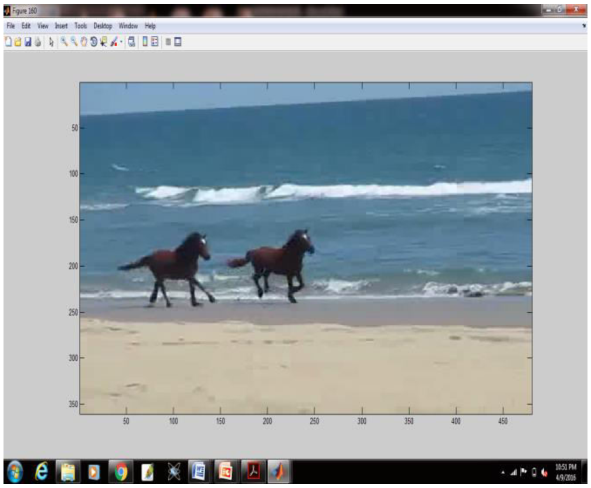
Frame 130



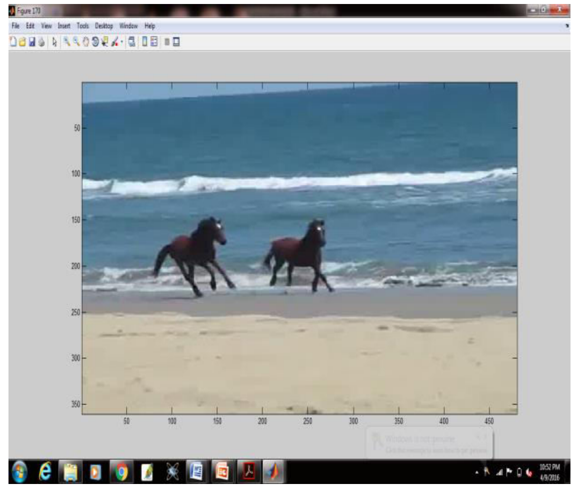
Frame 140



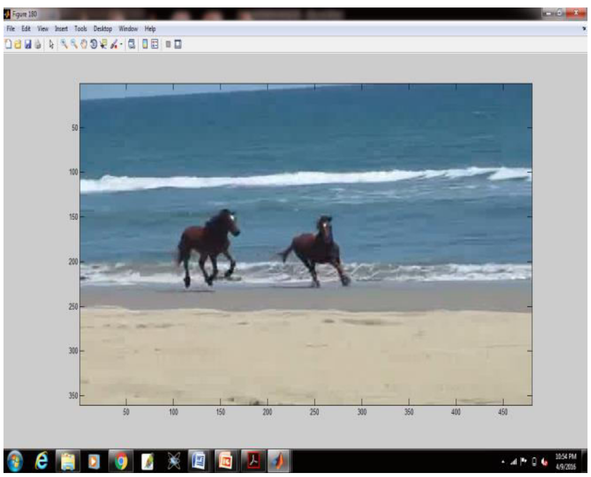
Frame 150



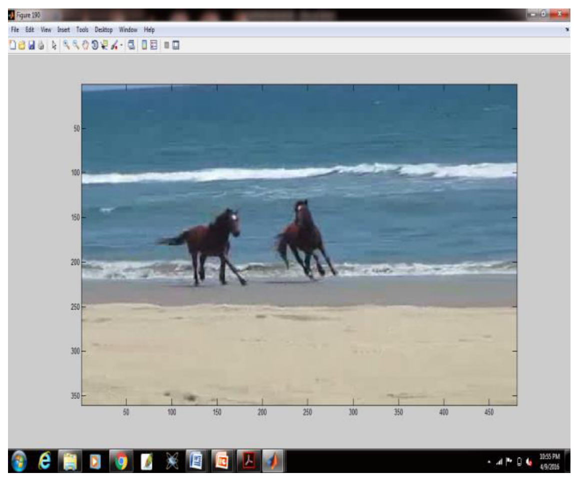
Frame 160



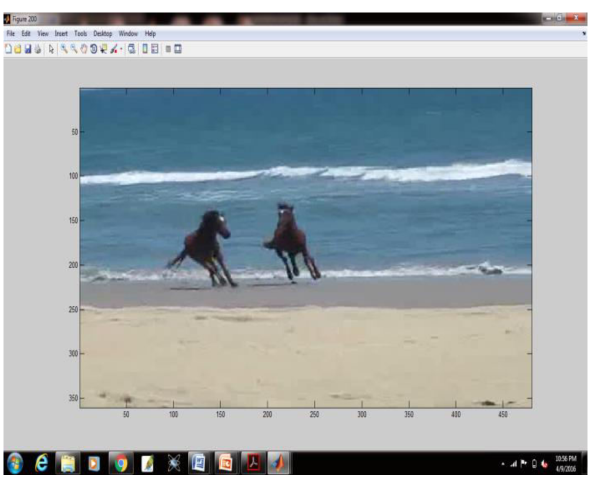
Frame 170



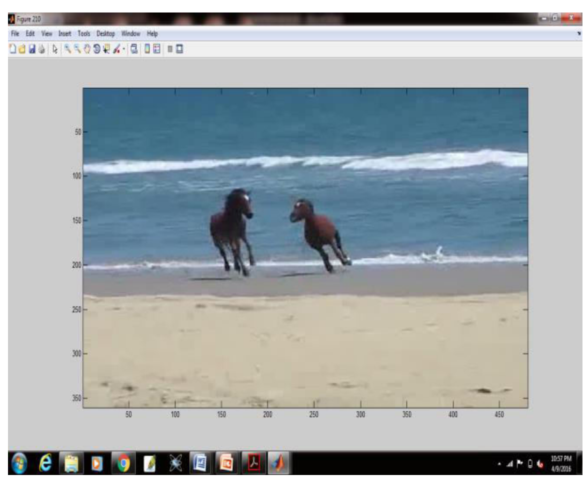
Frame 180



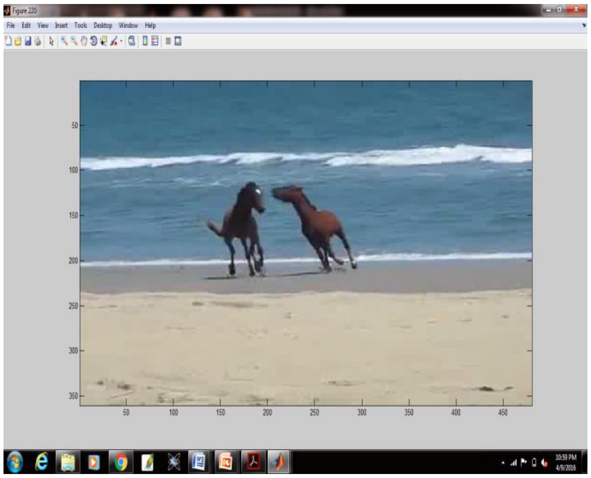
Frame 190



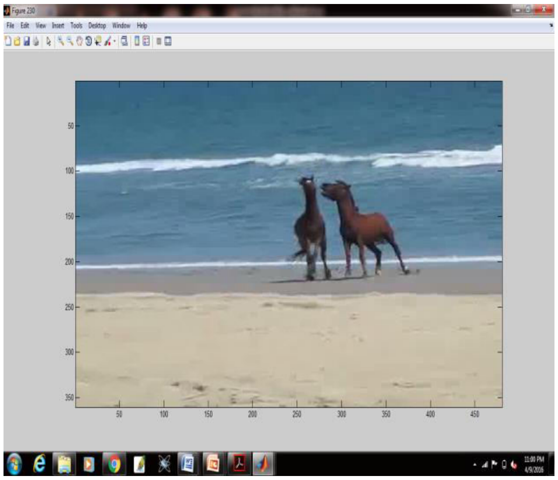
Frame 200



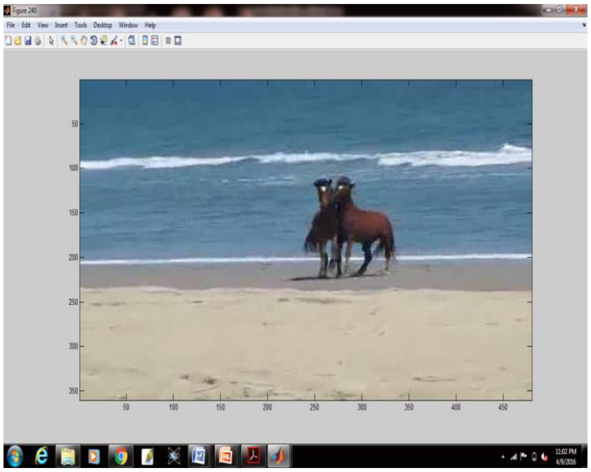
Frame 210



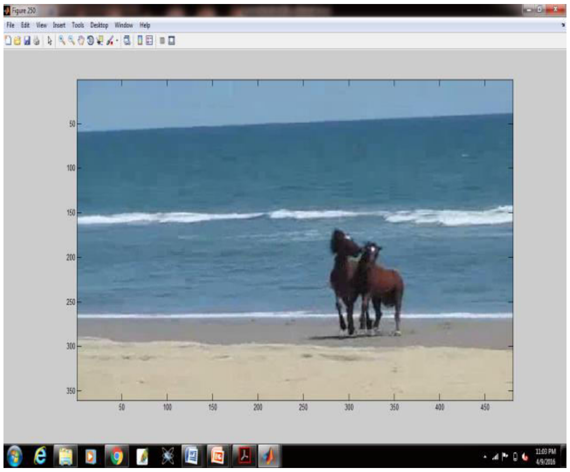
Frame 220



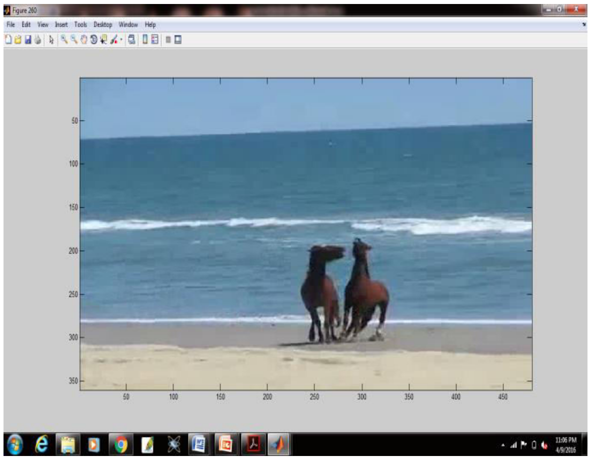
Frame 230



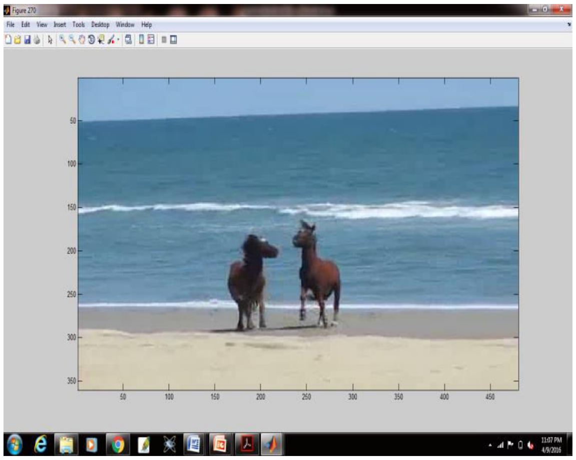
Frame 240



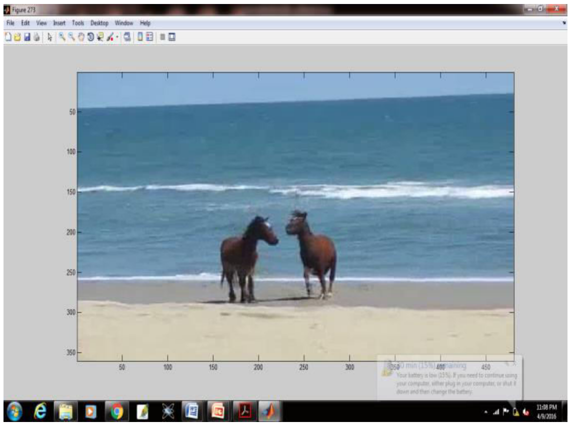
Frame 250



Frame 260



Frame 270



Frame 273

Step-3: Initialization of the background frame

The background is found by taking the mean of all the intensity values from all the frames in the video. Such a background of our video is shown below.

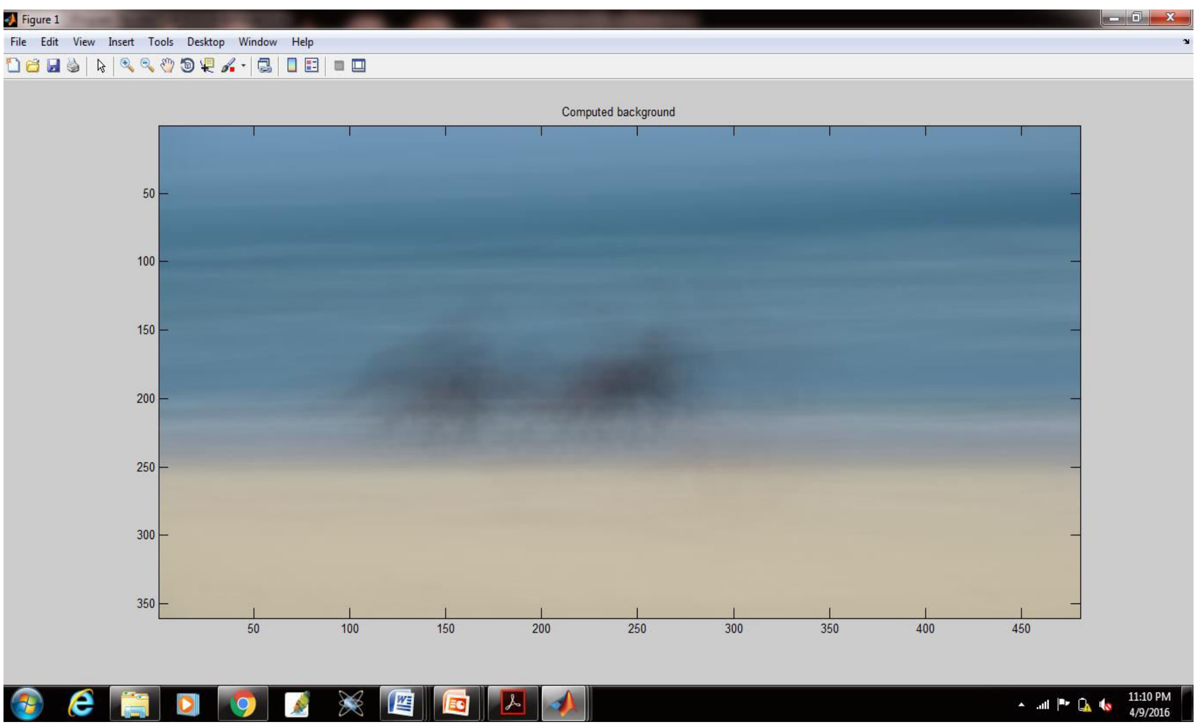
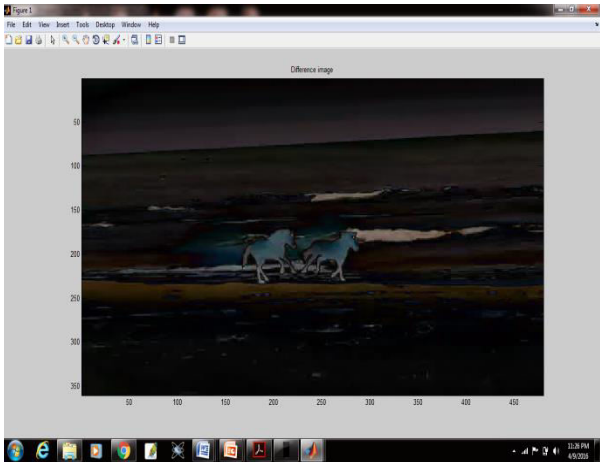


Fig 6.3- Background Frame

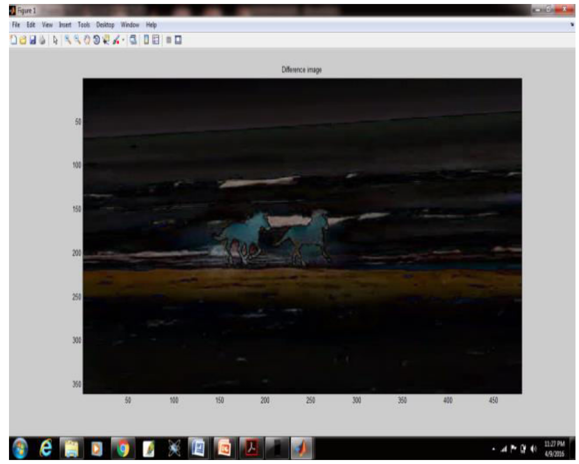
Step-4: Subtraction of the background frame from current frame

Now the above background is subtracted from all the 273 frames continuously in a loop and the difference images formed are as shown below.

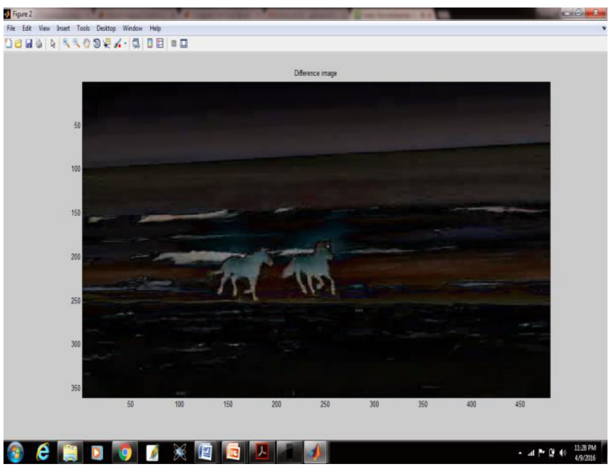
Fig Box 6.4- Results of subtracting Background from frames



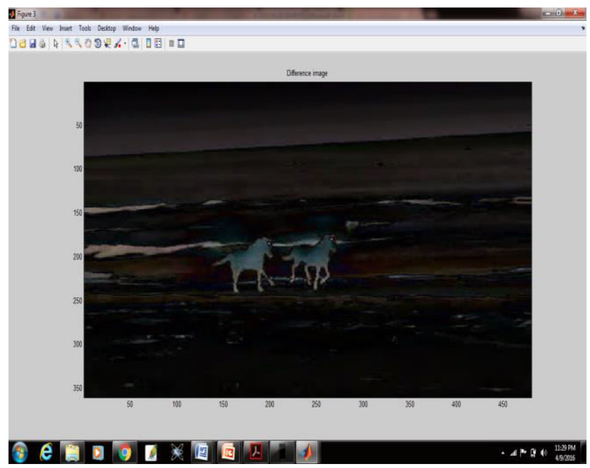
Difference Image of frame 1



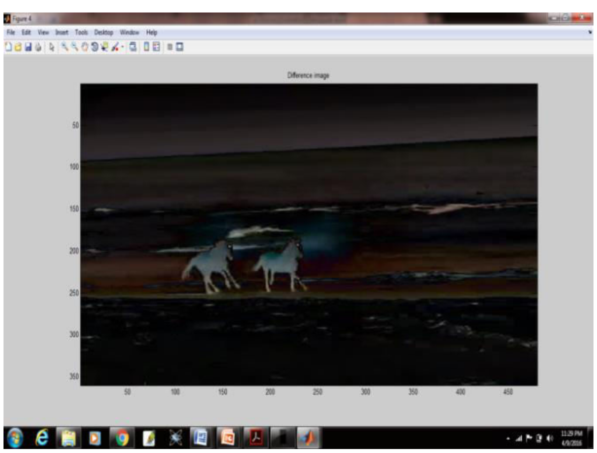
Difference Image of frame 10



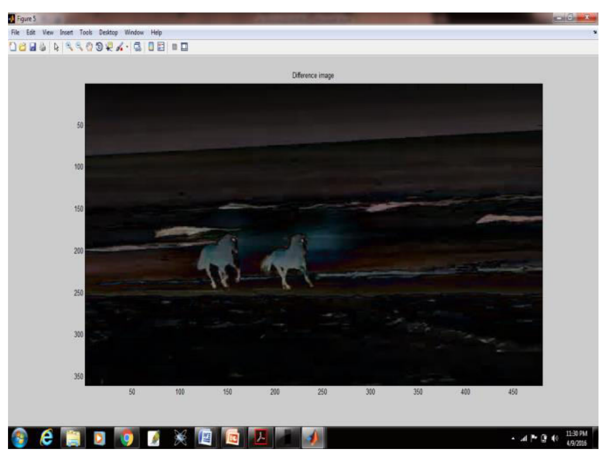
Difference Image of frame 20



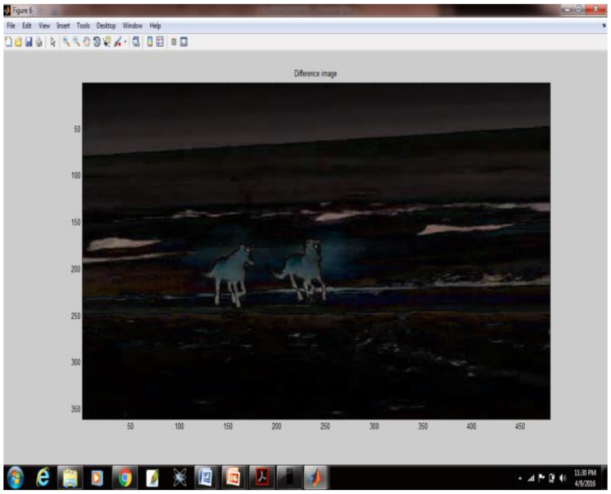
Difference Image of frame 30



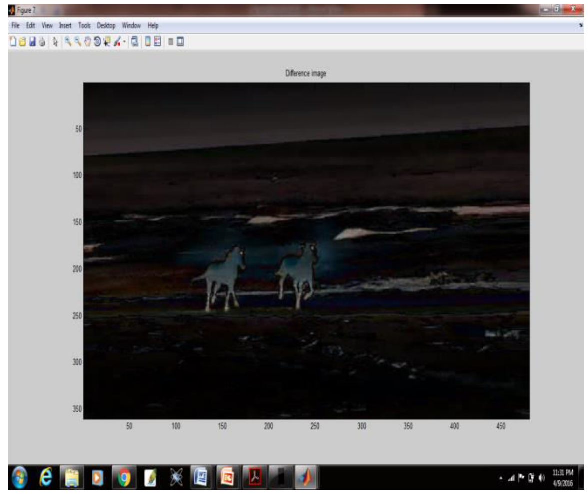
Difference Image of frame 40



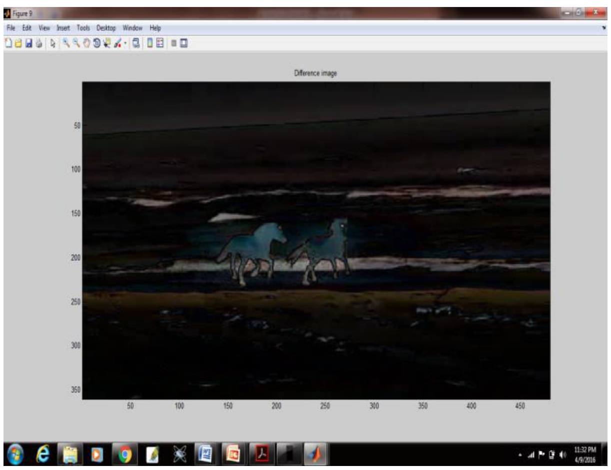
Difference Image of frame 50



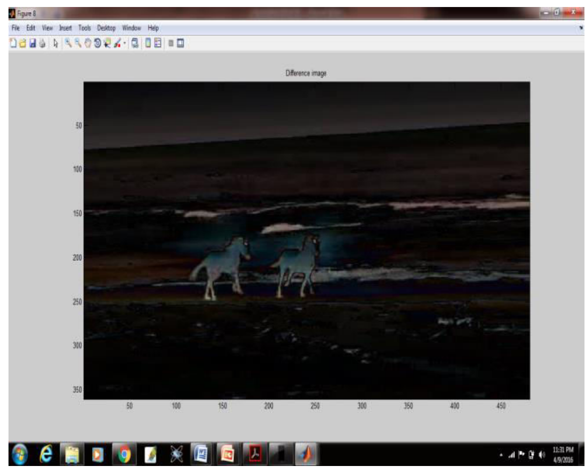
Difference Image of frame 60



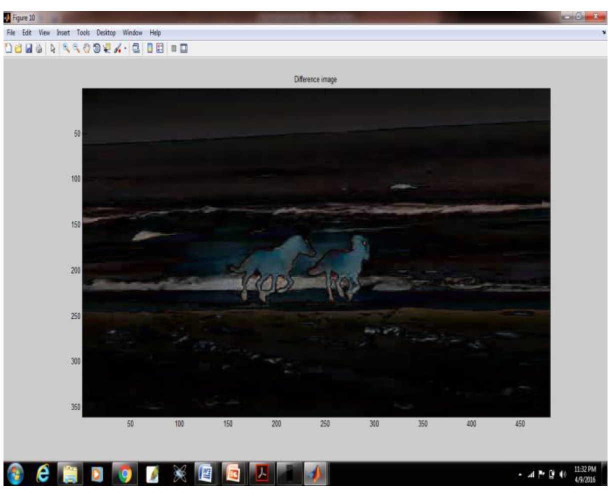
Difference Image of frame 70



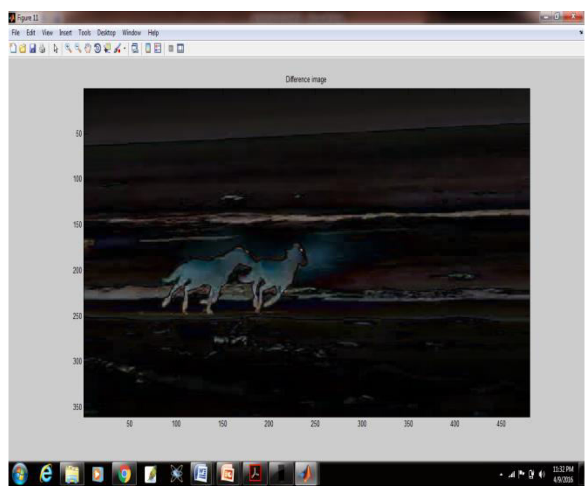
Difference Image of frame 80



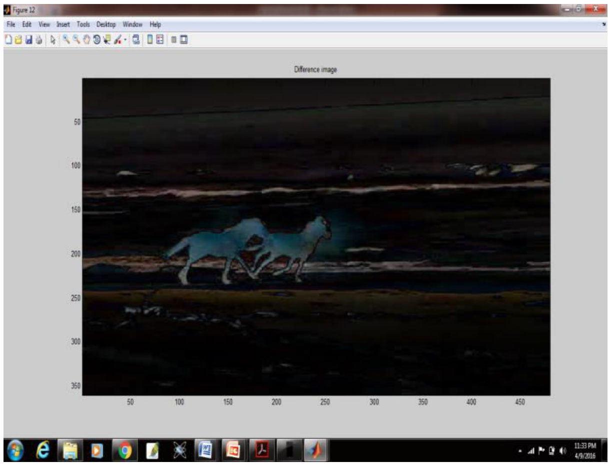
Difference Image of frame 90



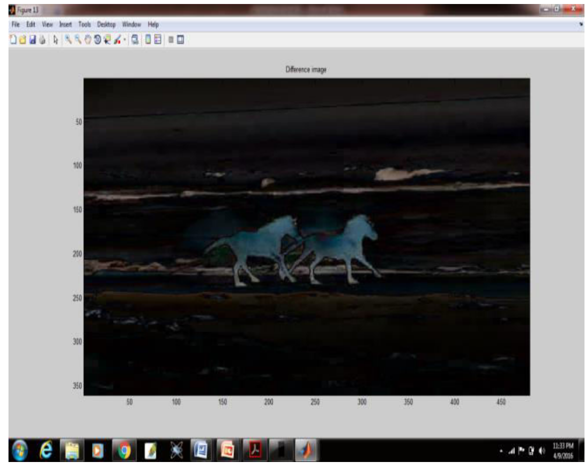
Difference Image of frame 100



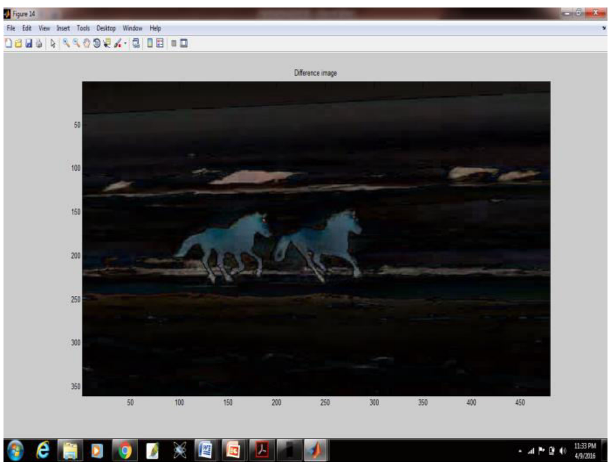
Difference Image of frame 110



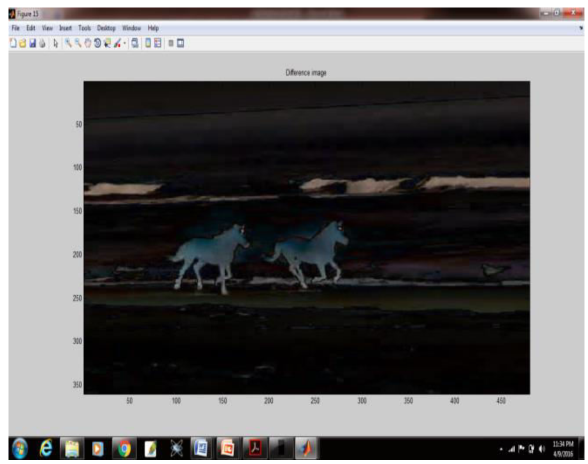
Difference Image of frame 120



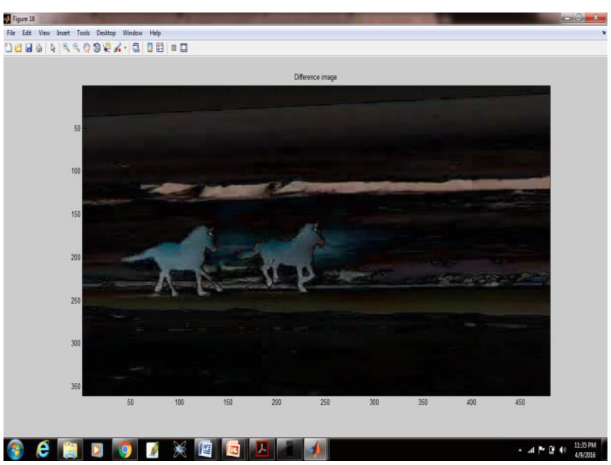
Difference Image of frame 130



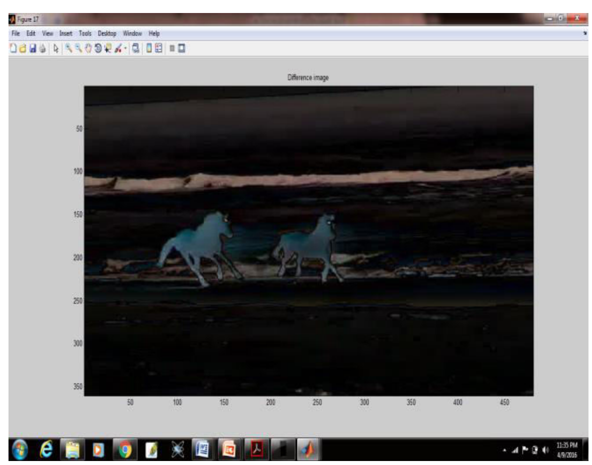
Difference Image of frame 140



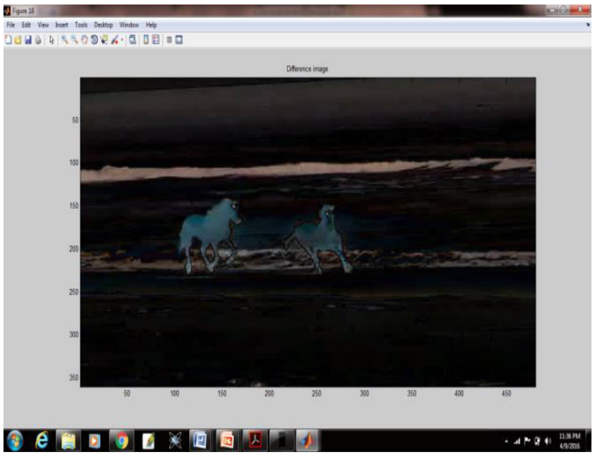
Difference Image of frame 150



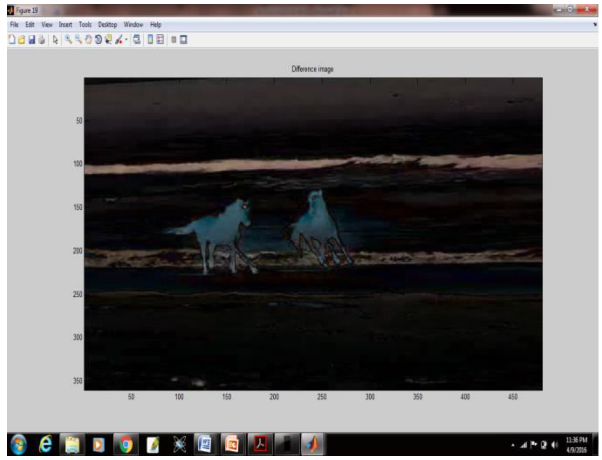
Difference Image of frame 160



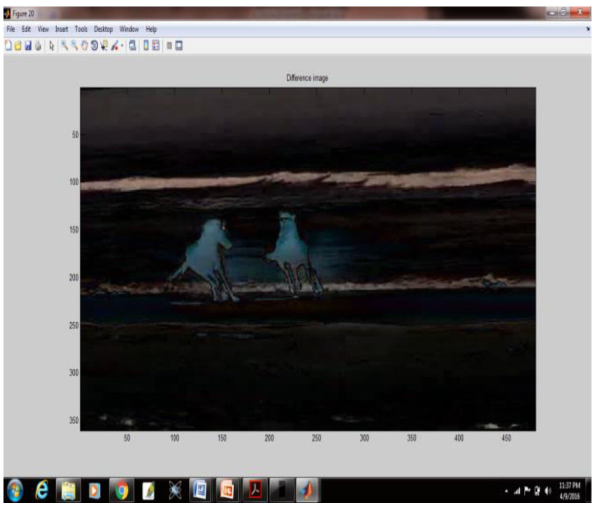
Difference Image of frame 170



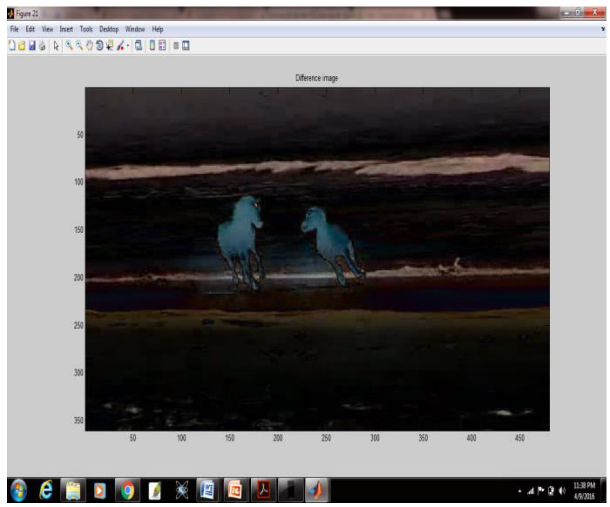
Difference Image of frame 180



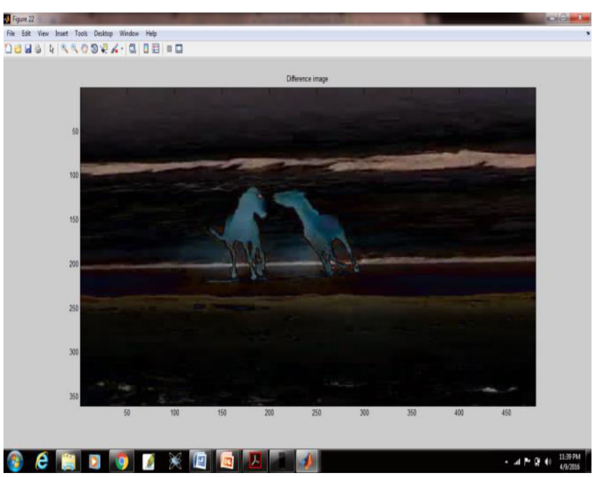
Difference Image of frame 190



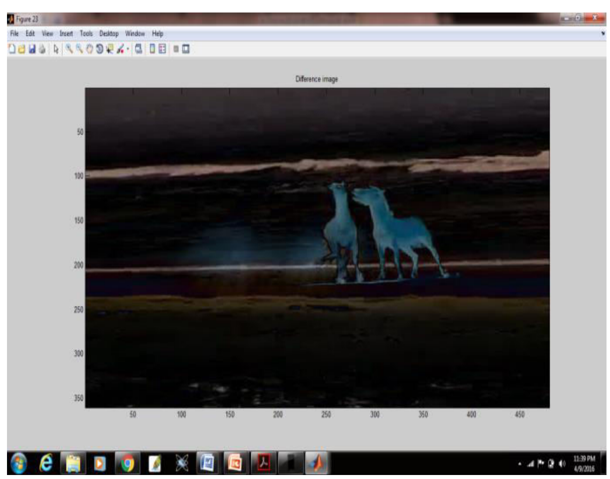
Difference Image of frame 200



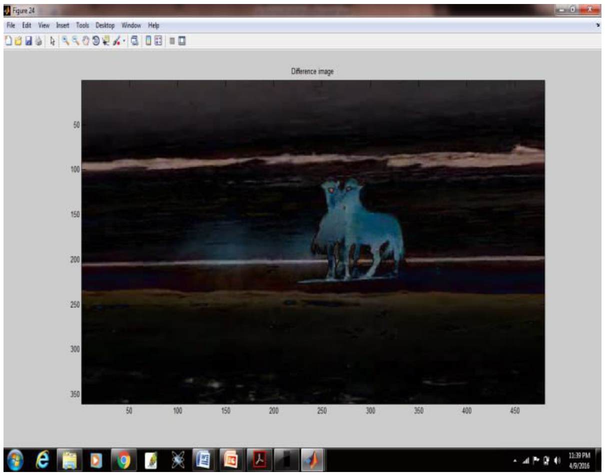
Difference Image of frame 210



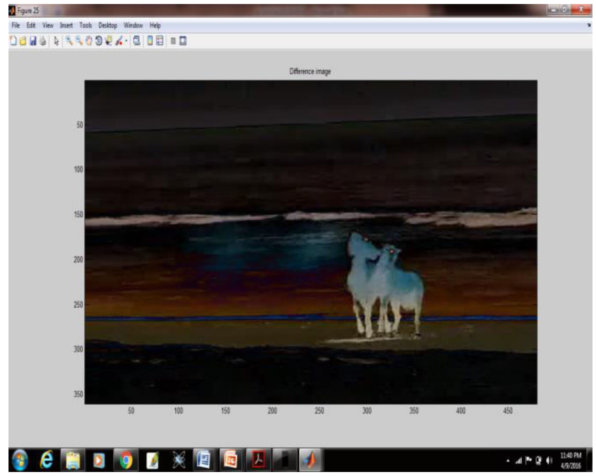
Difference Image of frame 220



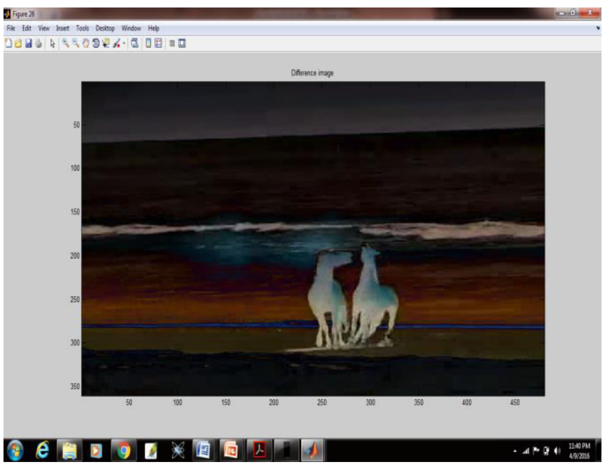
Difference Image of frame 230



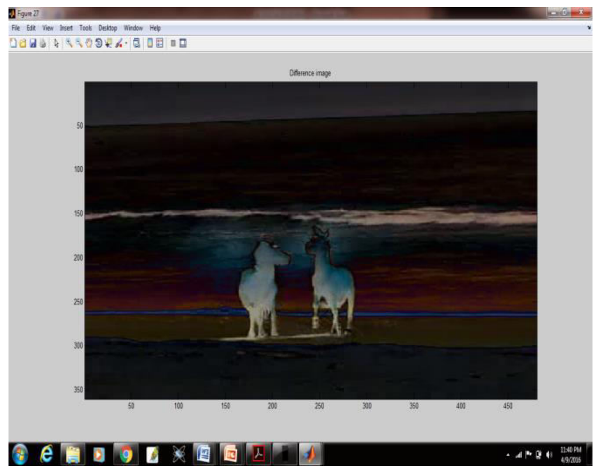
Difference Image of frame 240



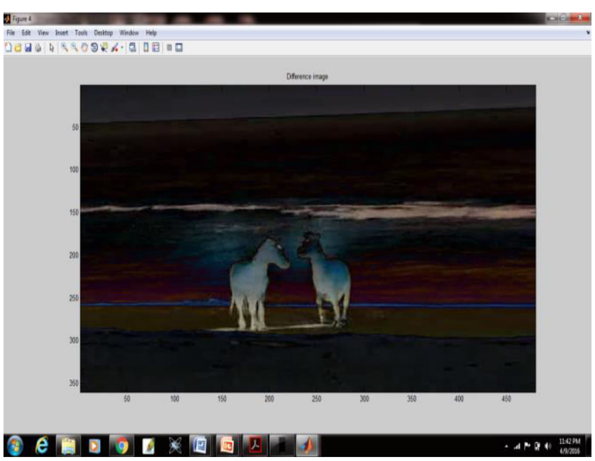
Difference Image of frame 250



Difference Image of frame 260



Difference Image of frame 270

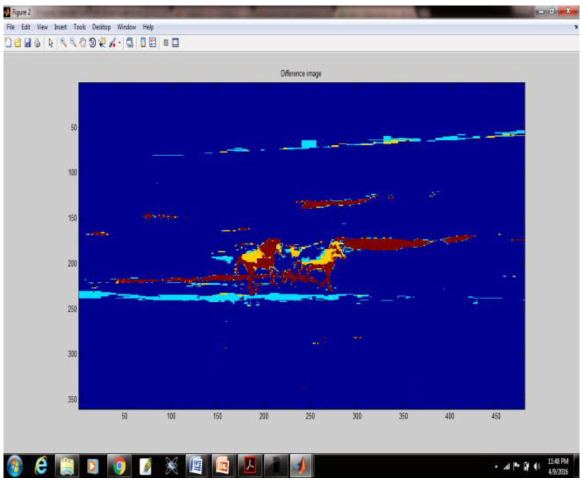


Difference Image of frame 273

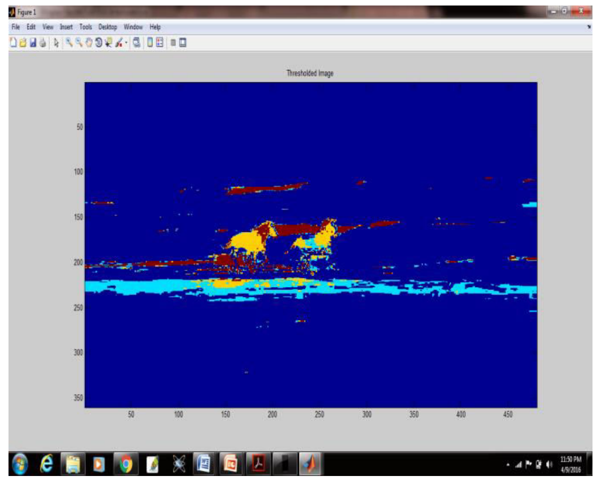
Step-5: Thresholding

Now for the difference images obtained a threshold is fixed and further processing is done i.e., the intensity values of pixels above the threshold are considered as foreground and below threshold are considered as background pixels.

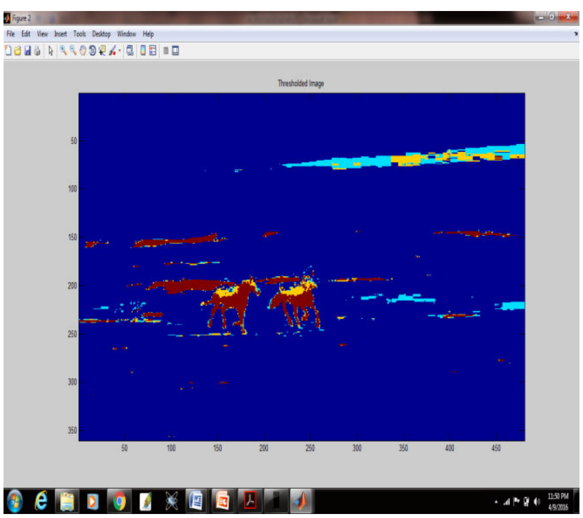
Fig Box 6.5- Results of Thresholding



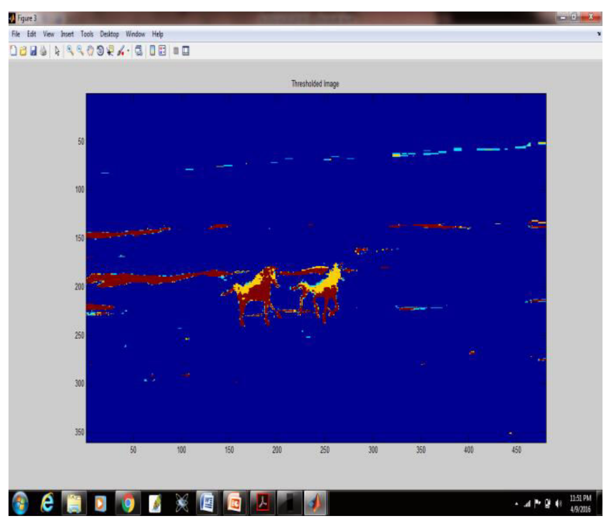
frame 1 after thresholding



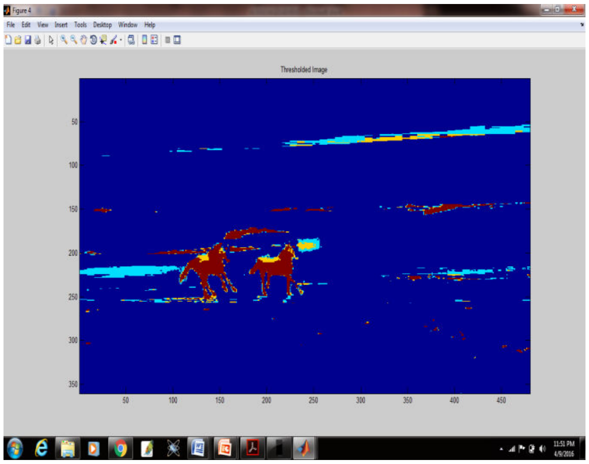
frame 10 after thresholding



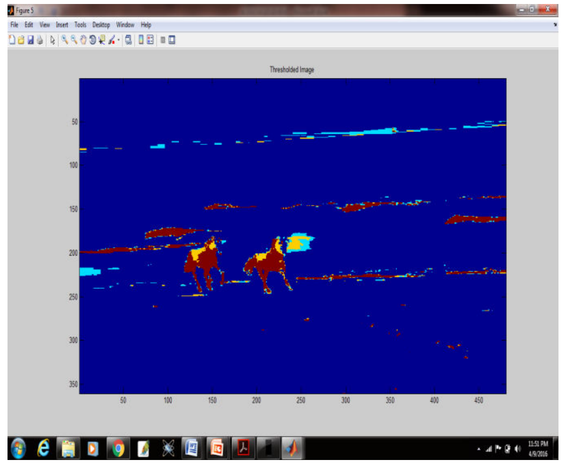
frame 20 after thresholding



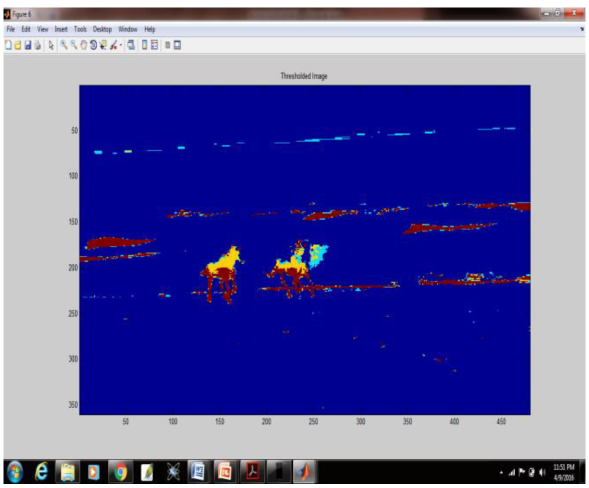
frame 30 after thresholding



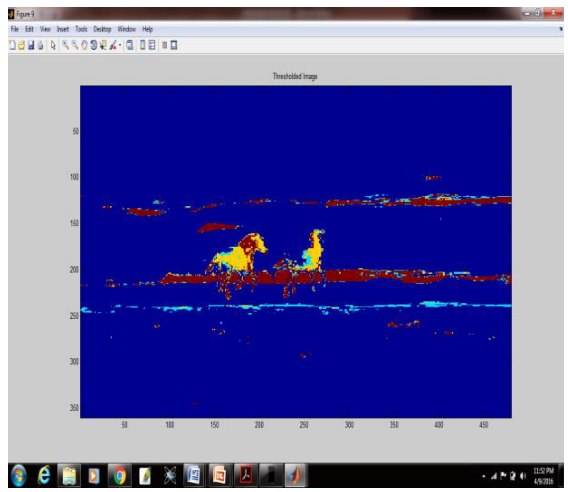
frame 40 after thresholding



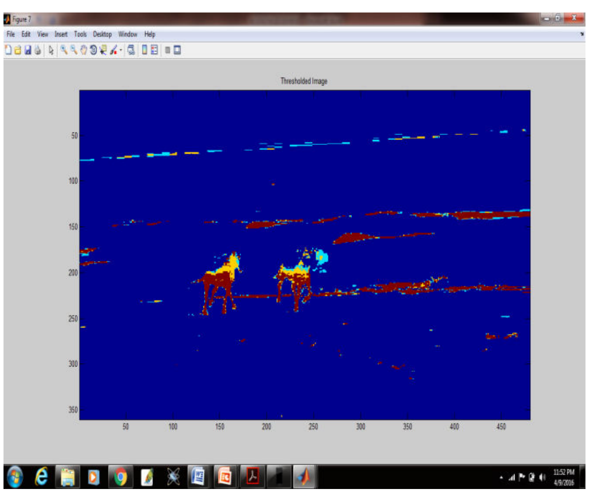
frame 50 after thresholding



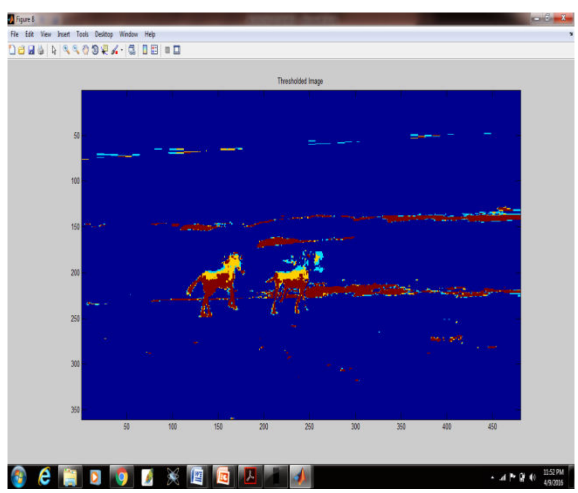
frame 60 after thresholding



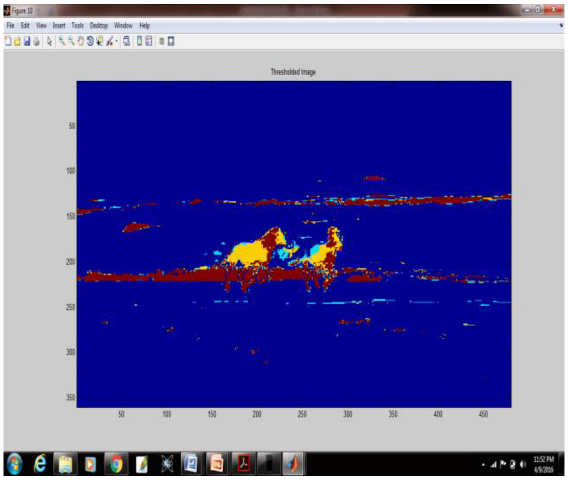
frame 70 after thresholding



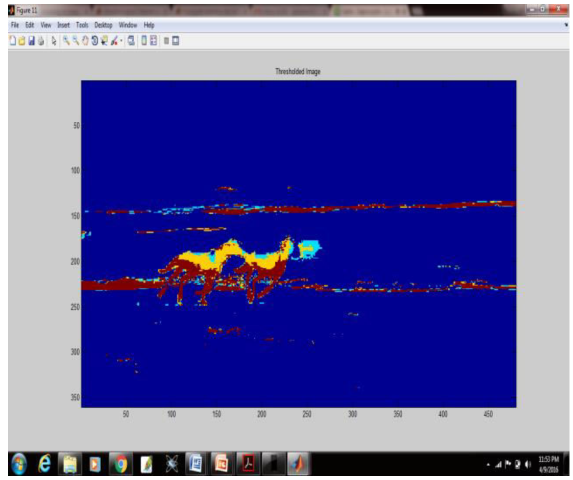
frame 80 after thresholding



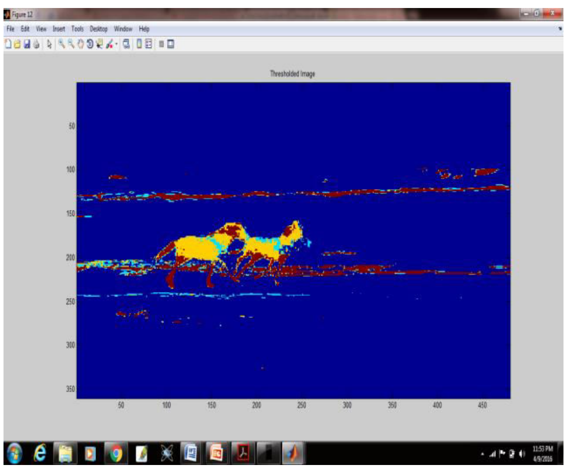
frame 90 after thresholding



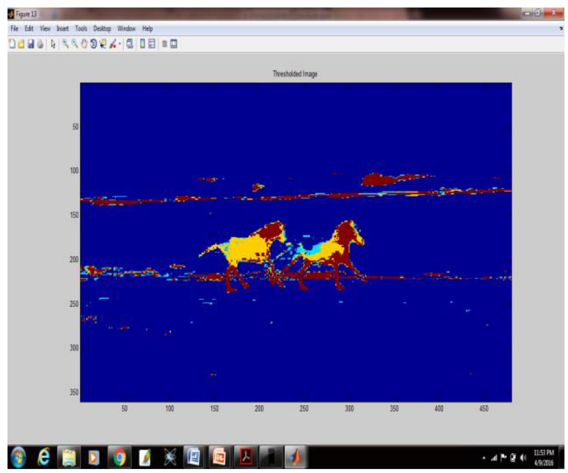
frame 100 after thresholding



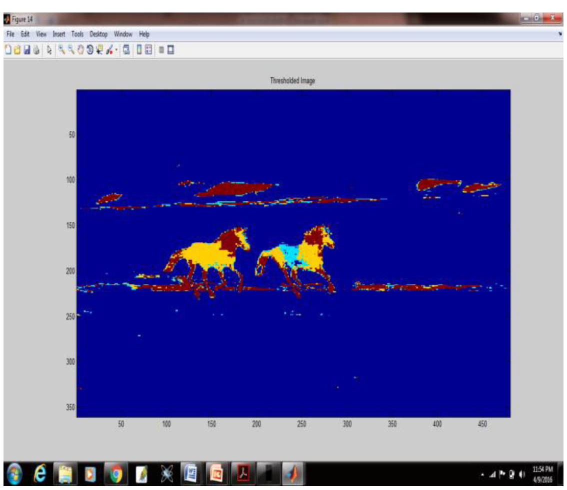
frame 110 after thresholding



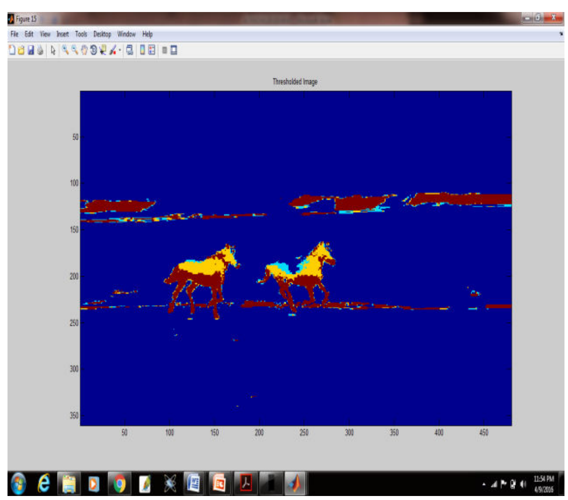
frame 120 after thresholding



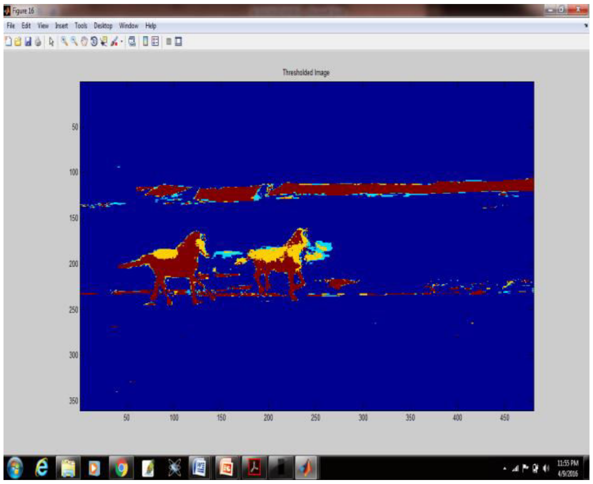
frame 130 after thresholding



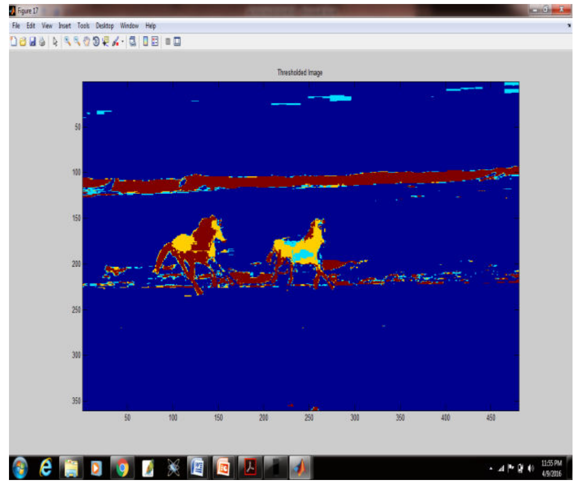
frame 140 after thresholding



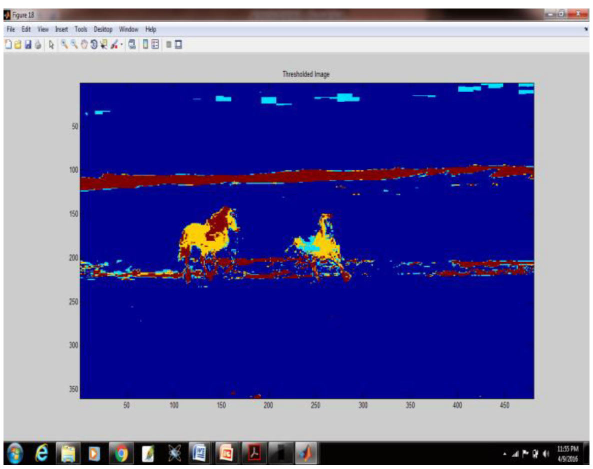
frame 150 after thresholding



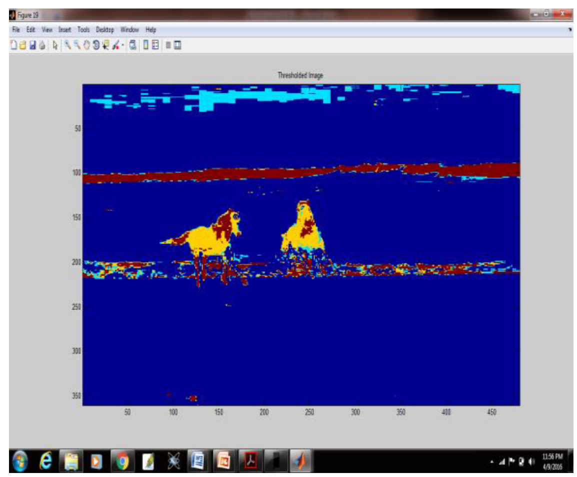
frame 160 after thresholding



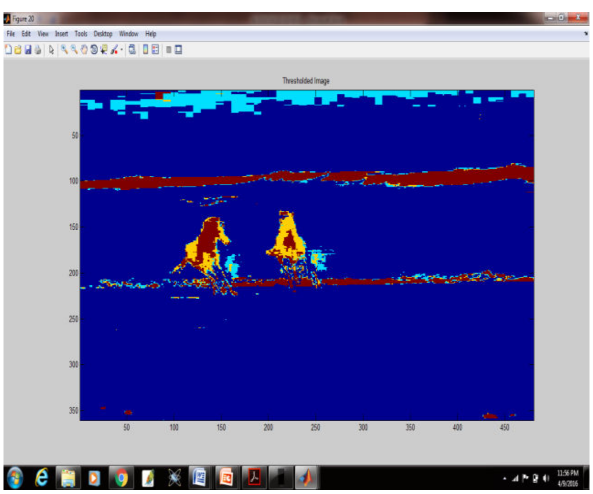
frame 170 after thresholding



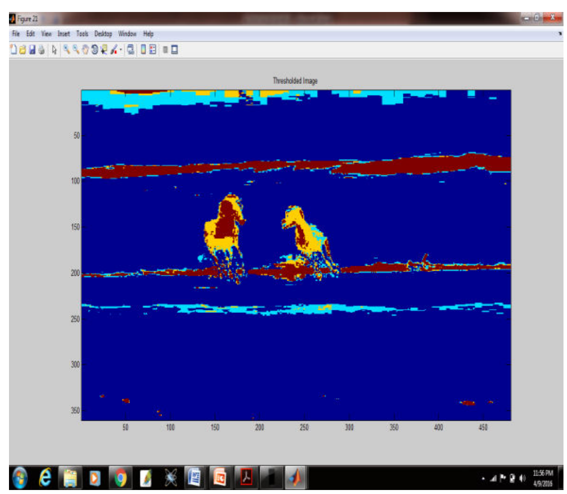
frame 180 after thresholding



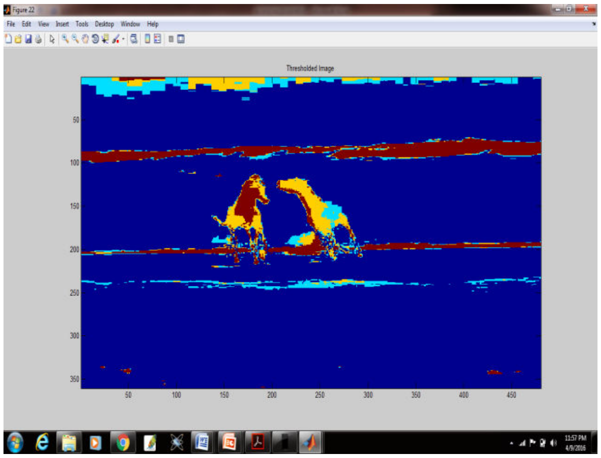
frame 190 after thresholding



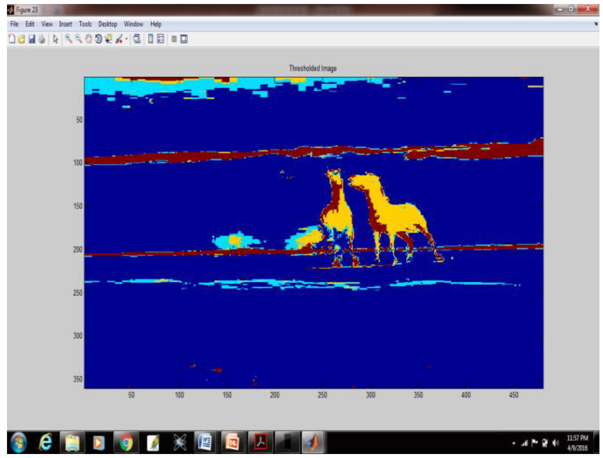
frame 200 after thresholding



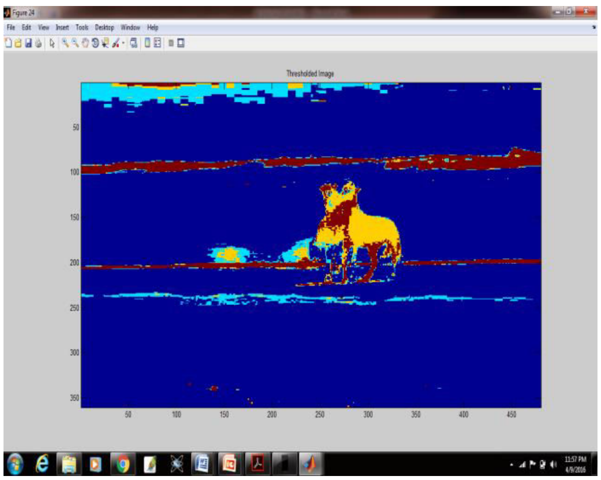
frame 210 after thresholding



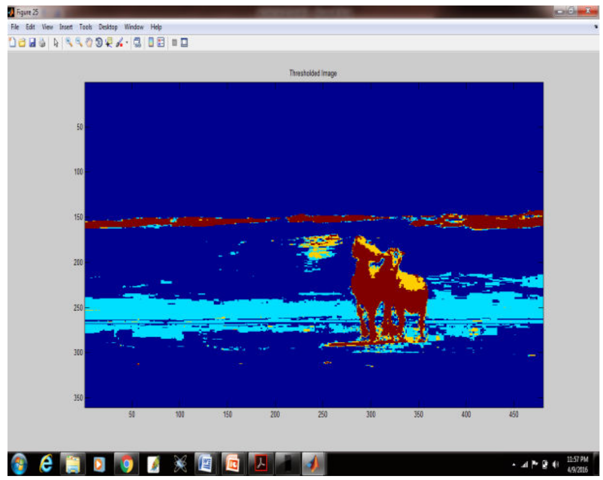
frame 220 after thresholding



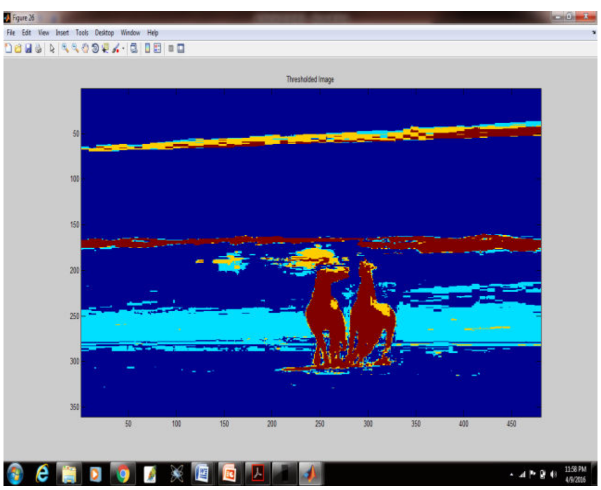
frame 230 after thresholding



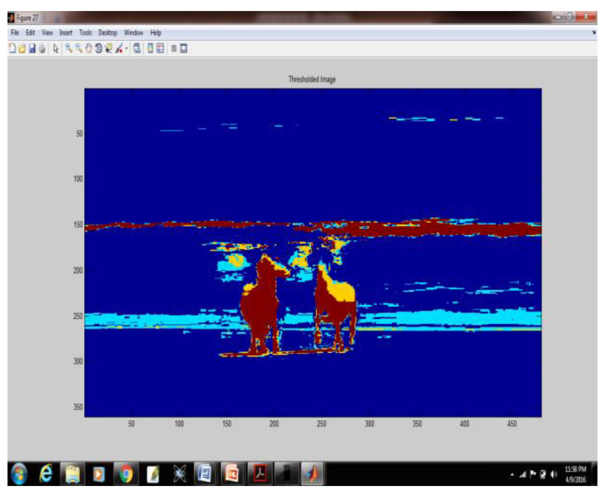
frame 240 after thresholding



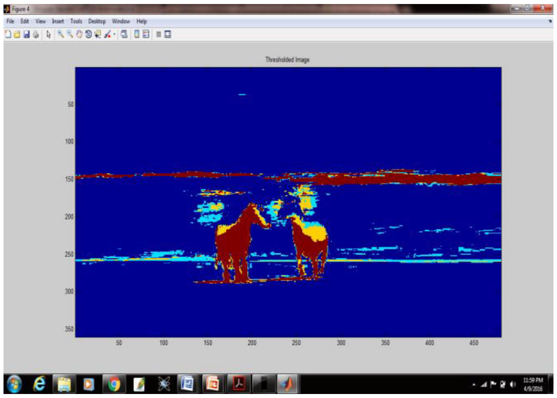
frame 250 after thresholding



frame 260 after thresholding



frame 270 after thresholding



frame 273 after thresholding

Step-6: Morphological Processing

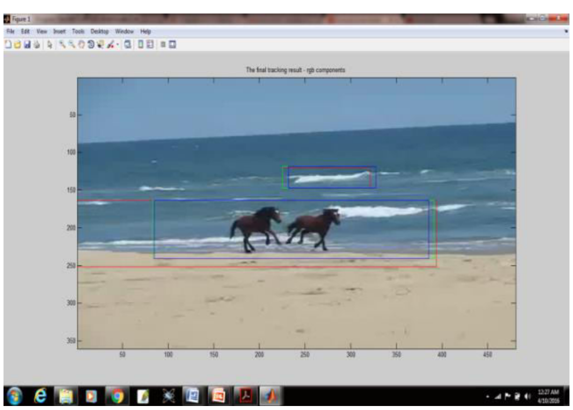
In this step the frames obtained after thresholding are dilated, so as to remove noises occurred while capturing the video or due to jitters.

Step-7: Blob Analysis

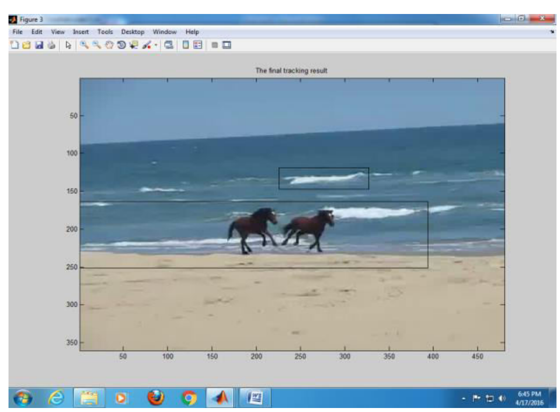
In this step after morphological filtering, all the foreground pixels which are connected (through 8-connectivity) are collected into an area and a bounding box is drawn around that area so that those boxes represent the moving objects in the respective frames.

The moving objects in the RGB layers of an image frame are tracked and are shown below with red, blue and green bounding boxes. Also correspondingly wholly tracked object in the complete frame is shown by a yellow bounding box.

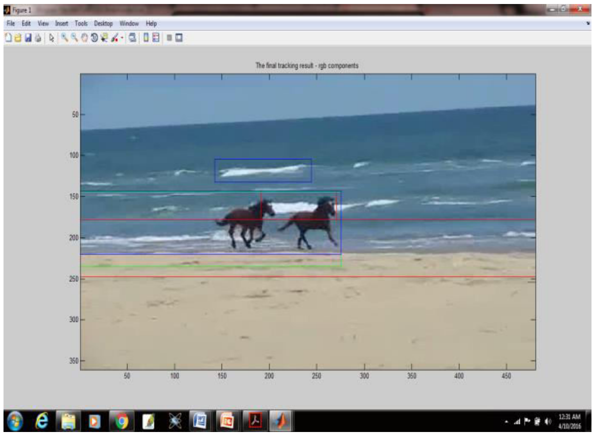
Fig Box 6.6- Results of Morphological Filtering and Blob Analysis



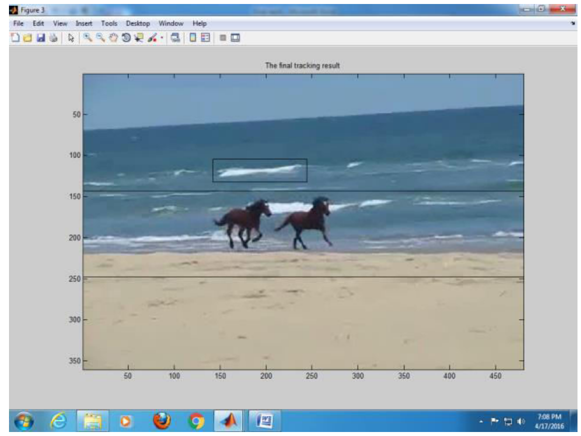
RGB frame1



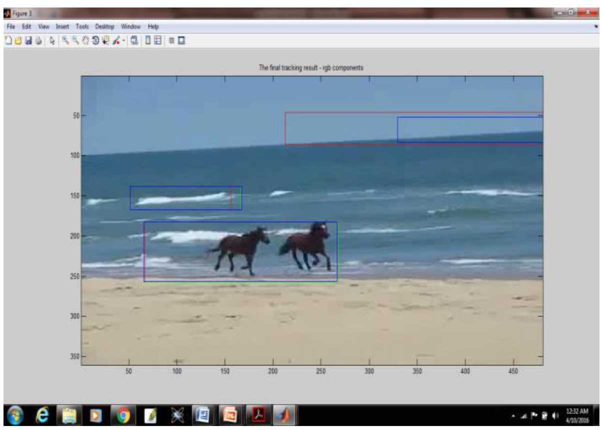
Moving Object in Frame1



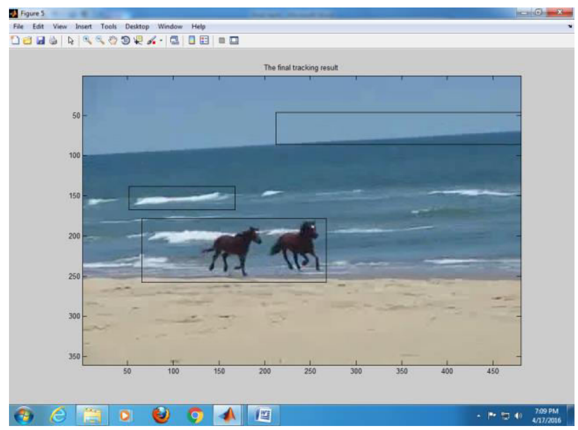
RGB frame 10



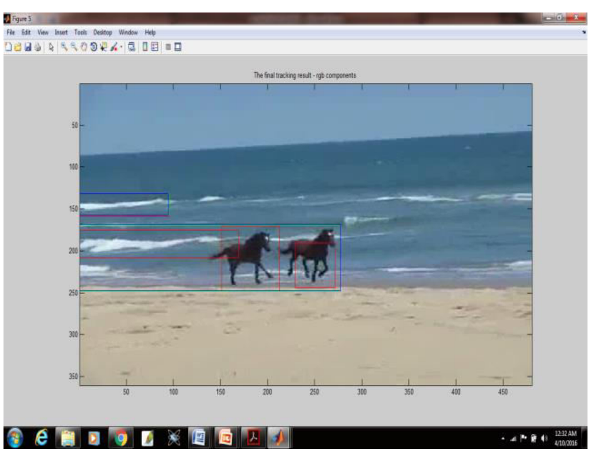
Moving Object in Frame 10



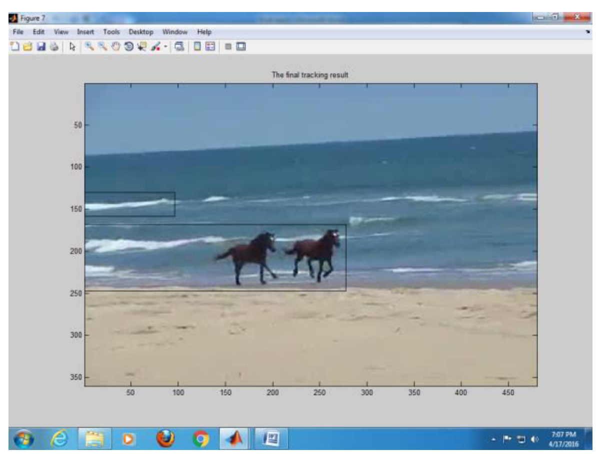
RGB frame 20



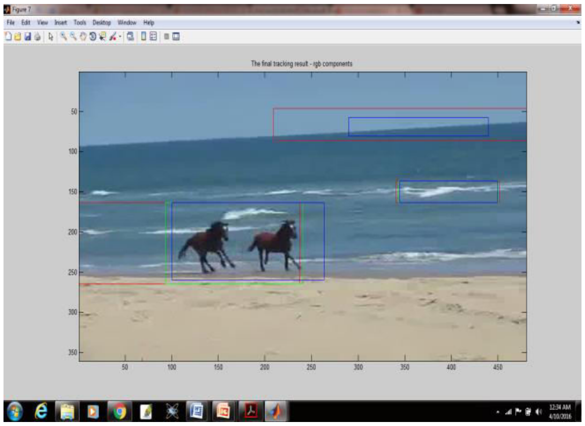
Moving Object in Frame 20



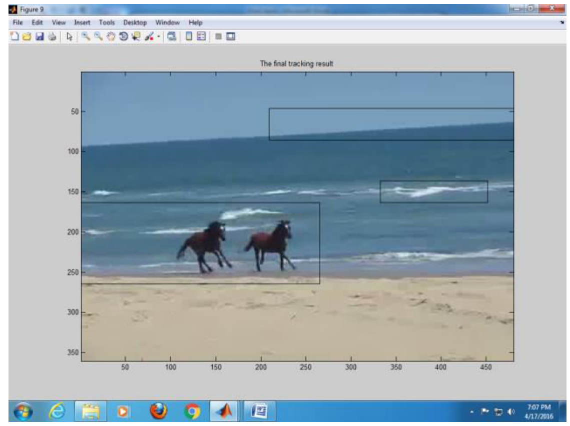
RGB frame30



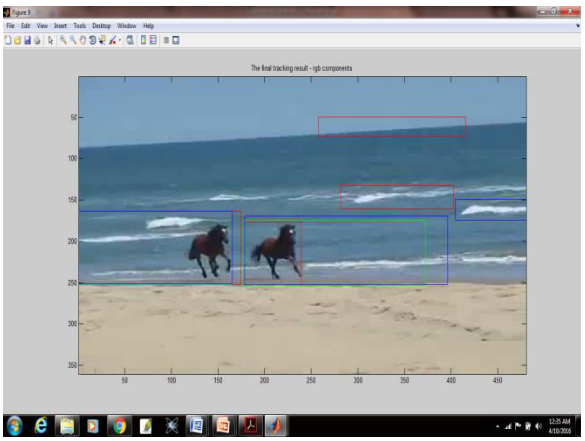
Moving Object in Frame30



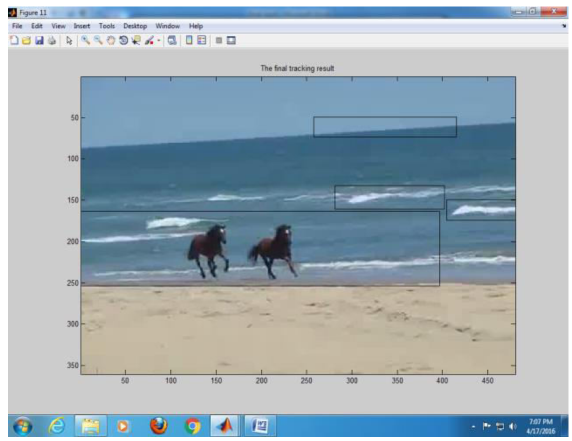
RGB frame40



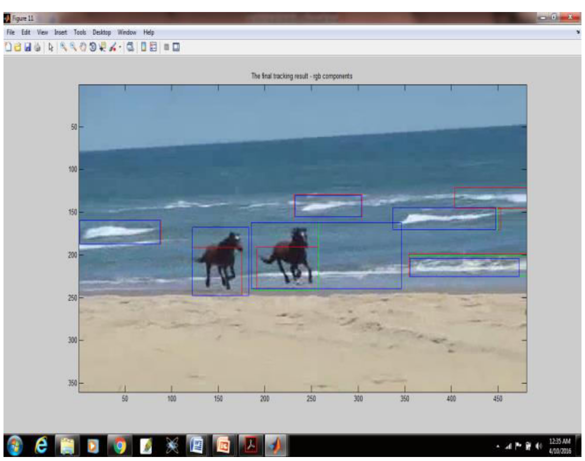
Moving Object in Frame40



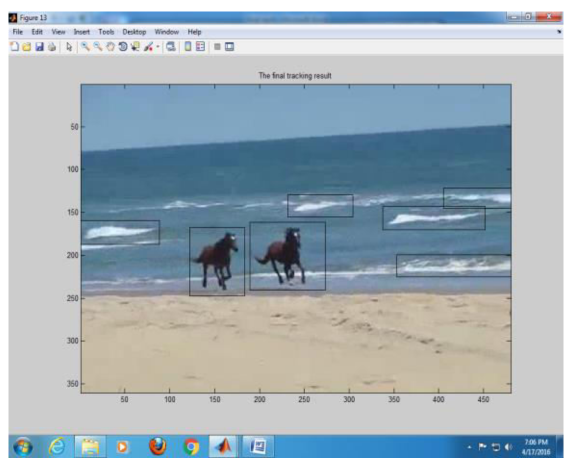
RGB frame50



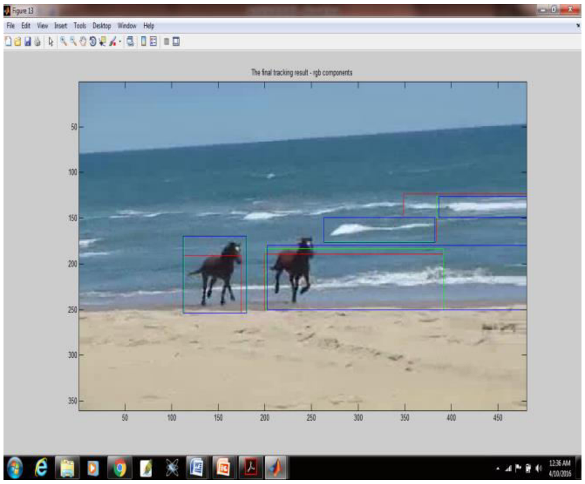
Moving Object in Frame50



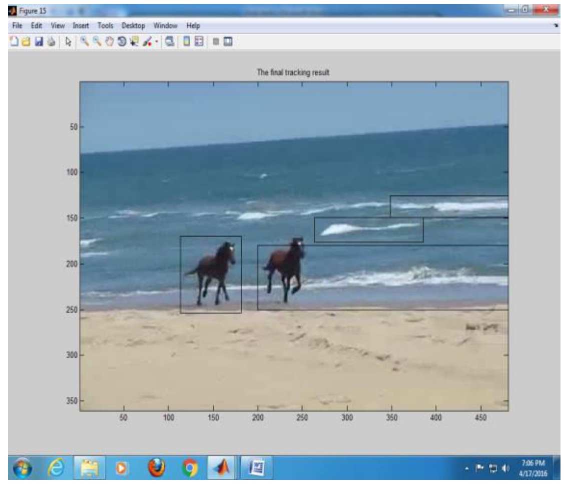
RGB frame60



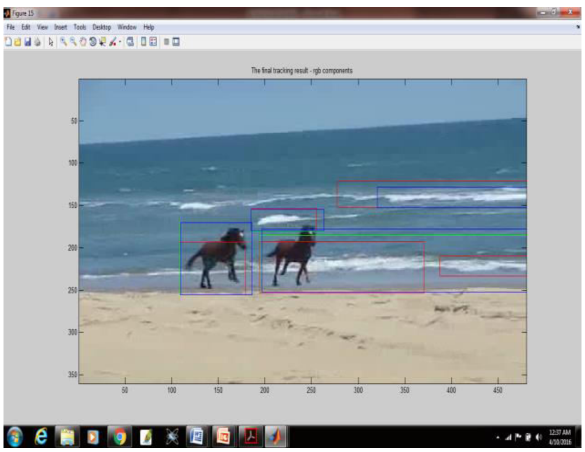
Moving Object in Frame60



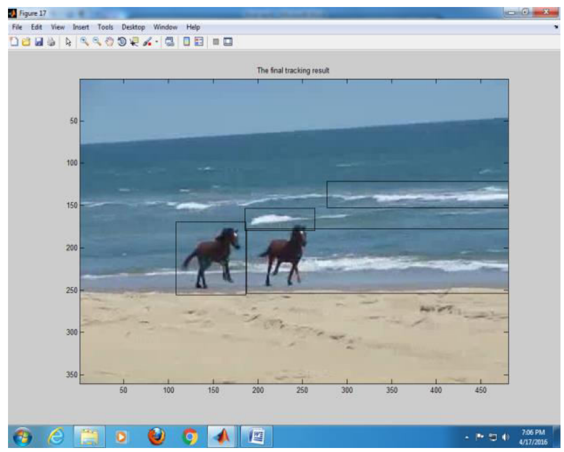
RGB frame70



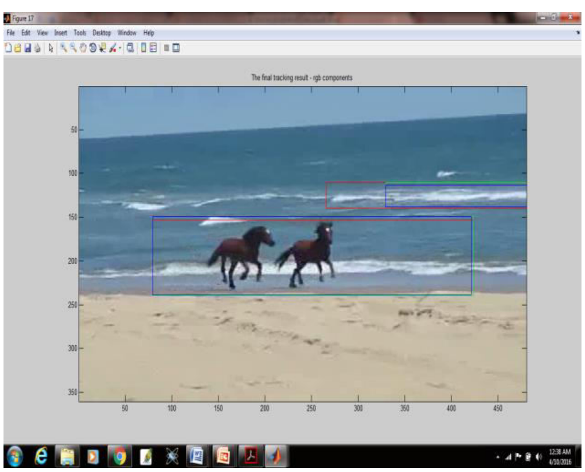
Moving Object in Frame70



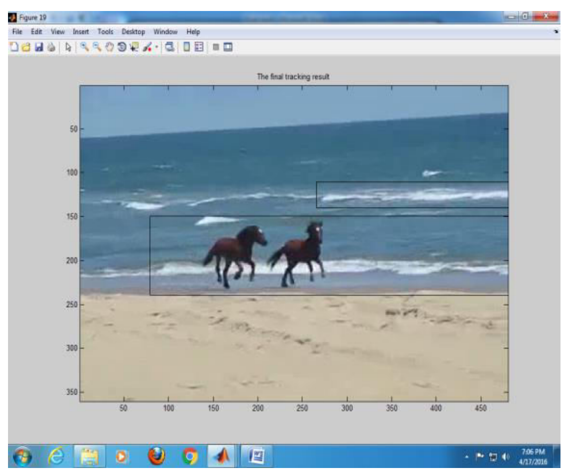
RGB frame80



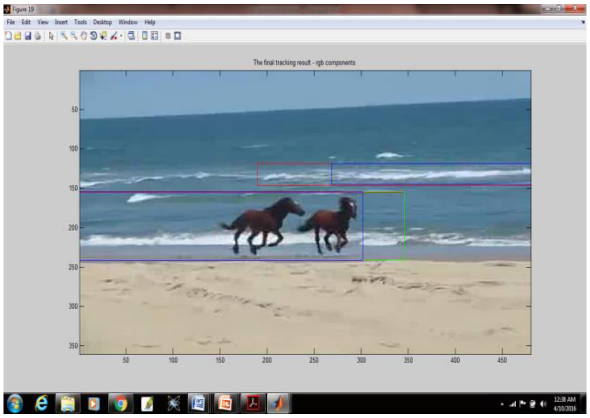
Moving Object in Frame80



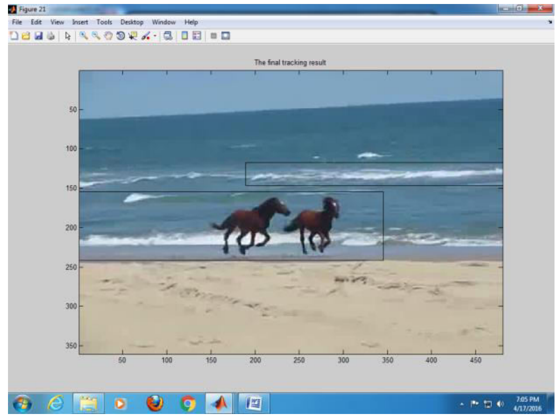
RGB frame90



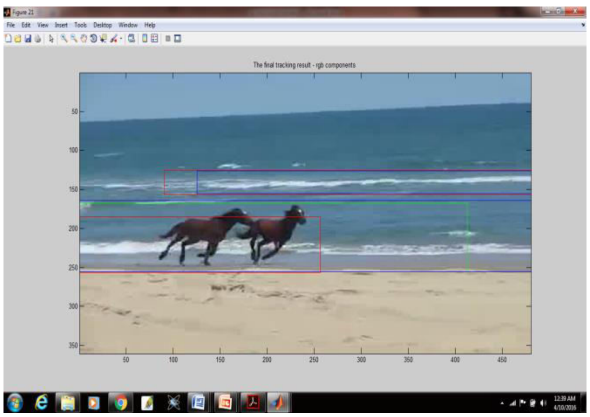
Moving Object in Frame90



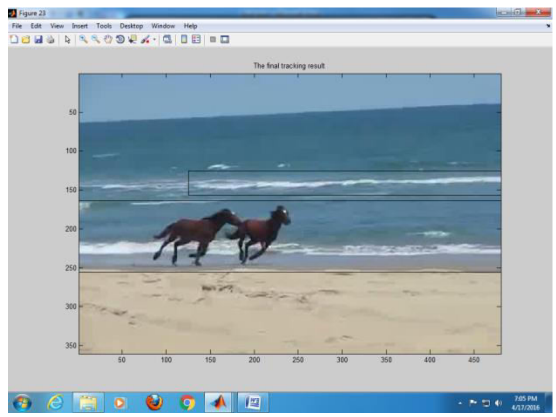
RGB frame100



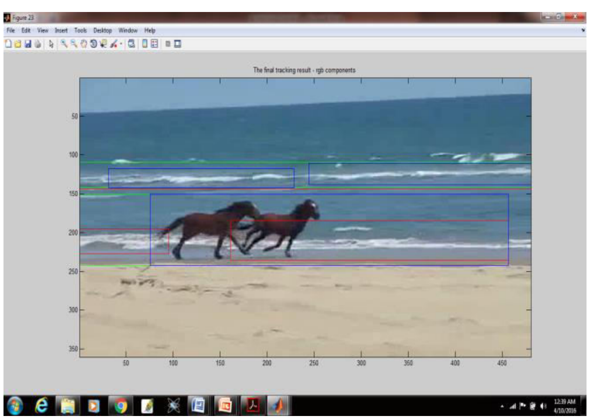
Moving Object in Frame100



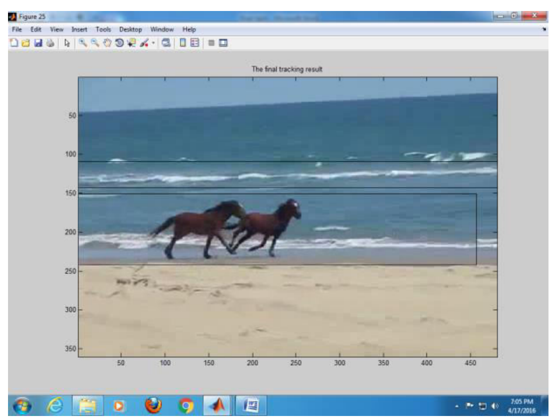
RGB frame110



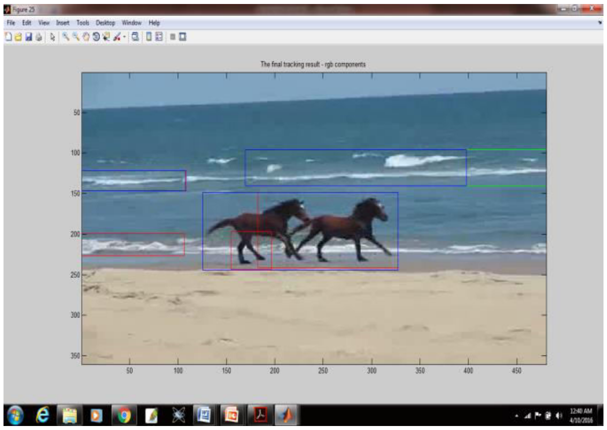
Moving Object in Frame110



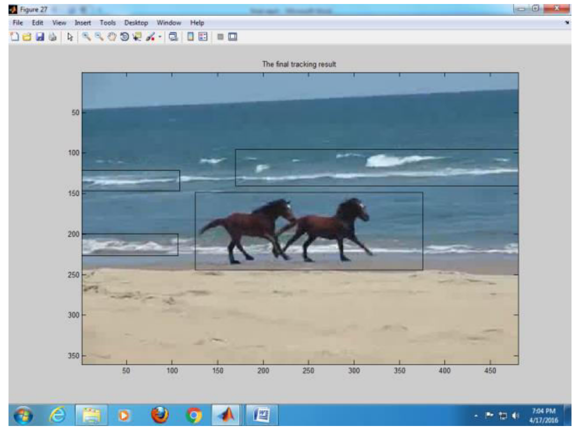
RGB frame120



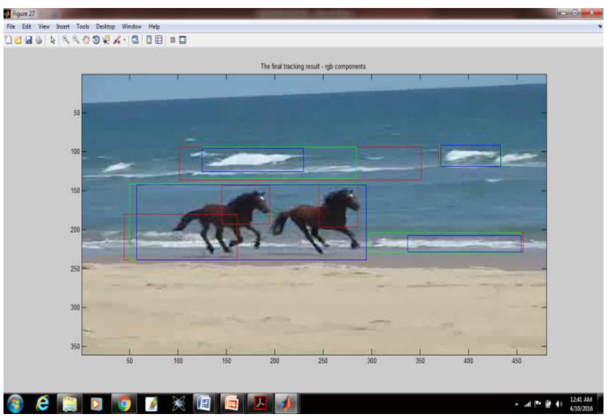
Moving Object in Frame120



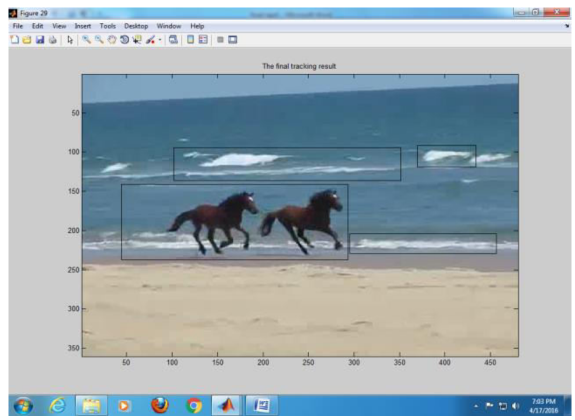
RGB frame130



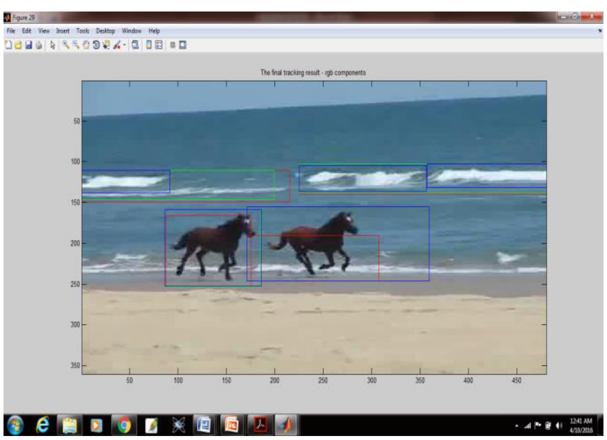
Moving Object in Frame130



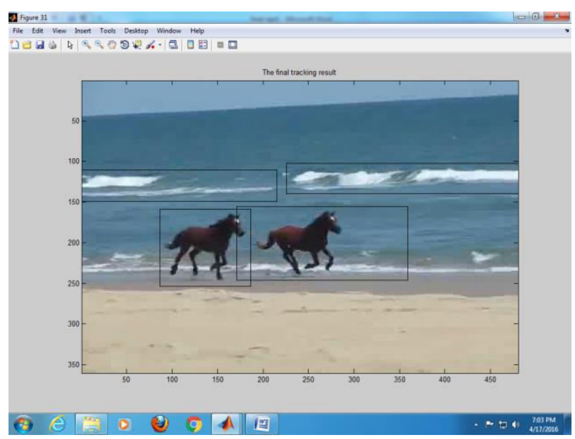
RGB frame140



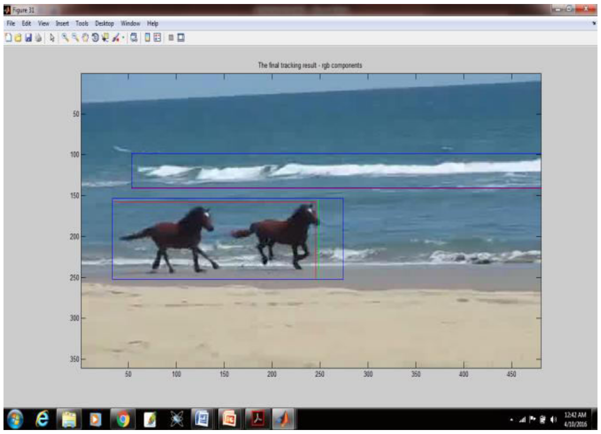
Moving Object in Frame140



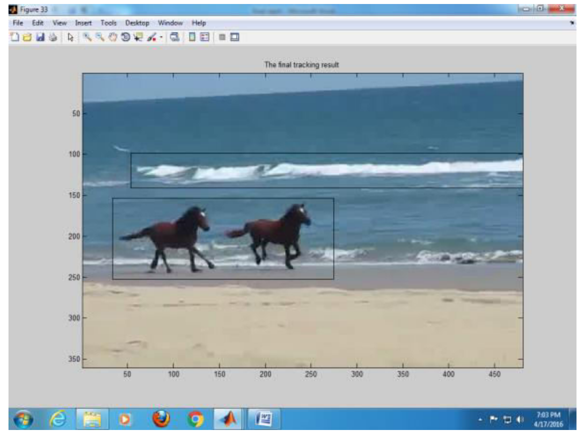
RGB frame150



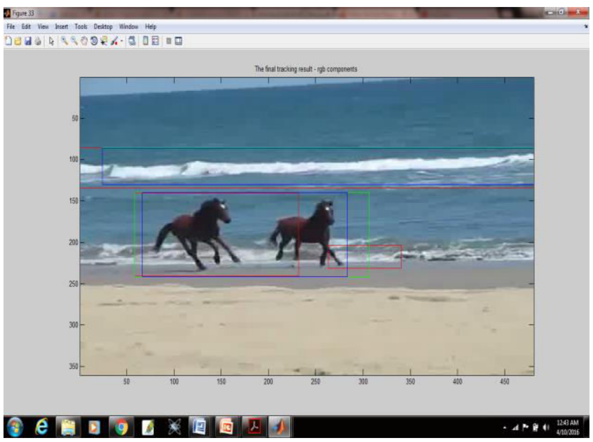
Moving Object in Frame150



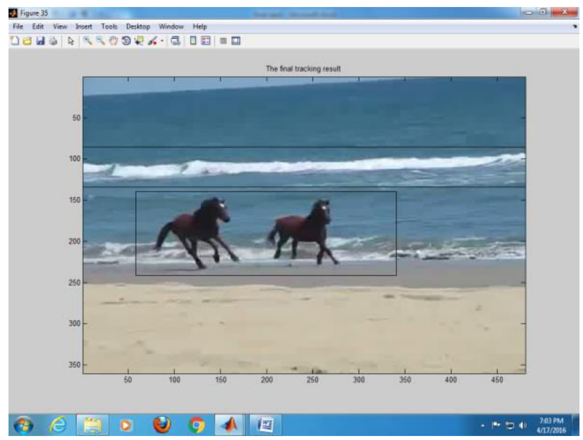
RGB frame160



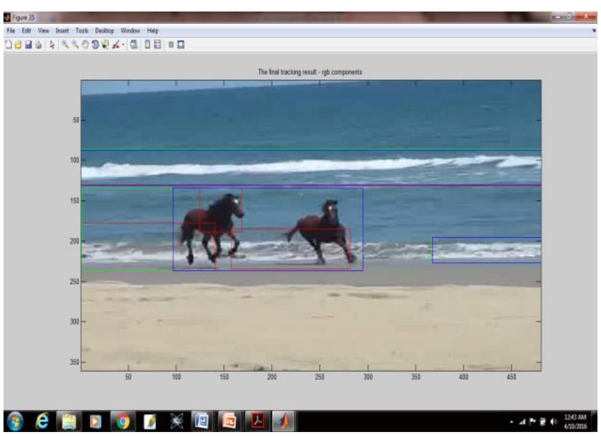
Moving Object in Frame160



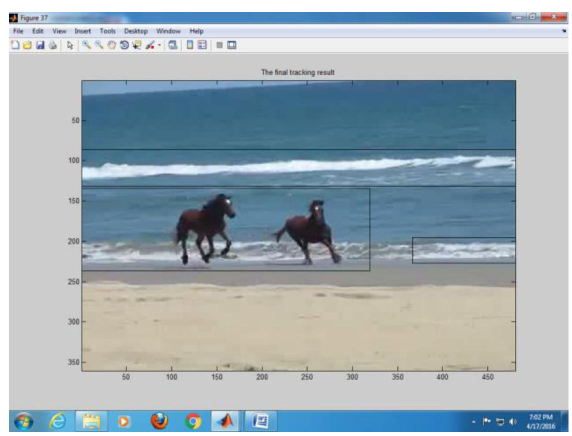
RGB frame170



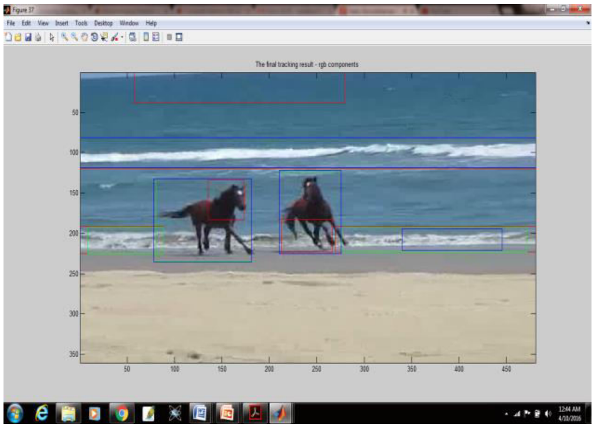
Moving Object in Frame170



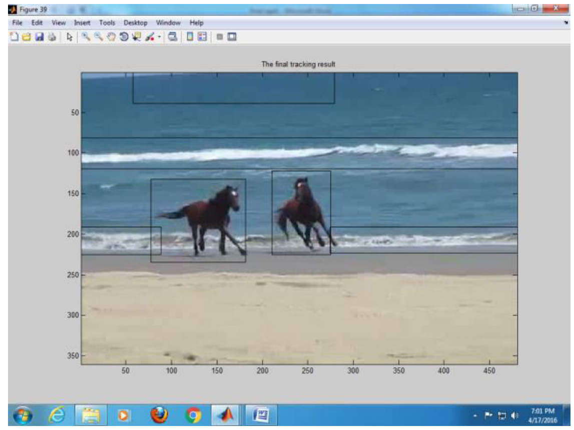
RGB frame180



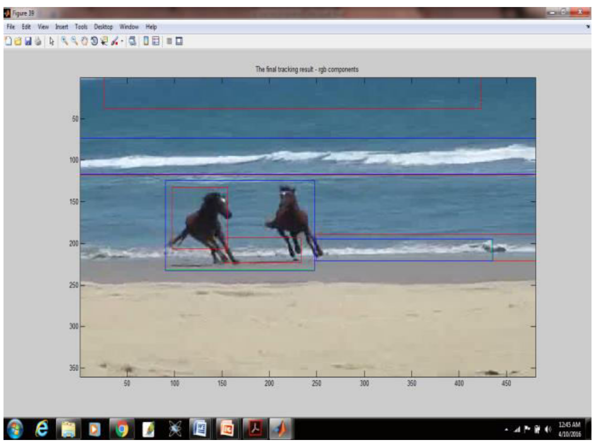
Moving Object in Frame180



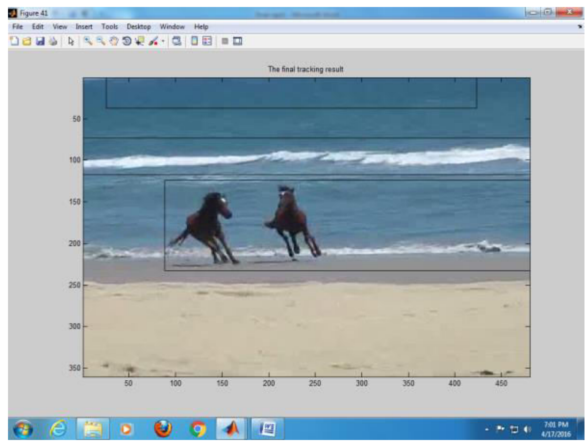
RGB frame190



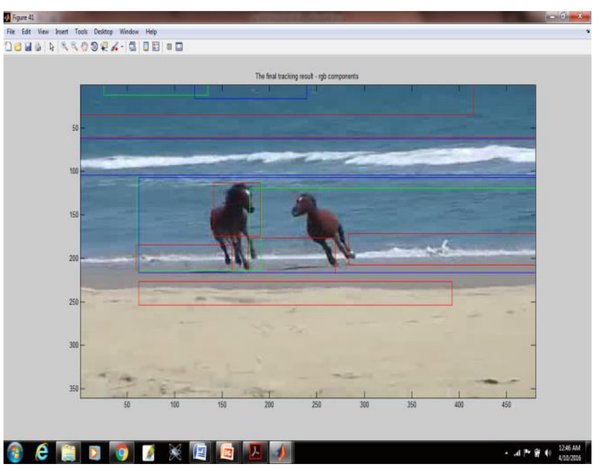
Moving Object in Frame190



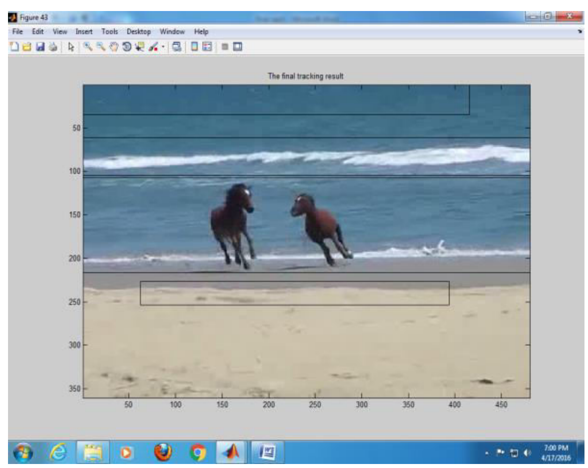
RGB frame200



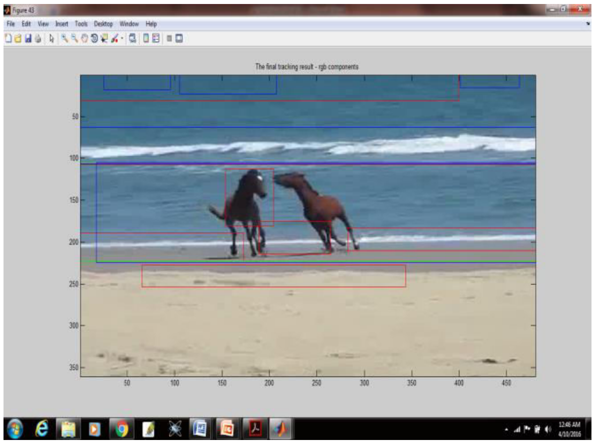
Moving Object in Frame200



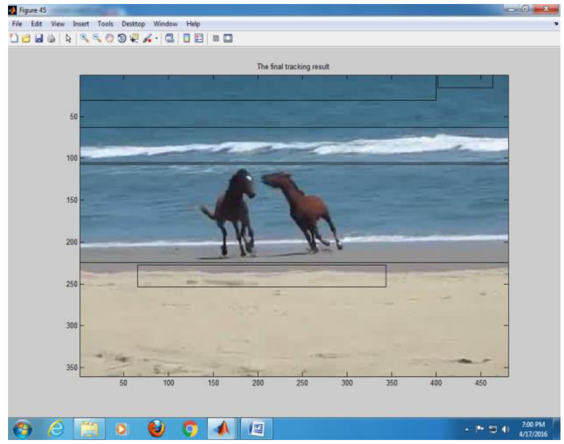
RGB frame210



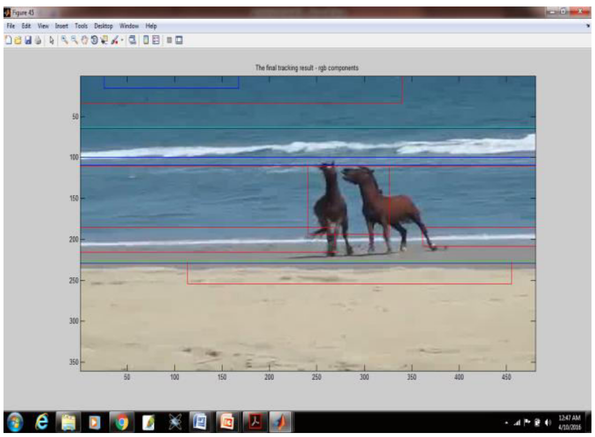
Moving Object in Frame210



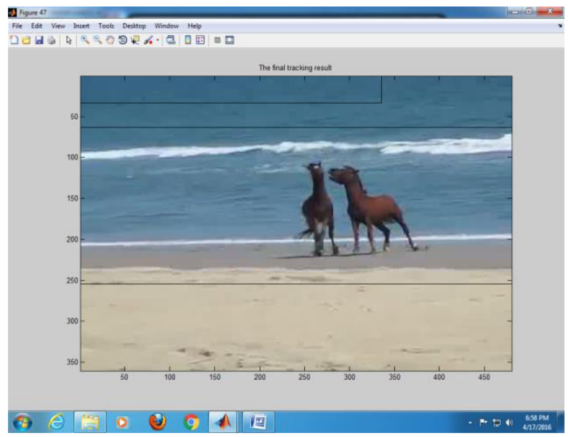
RGB frame220



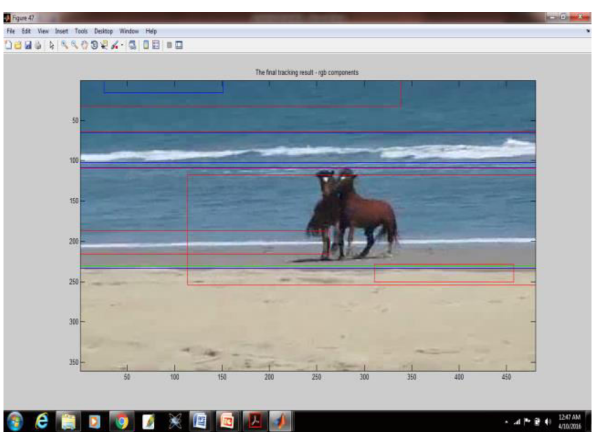
Moving Object in Frame220



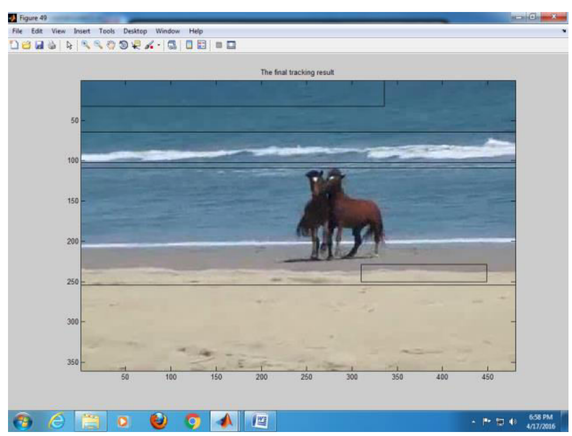
RGB frame230



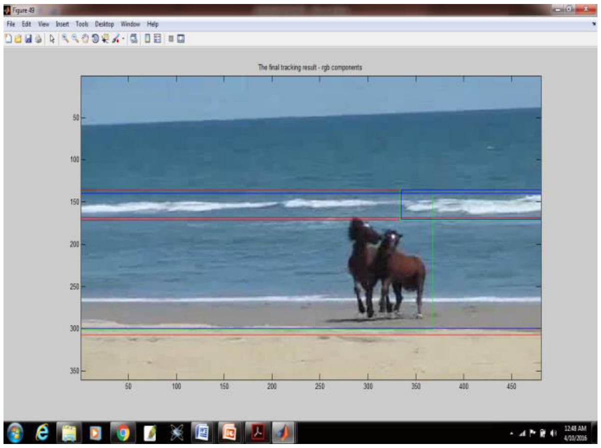
Moving Object in Frame230



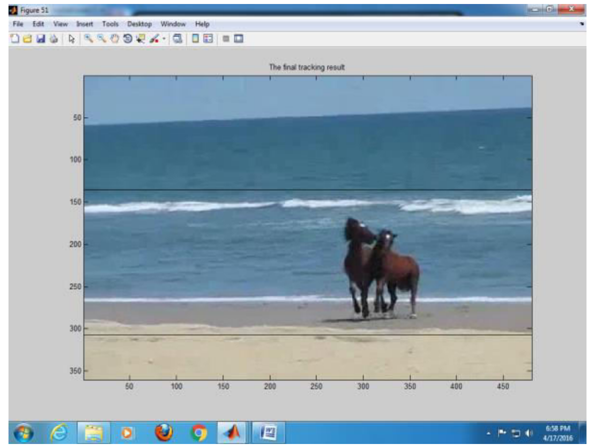
RGB frame240



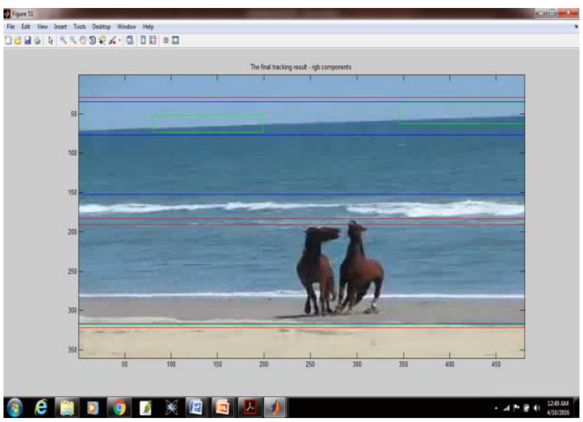
Moving Object in Frame240



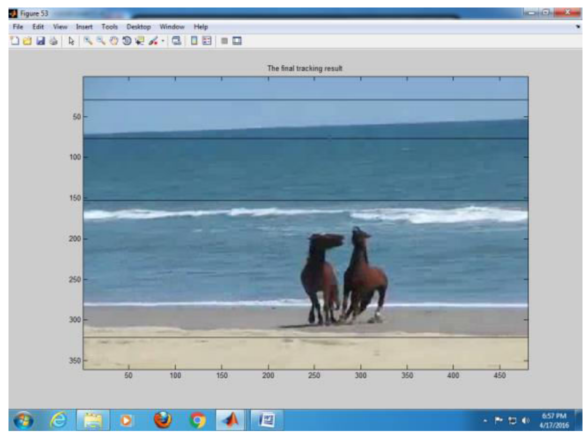
RGB frame250



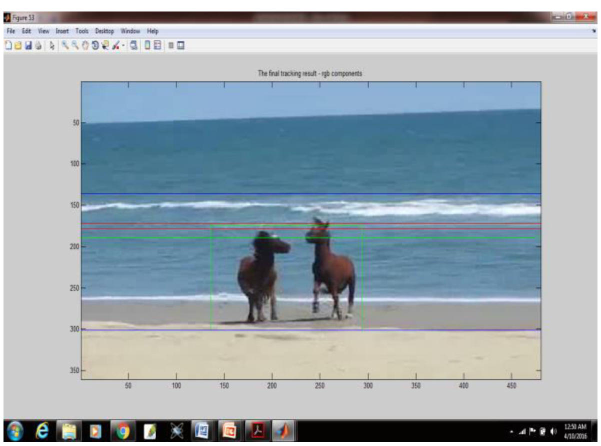
Moving Object in Frame250



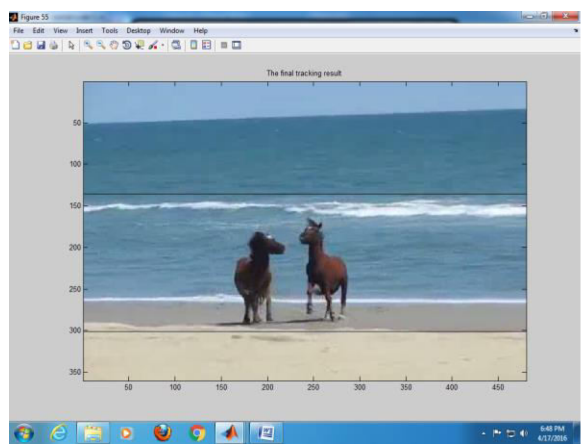
RGB frame260



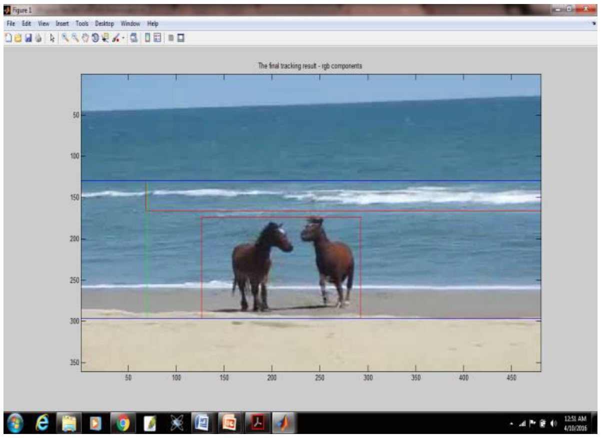
Moving Object in Frame260

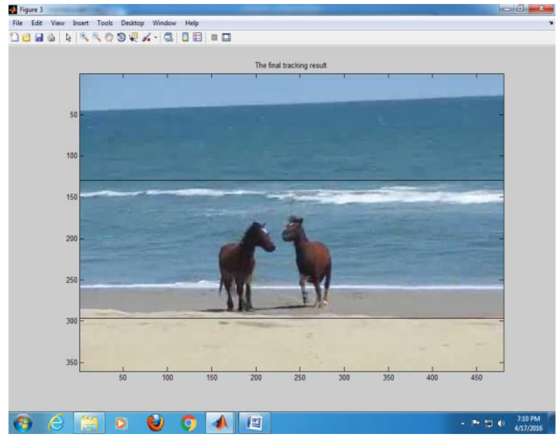


RGB frame270



Moving Object in Frame270





RGB frame273

Moving Object in Frame273

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

Our project explains how to track moving objects in a video. In this firstly, a video is acquired by video acquisition system which uses a high speed camera. Such, high frames per second containing video is send to a computer system using frame grabbers for further processing. In the system we write the code in MATLAB, where first the video is converted into frames depending on the speed of the processor in the computer since processing directly on the video would we difficult. Now an object detection method called **Background Subtraction** is performed. In this, first a background model is formed by considering the average intensity values in all the frames. Such background model formed is subtracted from all the other frames and difference images are formed. Those difference images are sent for thresholding to separate foreground and background pixels. In order to remove noise in the difference frame we have performed morphological filtering and finally blob analysis to detect and classify moving objects in the image. Background subtraction method detects moving objects by subtracting the background model from the current image. Here we implemented a method suitable for sudden or gradual illumination changes by improving the traditional method of background subtraction by considering all the frames for background modeling. Compared to other common moving object detection algorithms, background subtraction is simplest; segments foreground objects more accurately and detect foreground objects even if they are motionless.

Thus we conclude that our proposed algorithm is best suitable for object tracking within small areas and large field of view. This is suitable for the real time surveillance system because it has fast computation and is robust against environmental disturbances.

7.2 FUTURE SCOPE

The use of background model with mean of all intensities in all the frames makes our project suitable for even small sudden illumination changes.

In order to improve tracking, a Background model should be adaptive to the background dynamics such as large illumination changes, shadows and if background is non-static (e.g., waving trees, swaying curtains, escalators, rippling water surfaces, etc.).

So to improve the background model, each pixel is to be updated according to the difference of the background model and the previous frame for more accuracy; this can be done by implementing learning rate in the algorithm.

Also for effectively updating background model adaptively in dynamic scenes, unlike methods using same learning rate for entire frame or sequence, a method can be proposed for adaptive learning rate depending on mainly two parameters.

The first parameter depends on the difference between the pixel intensities of the background model and the current frame. The second parameter depends on the duration of the pixel being classified as a background pixel.

Tracking can also be improved by implementing an algorithm to determine the threshold automatically and dynamically depending on the pixel intensities of the current frame.

Object tracking may also be improved by implementing it using various filtering methods like kalman filtering, particle filtering and multiple hypothesis tracking.

Advance study may also be carried out to find efficient algorithm to reduce computational cost and to decrease the time required for tracking the object for variety of videos containing diversified characteristics.

7.3 REFERENCES

- http://www.ijircce.com/upload/2014/february/7J_A%20Survey.pdf
- https://www.researchgate.net/publication/258710427_A_Real-Time_Method_to_Detect_and_Track_Moving_Objects_DATMO_from_Unmanned_Aerial_Vehicles_UAVs_Using_a_Single_Camera
- http://www.mdpi.com/2072-4292/4/4/1090
- http://marmota.dlsi.uji.es/WebBIB/papers/2001/Badenas-PR-2001.pdf
- http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5941596&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5941596
- https://www.purdue.edu/discoverypark/vaccine/assets/pdfs/publications/pdf/Object%20Tracking%20Initialization%20Using%20Automatic%20Moving.pdf
- http://spie.org/Publications/Proceedings/Paper/10.1117/12.872610
- http://research.microsoft.com/en-us/um/people/jckrumm/Publications/WallFlower/final%20published.pdf
- http://spie.org/Publications/Journal/10.1117/1.JEI.24.4.043011

7.4 BIBLIOGRAPHY

- Digital Image Processing, 2nd edition by Rafael C. Gonzalez and Richard E. Woods
- Digital Image Processing, 2nd edition by Vipul Singh

APPENDIX

CODE:

```
clc
clear
close all

%% read video
readerobj = mmreader('video3.avi', 'tag', 'myreader1');

% Read all video frames
vidFrames = read(readerobj);

% Get the number of frames.
numFrames = get(readerobj, 'numberOfFrames')
height = get(readerobj, 'Height')
width = get(readerobj, 'Width')

% Create a MATLAB video struct from the video frames
for k = 1:numFrames
    mov(k).cdata = vidFrames(:,:,:,k);
    mov(k).colormap = [];
    figure
    imagesc(mov(k).cdata);
end

%% Computing the background
  [MR,MC,Dim] = size(mov(1).cdata);
  Imback = zeros(MR,MC,Dim);

for i = 1:numFrames
    Imback = double(mov(i).cdata)+ Imback;
end

Imback =  Imback/numFrames;
figure
imagesc(uint8(Imback));
title('Computed background');

%% Background Subtraction
 for i=1:numFrames
      thr=50;
      Diff1=uint8(abs(double(mov(i).cdata)-double(Imback)));
      figure
```

```
    imagesc(uint8(Diff1));
    title('Difference image');

    DiffR=Diff1(:,:,1)>thr;
    DiffG=Diff1(:,:,2)>thr;
    DiffB=Diff1(:,:,3)>thr;
    DiffT=DiffR+DiffG+DiffB;
    figure
    imagesc(uint8(DiffT));
    title('Thresholded Image');

% bwareaopen removes small objects from binary image.
  DiffR=bwareaopen(DiffR,300,8);
  DiffG=bwareaopen(DiffG,300,8);
  DiffB=bwareaopen(DiffB,300,8);
  DiffT=bwareaopen(DiffT,300,8);

% performing morphological processing using dilation function
  SE=strel('disk',8);
  DiffR=imdilate(DiffR,SE);
  DiffG=imdilate(DiffG,SE);
  DiffB=imdilate(DiffB,SE);
  DiffT=imdilate(DiffT,SE);
  figure
  imagesc(uint8(DiffT));
  title('Image after morphological filtering');

%% Blob Analysis

% bwconncomp finds connected components in binary image.
 CCR=bwconncomp(DiffR);
 CCG=bwconncomp(DiffG);
 CCB=bwconncomp(DiffB);
 CCT=bwconncomp(DiffT);

% regionprops Measure properties of image regions.
 SR=regionprops(CCR,'BoundingBox');
 SG=regionprops(CCG,'BoundingBox');
 SB=regionprops(CCB,'BoundingBox');
 ST=regionprops(CCT,'BoundingBox');


  figure
  imagesc(mov(i).cdata);
  title('The final tracking result - rgb components')
```

```
for k=1:numel(SR)
     BB1=SR(k).BoundingBox;
      rectangle('position',[BB1(1), BB1(2), BB1(3), BB1(4)],...
              'EdgeColor','r');
 end
 for k=1:numel(SG)
      BB2=SG(k).BoundingBox;
      rectangle('position',[BB2(1), BB2(2), BB2(3), BB2(4)],...
                'EdgeColor','g');
 end
 for k=1:numel(SB)
      BB3=SB(k).BoundingBox;
      rectangle('position',[BB3(1), BB3(2), BB3(3), BB3(4)],...
                'EdgeColor','b');
 end

  figure
  imagesc(mov(i).cdata);
  title('The Final Tracking Result - moving object');

% choosing the bounding box for highly connected pixels

  for k=1:numel(ST)
       BB=ST(k).BoundingBox;

%choosing the bounding box to be rectangle and with yellow color boundary
      rectangle('position',[BB(1), BB(2), BB(3), BB(4)],...
            'EdgeColor','k');
 end

% for continuous tracking
   pause(0.1)

end
```