Moving Target Detection and Tracking

Book · May 2015

Javed Iqbal
Information Technology University of the Punjab

Asim Ul Haq
Abasyn University

Moving target detection and tracking with camera in real-time is a hot field of research in the current era. Due to the advancement in computer, it is need of the day to make machine's perception, vision and intelligence like humans. The goal of this book is to present a research and real-time implementation of moving target detection, tracking and locking system. Many algorithms are reviewed and the latest and efficient ones are implemented. Hope that this work will help many researches who want to do research in the field of detection and tracking in real-time scenarios.

Javed Iqbal

Asim Ul Haq

Said Wali

Moving Target Detection and Tracking

Algorithms and Hardware System Real-time Implementation

Javed received B.E (Electrical Engineering) from CIIT, Islamabad, Pakistan in 2013. He is pursuing his MS in Electrical (control and automation) Engineering from the same institute. He was Research Associate at RCR Group, CIIT, Islamabad in 2013. Currently Working as Design Engineer at RENZYM (Pvt) Ltd.

978-3-659-69832-3

LAP LAMBERT Academic Publishing

Javed Iqbal
Asim Ul Haq
Said Wali

Moving Target Detection and Tracking

Javed Iqbal
Asim Ul Haq
Said Wali

Moving Target Detection and Tracking

Algorithms and Hardware System Real-time Implementation

LAP LAMBERT Academic Publishing

Impressum / Imprint
Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek: The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Coverbild / Cover image: www.ingimage.com

Verlag / Publisher:
LAP LAMBERT Academic Publishing
ist ein Imprint der / is a trademark of
OmniScriptum GmbH & Co. KG
Heinrich-Böcking-Str. 6-8, 66121 Saarbrücken, Deutschland / Germany
Email: info@lap-publishing.com

Herstellung: siehe letzte Seite /
Printed at: see last page
ISBN: 978-3-659-69832-3

Zugl. / Approved by: Pakistan, COMSATS Institute of Information Technology, Islamabad. Diss. 2013

Preface

The field of computer vision and image processing is concerned with the query that how to enable machines that automatically do the tasks in the domain of computer vision like humans. In recent years many systems are developed that are doing the same task using different things, like radar, ladar, sensors etc. By comparing with humans it is still lacking because human have a very sensitive and accurate vision system.

The goal of this book is to present a research and real-time implementation which shows the path that it is possible to achieve machine having vision system approaching humans. Automation and control draws the concepts and results from the fields of artificial intelligence, image processing, signal processing, statistics, probability and distribution, information theory and machine learning.

Because of the embedded nature of the problem, this research makes some assumptions about the reader of the book. Instead of introducing basic concepts of the filed involve in the research it only focusses from application point of view. The research book is intended for both undergrad and graduate students who want to pursue their career in the domain of image processing, computer vision, artificial intelligence, machine learning, robotic vision, control and automation.

Acknowledgements and Dedication

First of all we would like to thank Almighty ALLAH for giving us the courage and persistence to achieve our goals. We are extremely thankful to sir Mustafa Pasha for his continued supervision and guidance for entire duration of the project. We are much thankful to Dr. Jamshed Iqbal, Dr. Nadeem Javaid, Dr. Naveed-Ur-Rehman and Mr. Abdussamad for their extended cooperation throughout the project. We are also very thankful to project lab staff specially Mr. Adeel for their continued support during the research work.

We dedicate our work to our beloved Prophet Hazrat Muhammad (PBUH), our parents and teachers whom prayers, courage and wishes have brought us here, to our brothers, sisters and friends whose love, courage, support and prayers able us to be here and finally to our supervisor who were very supporting for us.

Javed Iqbal

Asim Ul Haq

Said Wali

Table of Contents

List of Abbreviations and Acronyms

OCR……………………………………………..……..……Output Capture Register

PWM…………………………….………...….……..…..Pulse Width Modulation

FM…………………………………………….….……….Frame Differencing

TM……………………...…………….....................................Template Matching

MS………………………………………….....................................Mean-Shift

SIFT…...………………...….............................. Scale Invariant Features Transform

List of Figures

List of Tables

Abstract

Today's challenging Problem is security systems. Computer vision plays a vital role in security system. In computer vision real time video analysis using camera has its own importance. Moving target detection, tracking and locking is an important application of video processing and control system. It has great importance in military, sports, traffic and many other applications. Moving target detection and tracking manually with camera platform requires constant attention of humans, makes the Tracking task difficult, time consuming and erroneous. In this Report a complete simulation interface and Hardware platform is described which first detects moving object, track it and lock it. For detection and tracking two algorithms are used i.e. Frame Differencing (FM) algorithm and Mean-Shift algorithm, and locking is achieved using serial communication between matlab and proteus. In hardware implementation live video is captured through camera, moving objects are detected and tracked using computer processor and locked through hardware. The complete system is tested upon recorded videos and real time videos. The results show that the system has good efficiency for pedestrians walking on normal speed and ground vehicles at some acceptable speed.

Chapter-01

Introduction

1.1 Overview

Today's security systems are fast and efficient, offer very efficient monitoring system. This is possible because of self/computer operating systems, and the devices used for security systems are working very efficiently. But the problem with today's security system is that the operator some time loss the target to whom the operator want to cover and monitor constantly. We work on this project to overcome this problem up to some extent, and trying to make the whole system autonomous.

This system can be used for the security system, traffic monitoring, and detection of unmanned air vehicles (UAVs) as well as for ground vehicles. This system can also be implemented in sports activities, i.e. for human's motion detection, football detection and lots of other things.

1.2 Brief Introduction

Moving target detection and tracking is widely used for security purposes. Here a platform is presented which performs the required task of moving object detection, tracking and locking to make the whole system autonomous. First task is to detect the moving object in an image or frame, secondly find centroid of the detected object and track the detected moving object using centroid information. In the last step locking of moving object is achieved.

The system contains one Camera, two Servo motors (pan-tilt mechanism), one Microcontroller board, one Laser light (used as a pointer), and computer processor.

Camera is fixed in hardware platform. Pan-Tilt mechanism is fixed below camera and Laser light is mounted on Pan-Tilt system, which will point out the moving object. The computer processor is used for processing of image processing algorithms. Real time video stream is provided to computer by Camera. Algorithms for detection and tracking are written in Matlab. For locking of moving object Matlab is interfaced with AVR microcontroller board through serial communication.

In this system the detection and tracking are taken out through software. After detection and tracking of moving object, the object is locked. As a result where ever object will move within the camera view, laser mounted on Pan-Tilt system will follow the tracked object to lock and thus makes the whole system autonomous. An overview is shown in figure1.1.

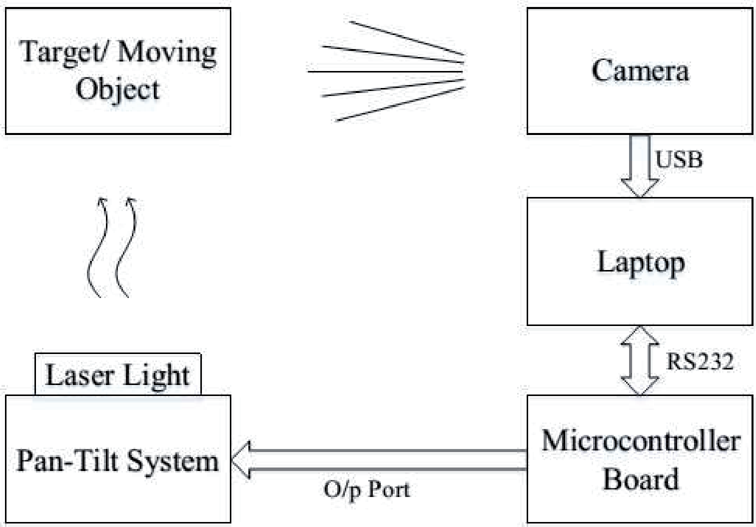


Figure 1.1: Block diagram describes the project overview.

1.3 Related Work

In the field of Image and Video processing most of the people want to work on motion detection and tracking system, identification system and recognition systems. As our project is on motion detection tracking and locking system, therefore we will discuss here just motion detection and tracking.

To achieve detection and tracking, there are many techniques/algorithms which have their own pros and cons. Some of the important algorithms are:

1. Frame Differencing algorithm [1],[2]
2. Background Subtraction algorithm[3]
3. Optical flow value algorithm[4]
4. Mean-Shift algorithm[5]
5. Template Matching algorithm[6]
6. Cellular neural network algorithm[7]
7. Kanade-Lucas-Tomasi Algorithm[8]
8. Kalman filtering[9],[10]

Some of the above algorithms are used for detection and some are used for tracking purposes. Algorithms like Frame differencing, Background subtraction, Optical flow, and cellular neural networks are generally used for detection purposes. Other algorithms like Kanade-Lucas-Tomasi, Kalman filtering, Template Matching and Mean-shift algorithms are used for tracking of moving objects. Frame differencing can be used for both detection and tracking purposes.

After the detection and tracking of moving object the next step is locking. For locking of the detected object there are lots of approaches depending upon the scenarios. Some of them are very expensive like implementation on DSP/FPGA kits. Some of the methods are not much expensive, for example the computer processor is used for detection and tracking and locking is done through microcontroller. As we all know microcontrollers are not much expensive as DSP/FPGA kits are. So the selection for hardware implementation totally depends upon the requirements as well as costing.

1.4 Project Background

Automatic moving object detection in real time scenario is very difficult task. There are lots of approaches used for Detection and Tracking in real time scenarios. We choose frame differencing algorithm for detection and tracking of moving object in real time implementation. Implementation of this algorithm is done in Matlab. This project can also be done with many other algorithms but we choose Frame Differencing algorithm [1]. This algorithm is very easy to implement and have no such complex things which make this algorithm very difficult. The reasons behind the choosing of this algorithm are: this algorithm is easy to implement, both detection and tracking can be achieved with this simple algorithm. After the successful detection and tracking of moving object we achieved locking with help of hardware platform.

1.4.1 Image

An image is two dimensional plane view of captured scene. The capturing of an image is the projection of real time scenarios of two dimensional planes taken out through a camera. The color on the single pixel is detected on the basis of reflection of color from real time three dimensional scenes. There are different layers in an image. Image is basically the combination of three layers. These layers are:

1. Saturation layer
2. Hue layer
3. Intensity layer or Grey scale layer

The value of each layer ranges 0-255 and in this manner the colors on an image is found.

There are different types of images. The normal images that we used in daily life are RGB images. Binary image and Gray scale image or intensity images are also the types of images. We are using RGB, gray level and binary images throughout our project.

1.4.2 Video

Video is the sequence of images. Video is produced from the two dimensional images sequence comes after one another respectively. Real time scenarios are three dimensional. So capturing of image is the conversion of three dimension scenes to two dimension or we can also say that the image is the projection of three dimensions scenes onto two dimension plane. In real time video the three dimensional scenes are converted to two dimensional and projected to the video camera plane and is changed one after another. The single image in video is called frame. The minimum frame rate in video is 20 frames per second. If the frame rate is less than 20 the frames act like static images.

1.4.3 Pixel

Pixel is the smallest unit of image. Image is made up of small dots or units which are responsible for resolution of an image and color intensities. The color in an image is determined by the value of the pixel present on a specific location. The value of the pixel varies from 0-255. In binary images the pixel value varies between [1-2].

1.4.4 Moving Object Detection

Moving object detection is a difficult task and real time detection in real time scenarios makes the task more complex. There are different techniques used for the detection of moving objects. As mentioned above Optical flow method, frame differencing, background Subtraction [3] and cellular neural networks and many other algorithms are used for the detection of moving object in real time video.

Frame differencing technique is implemented for detection of moving object in real time scenario. This algorithm is not very difficult to implement and the results obtained from this algorithm are very good.

1.4.5 Challenges in Object Detection

The 1st challenge in this project was how to detect a moving object in real time scenario. Instead of going on real time video processing first detection of moving object is achieved on recorded videos. Then the second problem was which algorithm should be used for the detection of moving objects. Frame differencing and mean-shift algorithms were selected for accomplishing the objectives.

1.4.6 Tracking of Moving Object

After successful detection of the moving object, tracking of the moving object is achieved by finding the central coordinates of detected object. The object is enclosed in bounding box and the path is plotted. Figure 1.2 shows the trajectory of the tracked object.

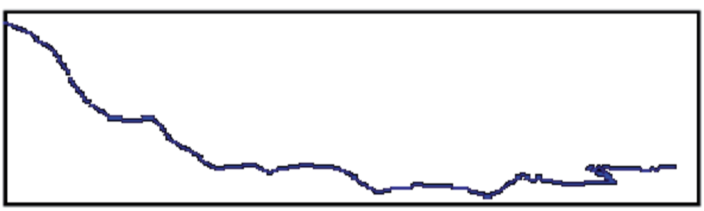


Figure 1.2: Trajectory of Moving object

1.4.7 Locking of Moving Object

Locking of the detected and tracked object is a difficult task. There are different methods used for locking of detected object. The central coordinates found in tracking are used to find angles of deviation of moving object from reference point.

These angles are passed to microcontroller board through serial communication which in turns moves motors. Laser light mounted on Pan-Tilt system point outs the tracked object.

1.5 Methodology

We divided our project in two sections; the first one that we choose to be done was the software design. After completion of the software design we then start the hardware section of our project.

1.5.1 Software

When we start the software design, the first problem was the selection of algorithm that can be implemented. The second problem was the source and compiler. There were two choices, C/C++ and Matlab, so we choose Matlab software due to efficient simulation results.

Processing Model

The code has been implemented in Matlab 2011a. First of all camera has been interfaced and the colors and size is fixed for Video coming from video camera. RGB color is used to display the normal video and 320x240 size is used. As mentioned above we choose the frame differencing algorithm to implement. The flow and steps of frame differencing algorithm is shown in figure 1.3:

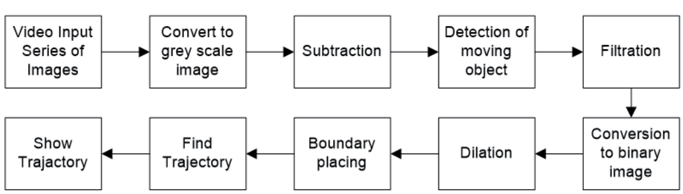


Figure 1.3: Operational flow diagram of Frame differencing Algorithm.

This operational flow diagram of Frame differencing algorithm will be discussed in detail in later chapters.

1.5.2 Hardware

Hardware is made for the locking of the detected object. As written in introduction the laser fixed on motors (Pan-tilt system) will follow the detected object.

Processing Model

After the successful detection of moving object, we then find the co-ordinates and then find the angles of translation. Those angles are then passed to Microcontroller through serial communication. The simulation of hardware was done in Proteus. In Proteus a model of this system was designed, for serial communication virtual serial port software is used. After the complete simulation process the operational flow diagram of the whole system is shown in figure 1.4.

Why Microcontroller

At the time of implementation we had two choices, Implementation on DSP kit and Implementation on Microcontroller. We choose microcontroller for implementation. DSP kit is very much expensive as compared to a microcontroller. That's why we use the microcontroller (Atmega-16) platform.

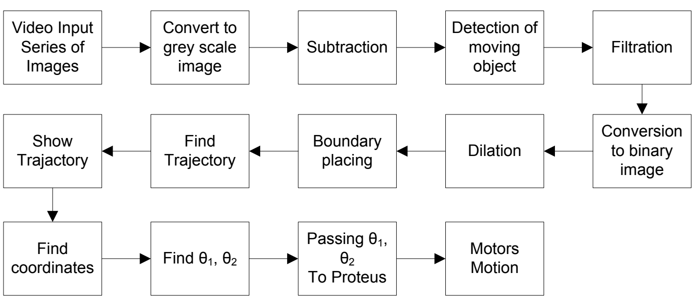


Figure 1.4: Operational flow diagram of Design and Implementation of Moving Target Detection, Tracking and Locking System.

1.6 Scope of Project

The system (Moving Target Detection, Tracking and Locking System) is capable of successful detection of moving object within the range of the video camera fixed on the frame. And can successfully find the track of the detected object. When Detection and Tracking is done the system can lock the object and will follow the detected object. The Detected object is pointed out by a laser fixed on the motors (Pan and Tilt System) and the laser capable of following the detected object.

1.7 Assumption and Dependencies

Nothing in this universe is complete. Everything present in this universe has their pros and cons. This system has also some dependencies and constraint. Some constraints are assumed to be not present or fixed.

1.7.1 Assumptions

It is assumed that the system background is fixed and has no shadow effect. The system will treat different moving objects in same frames as a one object.

1.7.2 Dependencies

The system depends on the lighting conditions and luminance. The system is also dependent on the environmental conditions.

Chapter-02

Literature Review

Moving target detection and tracking is an important task in video surveillance systems and is point of focus since 1960s. Many techniques, methods and algorithms are described in this field having its own importance. This chapter mainly briefly describes some techniques of moving target detection and tracking separately and then shows a comparison on the basis of different factors, properties, conditions and limitations.

2.1 Detection Algorithms

Detection of moving target is very important task in analysis of moving objects. To efficiently and accurately detect moving objects three basic methods are more popular. (a) Frame differencing algorithm (b) Background Subtraction Method and (c) Optical flow technique. Each of the three methods is briefly described below.

2.1.1 Frame Differencing Algorithm

Frame Differencing (FD) algorithm is widely used for motion analysis. It has very efficient results in detection of moving objects. FD algorithm depends upon reference frame and current frame, threshold and illumination changes. It gives centroid (center coordinates) of moving object and area of moving object as output. Using centroid information, we calculate trajectory of the desired tracked moving object. Angles of motors' motion relative to center position of the frame are calculated to lock the tracked object.

FD algorithm use previous frame as reference frame and frame at time (t) as current frame. These frames are input to FD algorithm. Difference between reference frame and candidate frame or current frame is taken. Moving objects are detected reprocessed and passed to tracking function to track moving object. After detection of moving object the resultant image is reprocessed to clarify the detected object. Reprocessing contains filtering, thresholding, morphological operations like corrosion is applied. Then the object coordinates are determined and passed to tracking function.

2.1.2 Background Subtraction Algorithm

Background Subtraction (BS) algorithm is widely used for moving object detection and is very efficient technique under conditions of static background and static camera. BS algorithm has its own importance and applications in motion analysis, Human Computer Interaction (HCI) and intelligent control systems. In motion analysis the most important part is detection of moving target. The objective is to separate the desired moving object from background in video and use it for some operations like segmentation, recognition and tracking. BS algorithm is based on dynamic background modeling and dynamic thresholding. BS technique is very sensitive to changes in external environment like effect of light and other background clutters.

The BS algorithm uses the technique of FD [1] for detection of moving objects. The previous frame with some dynamic modeling acts as background for the current frame. Taking difference of the two frames shows the moving objects. Then applying some operation i.e. morphological operations to refine the image and clearly show the detected target. It is very sensitive to background changes that’s why dynamic background modeling is used to overcome this effect up to some extent. The ideal condition is when the background is static means that working under static camera conditions. Further dynamic thresholding selection method based on the background subtraction is combined with dynamic background modeling on the basis of accurate detection of object to improve the results and enhance the effect of moving object detection.

2.1.3 Optical Flow Method

Optical flow method describes the Pixel Motion Process in which optical flow is adopted to obtain the motion information between every two adjacent frames. According to the optical flow theory: optical flow is based on three assumptions and those three assumptions are:

1. Grey-value assumption
2. Gradient assumption
3. Smoothness assumption

The smoothness assumption is used in place where the first two assumptions fail. The global deviations from grey value constancy assumptions and the gradient constancy assumption are measured by energy:

$$E_{\text{Data}\,(u,\,v)} = \int_{\Omega} (\Psi|\Delta I|^2 + r|\Delta G|^2)\, dx \tag{2.1}$$

$$\Delta I = I(\mathrm{x} + \mathrm{w}) - I(\mathrm{x}) \tag{2.2}$$

$$\nabla G = \nabla I(\mathrm{x} + \mathrm{w}) - \nabla I(\mathrm{x}) \tag{2.3}$$

Where $\mathrm{x}=(x, y, t)^T$ and $\mathrm{w}=(u, v, 1)^T$, r is a weight between both assumption, $\Psi(s^2)$ is an increasing concave function that leads to a robust energy.

$$E_{\text{Smooth}\,(u,\,v)} = \int_{\Omega} (\Psi|\nabla_3 u|^2 + r|\nabla_3 v|^2)\, dx \tag{2.4}$$

$$E_{(u,\,v)} = E_{\text{Data}} + \alpha E_{\text{Smooth}} \tag{2.5}$$

Pixel Process and Pixel Motion Process[4]

When the camera and background are both static, then pixel values changes as light condition changes. In this condition the pixel value is changing but this change is constant and constantly changing with respect to time. These changes are called "pixel process". Pixel Process is modeled using multiple and adaptive Gaussian by Stauffer and Grimson which leads to a stable background of the scene. When camera and background both are in motion, the pixel value is changing over time and this change is not constant so there is no regular pattern, in this situation it is not possible to calculate directly “pixel process”. These changes are considered as “pixel motion process”. In this situation background motion is modeled instead of modeling the background.

2.2 Tracking Algorithms

Tracking of moving object is an important task during analysis of moving objects. To efficiently track moving bodies many algorithms are defined. (1) Template Matching tracking algorithm. (2) Mean-Shift. (3) Scale Invariant Feature Transform [11],[12]. (4) Kanade-Lucas-Tomasi tracker and. (5) Kalman Filter. All the mentioned algorithms are described briefly below.

2.2.1 Template Matching Algorithm

The Template Matching (TM) is an efficient and simple algorithm for object tracking that we want to search. TM algorithm takes sequences of frames as input and searches the desired specified object on the basis of maximum correlation. In the first frame a template of the desired object is selected and it is correlated with the second frame. If any where it has maximum correlation it shows the message that the object is matched, otherwise goes to next frame.

TM algorithm works as tracker module. It can be used to track still object as well as moving objects. Only to detect interesting moving objects TM algorithm is used because it recognizes and detects respective specified object irrespective of motion or static condition of other objects. The template specified is known as target template which is selected having some previous information. The previous information may be a predefined template or selected on the basis of centroid or mean coordinates of desired object. In next upcoming frames target template is matched with coming frames. By using mathematical correlation if it matches somewhere, then the template is updated with current matched template. Correlation is done in bi-level or gray level images for which it is necessary that we should convert our image to that specified format. The matching process moves the target template to all possible positions in a larger source image because match is done on a pixel by pixel basis and computes a numerical index that indicates how the template matches the image in that position. The trajectory of the tracked object can be plotted.

In gray level mathematical correlation can be found using the equation,

$$cor = \frac{\sum_{i=0}^{N-1} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-1} (x_i - \bar{x})^2 \cdot \sum_{i=0}^{N-1} (y_i - \bar{y})^2}} \tag{2.6}$$

Where,

x is the template gray level image

x is the average grey level in the template (target) image

y is the source image section

y is the average grey level in the source image

N is the number of pixels in the section image

(N= template image size = columns * rows)

The value cor in equation above is between –1 and +1, with larger values representing a stronger relationship between the two images.

For using in real time scenario TM algorithm has some limitations, i.e.

1. Target template should be entirely located in source image.

2. Partial template matching was not accomplished due to boundaries.

3. Variation and scaling will cause pitiable matches

2.2.2 Mean-Shift Algorithm

Mean-Shift (MS) algorithm is widely used for moving object tracking and is an efficient technique for moving object tracking. It is based on the statistical calculations of color in the target model and candidate model. The similarity is measured using Bhattacharyya coefficients. The new position of the target is calculated on the basis of similarity function. MS algorithm is useful for clustering, mode seeking, probability density estimation and tracking.

MS is an algorithm that iteratively shifts a data point to the average of data points in its neighborhood. It uses the color histogram information and statistical calculations to locate the new position of the object.

MS algorithm uses Kernel function K(x), that helps in the estimation of mean. Typically, kernel K is a function of $||x||^2$:

$$K(x)= k(||x||^2)\ ) \tag{2.7}$$

Where k is called the profile of K and have the properties, (1) k is nonnegative (2) k is non-increasing. Commonly used kernels are: Flat kernel, Gaussian kernel and Epanechnikov kernel.

The simple mean m at point x with kernel K(x) is calculated as

$$m(x) = \frac{\sum_{i=1}^{n} K(x-x_i)x_i}{\sum_{i=1}^{n} K(x-x_i)}$$

$$m(x) = \frac{\sum_{i=1}^{n} K(x-x_i)x_i}{\sum_{i=1}^{n} K(x-x_i)} \tag{2.8}$$

And the difference m(x)-x is called the mean-shift.

The MS tracking based on color histogram is divided into three basic steps. In the first step probability density function (PDF) of the target model and the candidate model is calculated on the basis of color histogram and kernel density estimation. In the second step similarity between the target model and the candidate model is calculated. For this purpose Bhattacharyya coefficient is calculated. Finally, the new position of the object in the candidate model is calculated in neighborhood of previous target position with the help of similarity measure and object is tracked in the candidate model.

2.2.3 Scale Invariant Feature Transform Algorithm

Tracking of objects in a video is an important task in video analysis applications such as computer vision, surveillance etc. Here a feature extraction and matching method is applied i.e. scale Invariant Feature Transform (SIFT). SIFT extracts features of object which are invariant to changes in object like illumination, scaling and orientation changes etc, and continuously detects and matches the features.

SIFT algorithm extracts the features points in the target model and then matches it with features points in the candidate model. SIFT also calculates location of features points and matches it with the location of features points in the candidate model on the basis of Euclidian distance.

Generations of Image Features

SIFT have following major stages of computations used to generate the set of image features:

1. Scale-space extrema detection.
2. Keypoint localization.
3. Orientation assignment.
4. Keypoint descriptor.

After extracting the feature points and matching with initial template in the current frame by using SIFT algorithm, centroid of area surrounded by these feature points can be calculated, the calculation formula is:

$$M = \frac{\sum m_i r_i}{\sum m_i} \tag{2.9}$$

Where m_i is pixels, r_i is pixel coordinate position.

SIFT is an algorithm to achieve a stable multiple object tracking using the concept of feature points matched on the basis of location matching. Object tracking is performed by a rectangle window around the desired object at the reference frame and the current frame. The key concept is to find out the feature points and exclude those, which are location mismatched.

Initially, rectangular windows around objects in the reference frame and candidate frame are drawn. Keypoints are generated using SIFT and their local features are stored in a database. Then from the next consecutive frame, keypoints are generated by the same process. Secondly, matching keypoints are selected on the basis of Euclidian distance. These SIFT matched keypoints become candidates for tracking and among these candidates the location matched keypoints are determined. Finally, based on the location matched keypoints, a stable tracking for multiple objects is achieved.

2.2.4 Kanade-Lucas -Tomasi Algorithm

Kanade-Lucas-Tomasi (KLT) algorithm is widely used for moving object tracking. KLT is based on features which are tracked frame to frame. Moving objects can be tracked by correlating the features because the features are distributed all over the image. The features belonging to moving objects will be clustered and assign to the corresponding moving object motion layer. The first step is detection of moving objects and construction of motion layer. Then the maintenance of all moving object motion layers keeps all moving objects tracked. Features lost during tracking are replaced by newly detected good features to maintain the number of features. Consecutive two frames are used as control points to register a frame to its previous frames. In order to separate background features we propose a region division method.

2.2.5 Kalman Filter

Kalman filter is a powerful technique for tracking moving objects. It was originally invented by Rudolf Kalman at NASA to track the trajectory of spacecraft. Kalman filter is basically is a set of mathematical equations which provide an effective computational solution to successive systems. The filter is very powerful in several aspects: it supports estimation of past, present, and future states (prediction), and it can do so even when the precise nature of the modeled system is unknown. The state model for the target motion is defined by the following vector-matrix equations.

$$x_{\mathbf{k}} = \mathbf{A}x_{k-1} + \mathbf{B}u_k + w_{k-1} \tag{2.10}$$

$$z_{\mathbf{k}} = \mathbf{H}x_k + v_k \tag{2.11}$$

where $x=[x\ y\ x'\ y']^T$ is the state vector, y is the output, and wand v are white noises denoting the process and the measurement noises, respectively, all with appropriate dimensions. Moreover, the transition matrix and the observation vector are

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{2.12}$$

The equations for the Kalman filter fall into two groups: the time update equations and the measurement update equations. The Time Update or “Predict” step projects the current state estimate ahead in time to obtain 'a priori' estimate for the next time step. The Measurement Update or "Correct" step adjusts the projected estimate by an actual measurement at that time. The measurement update equations are responsible for the feedback. For incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate.

2.3 Comparison of Algorithms

In this section different detection and tracking algorithms are compared on the basis of their properties, efficiency, specifications and limitations. A brief over view of detection and tracking algorithms is given below.

2.3.1 Detection Algorithms

Moving target detection is very important task in analysis of moving bodies. To accurately and efficiently detect moving targets three basic methods are more popular. (1) Frame Differencing algorithm (2) Background Subtraction algorithm and (3) Optical flow method. All the above mentioned algorithms have its own usage, importance, specifications, level of complexity and limitations. Frame differencing algorithm has good detection results but is very sensitive to light intensity variations. Background subtraction algorithm have good detection results under static camera conditions and very sensitive to background changes. Optical flow method is an efficient detection method under ideal conditions but in real time scenario it requires large calculations, very sensitive to noise and anti-noise performance is very poor. All the factors and outcomes are shown in table 2.1.

2.3.2 Tracking Algorithms

While analyzing moving object tracking is an important task to be achieved. To accurately track moving targets many algorithms are defined. A comparison is done on the basis of some algorithms studied. (1) Template Matching (2) Mean-Shift Tracking algorithm (3) SIFT Tracking technique (4) KLT tracker (5) Affine Transformation and (6) Kalman Filtering. Every one of the mentioned algorithms has its own importance, properties and limitations. Template matching uses mathematical correlation between the target template and the source image. TM has good tracking results in case if the target template is completely present in the source image. Mean-Shift achieves tracking using kernel function and Bhattacharyya coefficients to calculate similarity measure and have good tracking results. MS algorithm can also be used for detection of slow speed moving objects. SIFT find outs features that are invariant to scaling, rotations and light conditions but is very difficult to implement specially in real time. KLT tracker is dependent upon optical flow field of the image but very sensitive to noise and requires a large quantity of calculations. A brief comparison of tracking algorithms is shown in table 2.1.

Table 2.1: Comparison of Different Detection and Tracking Techniques

	Name of the Technique	Detection	Tracking	Dependencies	Constraints	Efficiency	Complexity	Preference	Date
1	Frame Differencing	Yes	Yes	Reference and current frame, threshold, illumination	Multiple objects, Shadow effects	Better detection and poor tracking	Simple implementation	Gives center Position of object	2012
2	Template Matching	No	Yes	Threshold, correlation of templates	Rapid shape change, Illumination change, shadow effects	Tracking efficient	Simple Implementation	Good tracking result, object trajectory	2012
3	Background Subtraction	Yes	No	Dynamic background modeling, thresholding	Static camera, change in external environment	Detection efficient	Normal implementation	Better detection result, object position	2012
4	Mean-shift	Yes	Yes	Color information, similarity Measure, kernel function	Poor detection result, no spatial information, Illumination changes	Tracking efficient, poor detection	Complex implementation	Position of the object, better tracking results, object trajectory	2012
5	Scale Invariant Feature Transform	No	Yes	Feature points (local maxima and minima etc.), Euclidian distance	Background keypoints, shadow effects	Efficient tracking compared to above techniques	Complex Implementation	Best tracking result, invariant to change (scale, illumination, environment etc.)	2012

Chapter-03

Algorithms Implementation

Analysis of moving bodies in real time scenario is a difficult task to achieve. Many techniques are proposed in this domain as some of them are briefly described in chapter (2). On the basis of comparison done in chapter (2) we select Frame Differencing algorithm and Mean-Shift algorithm for detection and tracking purposes to implement on recorded video and real time scenarios. Implementation of both the algorithms is described in detail below.

3.1 Frame Differencing Algorithm

3.1.1 Introduction

Frame differencing algorithm is widely used for motion analysis. It has very efficient results in detection of moving objects. Frame differencing algorithm depends upon reference frame and current frame, threshold and illumination changes. It gives centroid (center coordinates) of moving object and area of moving object as output. Using centroid information, we calculate trajectory of the desired tracked moving object. Angles of motors' motion relative to center position of the frame are calculated to lock the tracked object.

3.1.2 Frame Differencing Algorithm Description

Frame differencing use previous frame i.e. frame at time (t-1) as reference frame and frame at time (t) as current frame. These frames are input to FM algorithm. Difference between reference frame and candidate frame or current frame is taken. Moving objects are detected and passed to tracking function to track moving object.

Frame differencing algorithm is implemented on predefined video and real time video. Area of the frame is $w \times h$, where w is width and h is height of the frame, (x_0, y_0) is center of the frame. Flowchart of algorithm is shown in figure (1). Following steps are involved in frame differencing algorithm when applied on predefined video,

Step1. Load video to the program.

Step2. Extract all frames from video.

Step3. Initialize current frame and previous frame.

Step4. Convert frames to gray scale and apply filter to remove noise.

Step5. Subtract the previous frame from current frame.

Step6. Convert resultant frame to binary.

Step7. Apply morphological operations i.e. dilation, corrosion etc. to refine the image.

Step8. Find area and centroid of the tracked object.

Step9. Enclose the desired object in bounding box.

Step10. Plot the trajectory of the tracked object.

When applying on real time video change the first two steps as follows,

Step1. Capture video from camera.

Step2. Separate frames and pass to the algorithm.

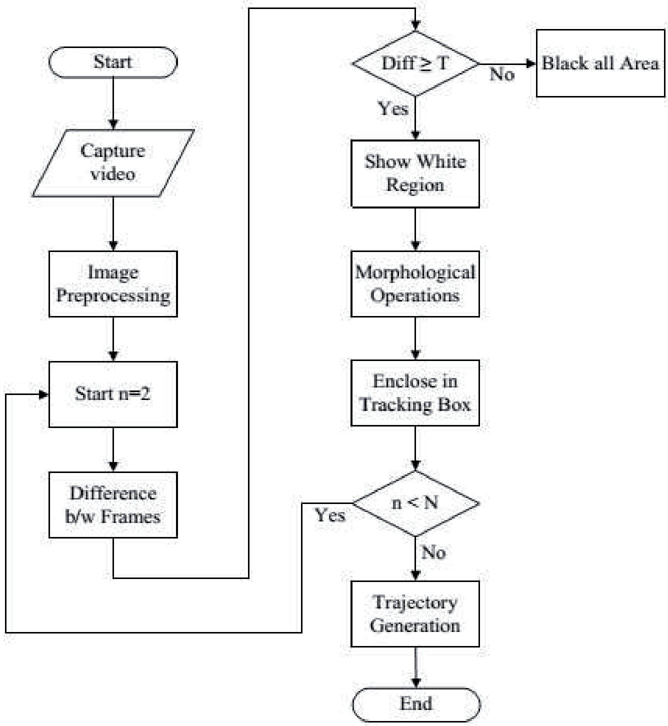


Figure 3.1: Flowchart of Frame Differencing algorithm

3.1.3 Object Detection

Detection of moving object is achieved using Frame differencing algorithm. In this algorithm (for both predefined video and real time video) we take two consecutive frames and compare them having some threshold. Objects which have changed their position with respect to previous frame will treat as moving objects.

Using frame differencing algorithm for detection first step is to take predefined video or capture video from camera and separate frames. For real time video capturing matlab image acquisition toolbox is used. Then convert RGB images to gray scale images. RGB image is composed of three layers i.e. hue layer, saturation layer and grey layer. The conversion to grey scale means the removing of hue and saturation layers. Secondly find the difference between two consecutive frames i.e. previous frame at time (t-1) and current frame at time (t) for detection of moving object. The difference is taken by following equation,

$$D = I_n - I_{n-1} \tag{3.1}$$

$$D = I_n - I_{n-1}$$

Where D is the difference of two images, I_n is the current frame at time (t) and I_{n-1} is the previous frame at time (t-1). Resultant image obtained from difference operation is then converted to binary image using the threshold level,

$$B = \begin{cases} 1 & D(x,y) \geq T \\ 0 & elsewhere \end{cases} \tag{3.2}$$

$$B = \begin{cases} 1 & D(x,y) \geq T \\ 0 & elsewhere \end{cases}$$

Where B is binary image, $D(x, y)$ is the difference at pixel (x, y) and T is threshold value. Here T=0.095 (9.5%) found by head and trial method and is very close to theoretical (10%). Pixels having values greater or equal to threshold will be turned white or moving object and those having value less than threshold will be turned black and will be considered as background.

3.1.4 Reprocessing

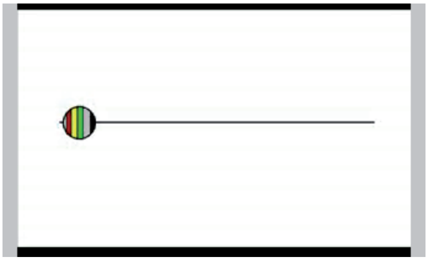
The difference image obtained contains motion regions, in addition, also a large number of noises and unwanted regions. Therefore, noises need to be removed. To remove noise we adopted a median filter of size 3x3. If we increase window size it decreases operation time at the cost of decreasing efficiency. Filter removes noise up to some acceptable extent. After the median filter, in addition, some unwanted objects are still there which is to be removed. For this purpose morphological operation i.e. corrosion, dilation etc. are performed. Dilation is applied to fill in the holes and gapes in the detected object keeping some threshold level. Corrosion is applied to remove some pixels values from edges which have increased due to dilation. After applying morphological operations the resultant image is good fair and indicates the detected object clearly.

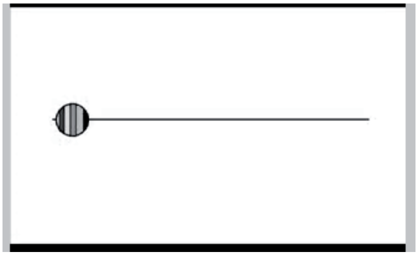
3.1.5 Object Tracking

Image obtained from reprocessing is then passed to tracking module. Two methods can be applied to track the object. First is using regionprops tool of image processing toolbox, find out number of detected objects, their areas and centroids information. Enclose the detected object in bounding box and plot trajectory using centroid information of the desired object. Second is to find out all the maxima's and minima's of the image axes. Then draw lines around detected object. Find out the center coordinates and area of the detected object for further processing.

3.1.6 Stepwise Implementation and results

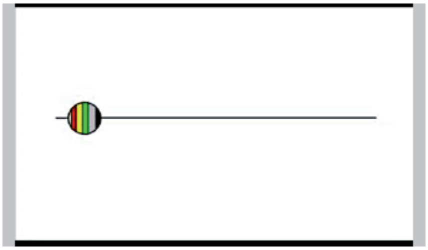
The stepwise implementation results of recorded video are shown in figure 3.2. First step is to load video. Then extract all frames, initialize current frame and previous frame as shown in figure (a) and (c). Convert previous frame and current frame to gray level images as shown in (b) and (d). Then take the difference of two gray level images as shown in (e). (f) Show's the image converted to black-white on the basis of some threshold. (g) Show's the reprocessed refined image and. (h) Show's the tracked object (the detected object is in closed in bounding box).

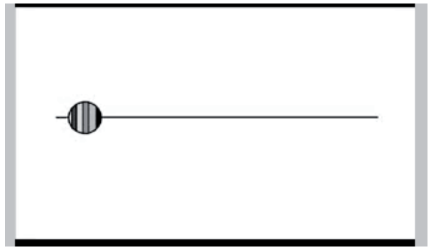




(a) First Frame of Movie

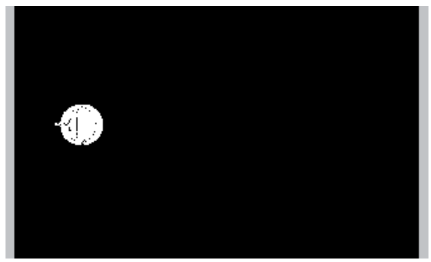
(b) Gray level Image of First Frame





(c) 2^{nd} Frame of Movie

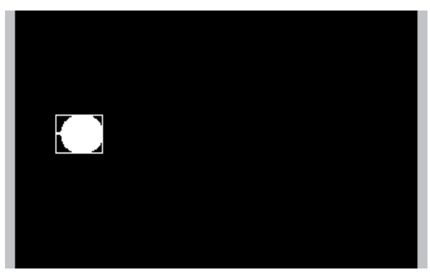
(d) Gray level Image of First Frame



(e) Difference Frame (gray level)

(f) Difference Frame (BW)

(g)



Refined Image

(h) Image Showing Tracked Object

Figure 3.2: This figure shows the stepwise processing results of Frame Differencing algorithm. (a) Shows the first frame of recorded video. (b) Shows the gray level of first frame. (c) Shows the

second frame of recorded video. (d) Shows the gray level of second frame. (e) Shows the difference of two frames in gray level. (f) Shows the difference image in binary level. (g) Shows the reprocessed refined image. (h) Finally shows the tracked object.

3.2 Mean-Shift Algorithm

3.2.1 Introduction

Mean shift algorithm is widely used for moving object tracking and is an efficient technique for moving object tracking. It is based on the statistical calculations of color in the target model and the candidate model. The similarity is measured using Bhattacharyya coefficients. The new position of the target is calculated on the basis of similarity function. Mean shift algorithm is useful for clustering, mode seeking, probability density estimation and tracking.

3.2.2 The mean shift algorithm

Mean shift is an algorithm that iteratively shifts a data point to the average of data points in its neighborhood. It uses the color histogram information and statistical calculations to locate the new position of the object.

Mean shift algorithm uses Kernel Function K(x), which helps in the estimation of mean. Typically, kernel K is a function of $||x||^2$:

$$K(x) = k(||x||^2) \tag{3.3}$$

Where k is called the profile of K and have the properties, (1) k is nonnegative (2) k is non-increasing. Commonly used kernels are: Flat kernel, Gaussian kernel and Epanechnikov kernel. Flat kernel is represented as

$$K(x) = \begin{cases} 1 & if\ ||x||^2 < 1 \\ 0 & otherwise \end{cases}$$

$$K(x) = \begin{cases} 1 & if\ ||x||^2 < 1 \\ 0 & otherwise \end{cases} \tag{3.4}$$

And Gaussian kernel is represented as

$$K(x) = \frac{1}{\sqrt{(2\pi)^d}} e^{\frac{1}{2}(||x||^2)}$$ $$K(x) = \frac{1}{\sqrt{(2\pi)^d}} e^{\frac{1}{2}(||x||^2)} \quad (3.5)$$

Where d represent the dimensionality. The distributions of the Flat kernel an Gaussian kernel are shown in figure 3.3.

The simple mean m at point x with kernel K(x) is calculated as

$$m(x) = \frac{\sum_{i=1}^{n} K(x-x_i)x_i}{\sum_{i=1}^{n} K(x-x_i)}$$ $$m(x) = \frac{\sum_{i=1}^{n} K(x-x_i)x_i}{\sum_{i=1}^{n} K(x-x_i)} \quad (3.6)$$

and the difference m(x)-x is called the mean shift.

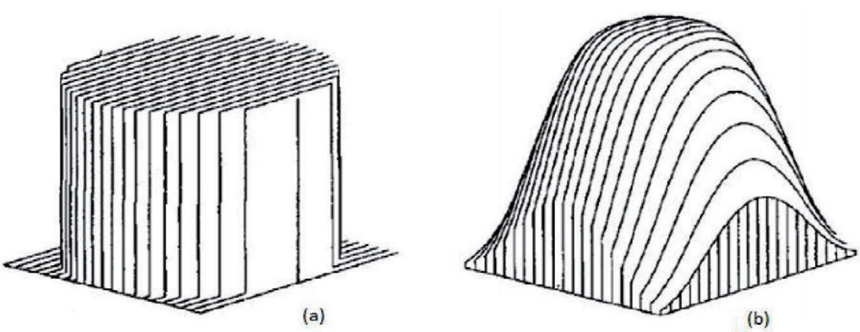


Figure 3.3: (a) Flat Kernel (b) Gaussian Kernel

3.2.3 Mean shift tracking

The mean shift tracking based on color histogram is divided into three basic steps. In the first step probability density function (PDF) of the target model q_u and the candidate model p_u is calculated on the basis of color histogram and kernel density estimation.

$$q_u(x) = C_q \sum_{i=1}^{n} k\left(\left\|\frac{x_i}{h_q}\right\|^2\right) \delta(b(x_i) - u) \quad u = 1,2, \dots, m$$

$$q_u(x) = C_q \sum_{i=1}^{n} k\left(\left\|\frac{x_i}{h_q}\right\|^2\right)\delta(b(x_i) - u) \quad u = 1,2,\ldots,m \tag{3.7}$$

$$p_u(y) = C_P \sum_{i=1}^{N_p} k\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)\delta(b(y_i) - u) \quad u = 1,2,\ldots,m$$

$$p_u(y) = C_P \sum_{i=1}^{N_p} k\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)\delta(b(y_i) - u) \quad u = 1,2,\ldots,m \tag{3.8}$$

Where $b(x_i)$ and $b(y_i)$ represent the histogram index function of pixels. h_p and h_q are the radius of the candidate and target model kernels respectively. δ is the Kronecker delta function

$$\delta(a) = \begin{cases} 1 & if\ a = 0 \\ 0 & otherwise \end{cases}$$

$$\delta(a) = \begin{cases} 1 & if\ a = 0 \\ 0 & otherwise \end{cases} \tag{3.9}$$

Thus, if $b(x_i)$=u the PDF will have some value otherwise kernel will contribute zero value to q_u and p_u. C_q and C_p are normalization constants and have values

$$C_q = \left[\sum_{i=1}^{n} k\left(\left\|\frac{x_i}{h_q}\right\|^2\right)\right]^{-1}$$

$$C_q = \left[\sum_{i=1}^{n} k\left(\left\|\frac{x_i}{h_q}\right\|^2\right)\right]^{-1} \tag{3.10}$$

$$C_P = \left[\sum_{i=1}^{N_p} k\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)\right]^{-1}$$

$$C_p = \left[\sum_{i=1}^{N_p} k\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)\right]^{-1} \tag{3.11}$$

In the second step similarity between the target model and the candidate model is calculated. For this purpose Bhattacharyya coefficient ρ is calculated. i.e.

$$\rho(y) = \rho[p(y), q] = \frac{1}{2}\sum_{u=1}^{m} \sqrt{p_u(y) q_u} \tag{3.12}$$

Let y denote current target location with color probability $\{p_u(y)\}$, $p_u(y) > 0$ for u = 1, . . . ,m. and let z denote estimated new target location near y, and color probability does not change rapidly. Now using taylor expansion (10) can be reduced to

$$\rho[p(z),q] = \sum_{u=1}^{m} \sqrt{p_u(y) q_u} \approx \frac{1}{2}\sum_{u=1}^{m} \sqrt{p_u(y) q_u} + \frac{1}{2}\sum_{i=1}^{N_p} w_i k\left(\left\|\frac{z-y_i}{h_p}\right\|^2\right)$$

$$\rho[p(z),q] = \sum_{u=1}^{m} \sqrt{p_u(y) q_u} \approx \frac{1}{2}\sum_{u=1}^{m} \sqrt{p_u(y) q_u} + \frac{1}{2}\sum_{i=1}^{N_p} w_i k\left(\left\|\frac{z-y_i}{h_p}\right\|^2\right) \quad (3.13)$$

where,

$$w_i = \sum_{u=1}^{m} \delta(b(y_i) - u) \sqrt{\frac{q_u}{p_u(y)}}$$

$$w_i = \sum_{u=1}^{m} \delta(b(y_i) - u) \sqrt{\frac{q_u}{p_u(y)}} \quad (3.14)$$

Finally, the new position 'z' of the object in the candidate model is calculated in the neighborhood of the previous target position y_i with the help of similarity measure and the object is tracked in the candidate model. The new position z is calculated as

$$z = \frac{\sum_{i=1}^{N_p} y_i w_i g\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)}{\sum_{i=1}^{N_p} w_i g\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)}$$

$$z = \frac{\sum_{i=1}^{N_p} y_i w_i g\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)}{\sum_{i=1}^{N_p} w_i g\left(\left\|\frac{y_i - y}{h_p}\right\|^2\right)} \quad (3.15)$$

Where, $g(x) = -k'(x)$.

Algorithm Description:

1. Initialize the first frame, the size of the frame is WxH with width 'W' and height 'H'.
2. Assume that the size of the bounding rectangle of the current object is W_0xH_0 .
3. Calculate the color histogram or PDF of the target template q_u .
4. Read the next frame image and calculate the candidate model PDF p_u.
5. Compute the similarity measure i.e. Bhattacharyya coefficient ρ(p(y), q), of the target model.
6. Compute the new target location.
7. Calculate the PDF of candidate model regarding to the new position and compute respective similarity measure, ρ(p(z), q).

8. If $\rho(p(z), q) < \rho(p(y), q)$, then
   $z = \frac{1}{2}(y + z)$.

9. If $||z - y|| < \varepsilon$ i.e. is small enough, then stop and goto step 4.

else, set $y \leftarrow z$ and goto step 4.

3.2.4 Mean-Shift Implementation

Firstly MS algorithm is implemented on the basis of simple mean. In this concept select a template and find out it simple mean. Then going to second frame and wider the area by 10 pixels each side using previous center coordinates. Then start searching window on that region and where the mean matched updated the template enclose the object in a box and moved to next frame. The flowchart of simple MS algorithm is shown in figure 3.4.

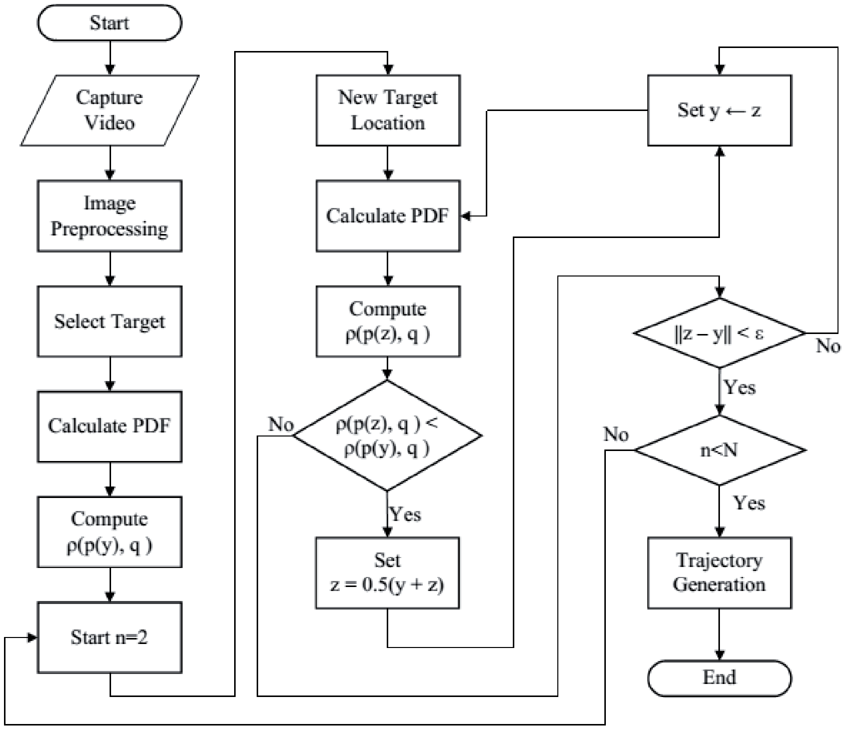


Figure 3.4: Simple MS algorithm flowchart

The results obtained from simple MS approach are acceptable but very sensitive to light changing effects. To avoid light effects and intensity limitations, implement MS algorithm in its traditional form. The results obtained are good compared to simple MS approach. MS algorithm is tested on recorded videos and also in real time scenarios. It has poor detection capability but good tracking results.

Chapter-04

Simulation Platform

In simulation platform a recorded video or live video from a camera is taken out for processing. This video (recorded/live) is then processed in matlab to detect moving object and find its location. The object position is found in the form of xy coordinates which is then converted into horizontal and vertical angles through which the target has been moved w.r.t to the reference point. The calculated angles are then transmitted to Proteus from matlab through virtual serial port. The microcontroller circuit in Proteus processes these angles and generate PWM signal according to the received angles. PWM is applied to servo to move servo motor through an angle received from matlab. This process is shown in figure 4.1.

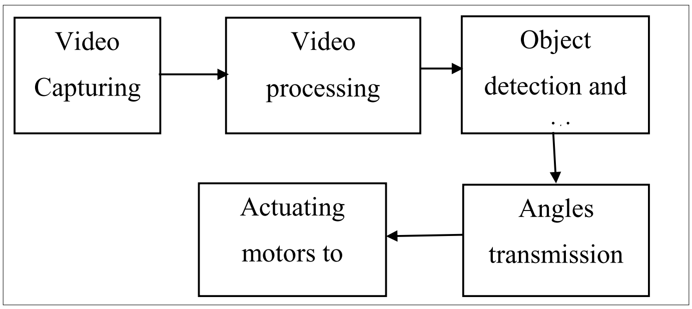


Figure 4.1: Block diagram of the system simulation

4.1 Main components

Before implementing the system on hardware, simulation of whole system has been performed.

In simulating the whole system following tools are used.

1. Matlab

2. Proteus
3. Virtual serial port driver.
4. AVR studio 5
5. Hyper terminal

4.1.1 Matlab

Matlab is a powerful tool for video processing as well as for performing computation. For simulation all the video processing work has been in matlab. Through video processing object is detected in live video and its position is calculated in xy coordinate system w.r.t to a reference point. x, y coordinates are converted into horizontal and vertical angles through the object has been moved. Serial communication with is established with microcontroller and calculated angles are transmitted through this connection. “Serial” command is used in matlab for establishing serial communication. Parameters such as baud rate, parity bits, number of bits transmitted in one frames and the time out etc. are also settle down using the serial command.

4.1.2 Proteus

Proteus is a power tool for simulating circuits. In simulating the system a microcontroller circuit has made in Proteus. This circuit includes a microcontroller (i.e. Atmega16), two servo motors, a compim and an oscilloscope. Microcontroller generates PWM signal for controlling servo motor angular position according to angles received from matlab. Complete schematic of circuit is shown in Figure 4.2.

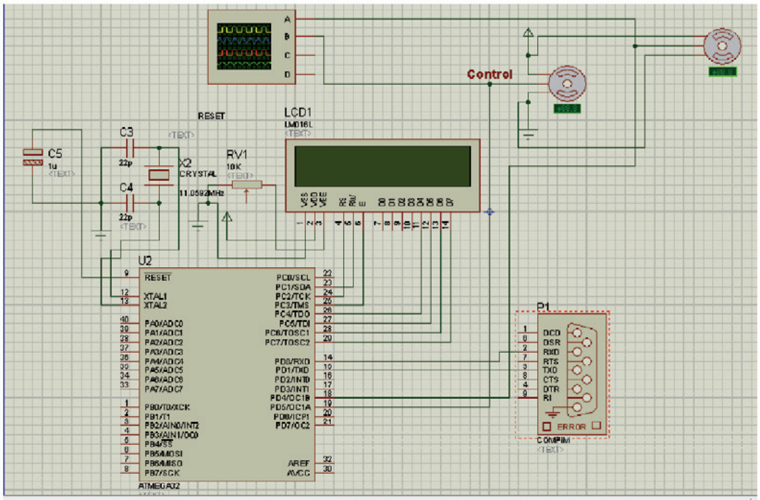


Figure 4.2: Schematic diagram for simulation in proteus

In the figure it can be seen that a compim is connected to microcontroller. This is used for establishing connection between hyper terminal of the computer or MATLAB and Proteus. Parameters for serial communication are specified by double clicking on the compim. Compim and parameters setting is shown in Figure 4.3.

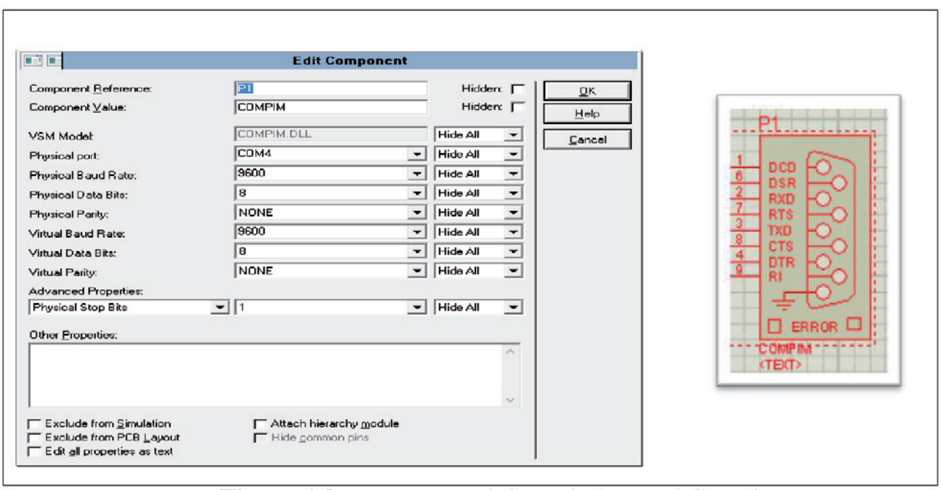


Figure 4.3: Parameters sitting window and Compim

Parameters setting on compim

Double click on the compim in proteus creates a window, shown in figure 4.2. Parameters values for serial communication are set to same values as that set in matlab serial command. Settings will be the same except for terminal selection. The virtual ports on the computer are always created in pairs e.g. COM3, COM4. From the pair one port is used for transmission and other for reception. Data transmission between matlab and microcontroller has been carried out using this pair.

4.1.3 Virtual Serial Port Driver

Virtual serial port driver is software that is used to create virtual ports on the computer.it create port in pairs as shown in figure 4.4. It also shows real serial ports of the computer. It's very simple and straight forward software. Just open up the software the following window appears.

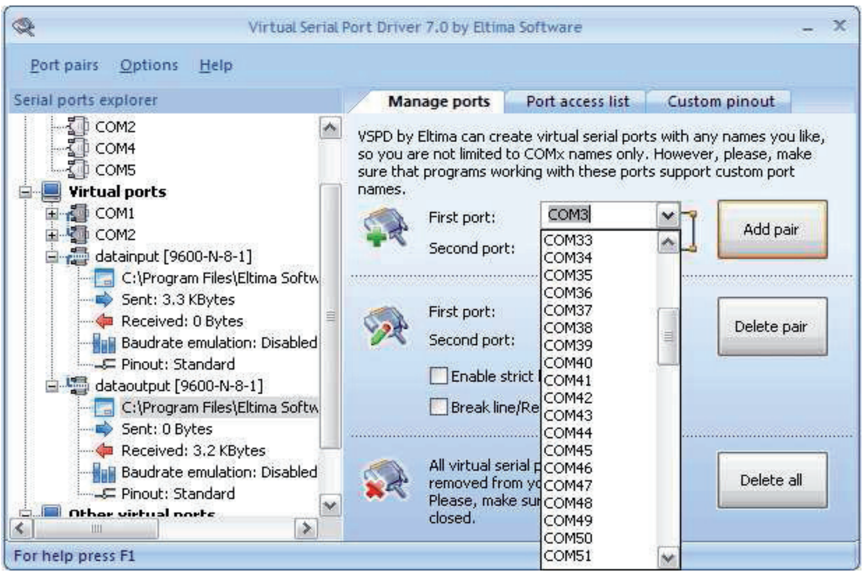


Figure 4.4: Virtual serial port driver

Just making a click on the add pair button the pair of port shown with add pair in the widow will created on your computer. After this another pair will appear in the window and if click on add pair button another pair will be added. The ports created can be checked from the device manager. This is shown in Figure 4.4.

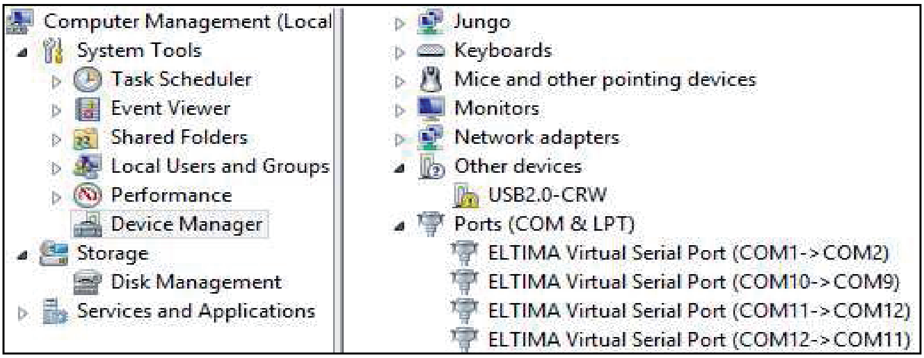


Figure 4.5: Virtual serial ports on computer

4.1.4 AVR Studio

AVR studio is a tool for programming AVR microcontroller. Its 5^{th} version is quite simple and user friendly. In AVR studio code of microcontroller for reception of data from matlab and generating PWM signal for servo motors control connected to microcontroller is written out.

4.1.5 HyperTerminal

In order to set serial communication between microcontroller and computer hyper terminal is used. On hyper terminal some parameters are need to be set. These parameters include baud rate, parity bits and number of bits transmitted from computer to microcontroller. Use PC on hyper terminal or download hyper terminal from the net and install it on your PC if pc own HyperTerminal does not work. To install hyper terminal double click on the icon of hyper terminal the following window will appear. Name the new connection press ok.

New connection establishment and parameters settings are shown in figure 4.6.

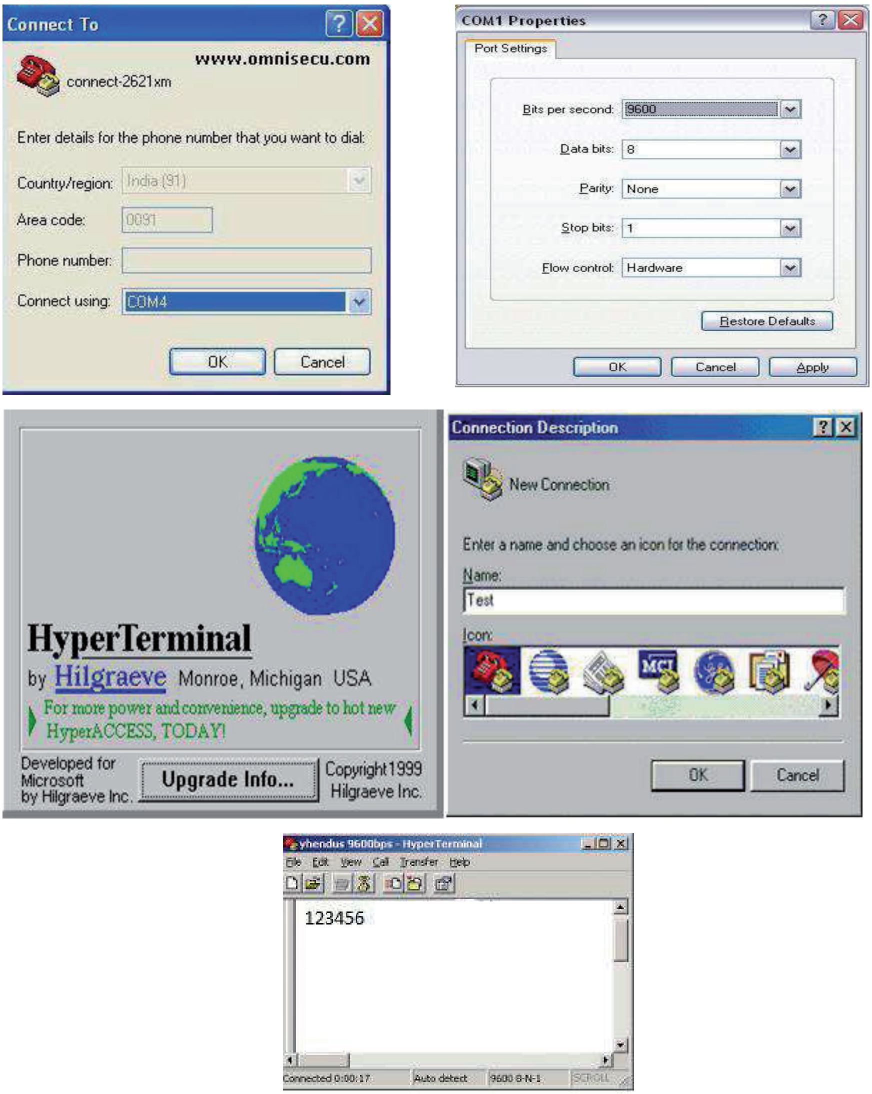


Figure 4.6: Step wise setting of serial port connection using hyper terminal.

Chapter-05

Hardware Platform

The project is based on real time moving object detection and tracking. Static camera mounted on the platform is used to monitor the environment. Video is then processed to detect moving object in the frame sequence. After detecting the object its path is tracked by finding its coordinates. Matlab has been used as intermediate tool. The live feed from camera is processed to detect and track the object's path and then to compute the angles for the motor movement. Calculated angles in matlab are serially transmitted to microcontroller. Microcontroller then generates PWM signal of particular duty cycle depends upon the angles received. This PWM is control signal for servo motor to acquire the dictated angular position which was calculated in matlab. This whole process in figure 5.1.

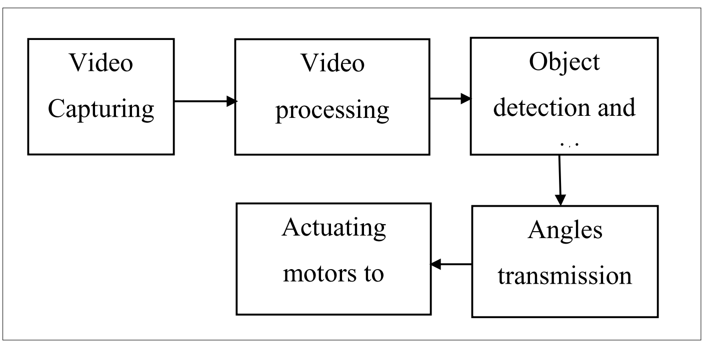


Figure 5.1: Block diagram of the whole system

5.1 Main components

The main components involve in the hardware implementation of the moving target detection and tracking system are,

i. Pan-tilt system

ii. Microcontroller board
iii. Power supply board
iv. Demo Frame

The details of above mention components are given below.

5.1.1 Pan-Tilt System:

The Pan-tilt system is a system consists of two motor, one is pan motor and the other is tilt motor and some other components for fixing these motor in such a way they could move in x-y plane. The motors used to build the Pan-tilt system are Hitec Dc servo motors. Servo motors are very accurate in terms of position and built-in feedback mechanism. Pan-tilt physical description is shown in figure 5.2.

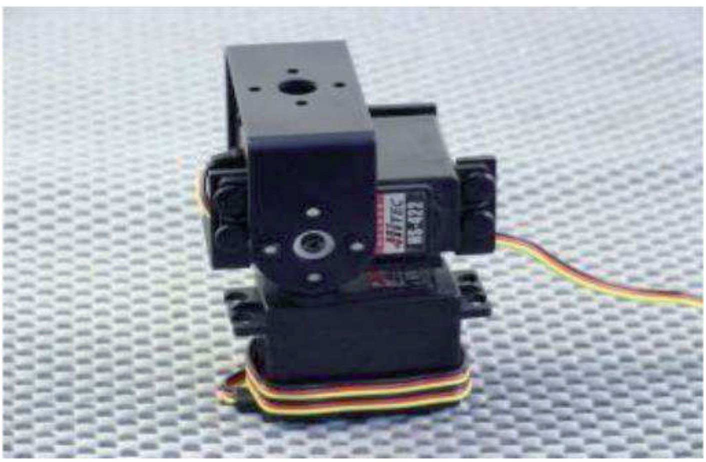


Figure 5.2: Pan-tilt system

Servo motors

Servo motors are a type of Dc motors, with a servo mechanism to provide a precise angular position. Servo motors have different kinds but only three wires servo motors have been considered.

The servo motor is an assembly of the following four things:

i. A normal Dc motor.
ii. A gear reduction unit
iii. A position-sensing device (potentiometer)
iv. A control circuit.

The function of servo is to receive a control signal from a microcontroller or from other controlling device like timer IC555 that represent a desired output position of the servo motor shaft. As servo motor receive control signal from a controlling device (i.e. Atmega16) the servo mechanism apply power to its DC motor until its shaft turn to the desired position. Servo motor has a position-sensing device (potentiometer) which acts as feedback to error detection amplifier. It determines the rotational position of motor shaft, convert it into voltage and applied to error detection amplifier for finding the difference between actual and desired position of shaft. Voltage difference determines which way the motor has to turn. The range of motion for servo motor is 0-180 degrees. The physical description of servo motor is shown in Fig. 5.3.

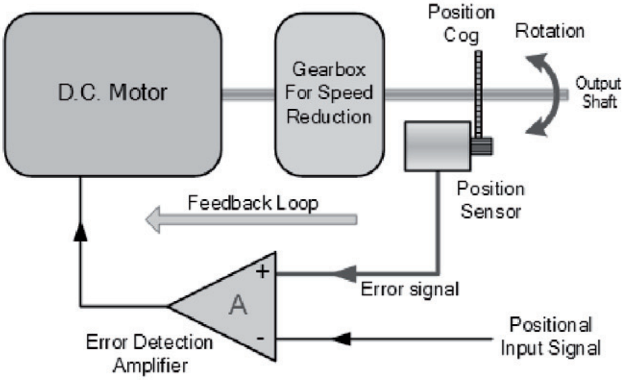


Figure 5.3: Block diagram of servo motor

Servo motor has three wires connection system. These wires along with their purposes are described below.

i. Red
ii. Black
iii. Yellow, white or some other color depending upon the manufactures.

Wires configuration of the servo motors are shown in figure 5.4.

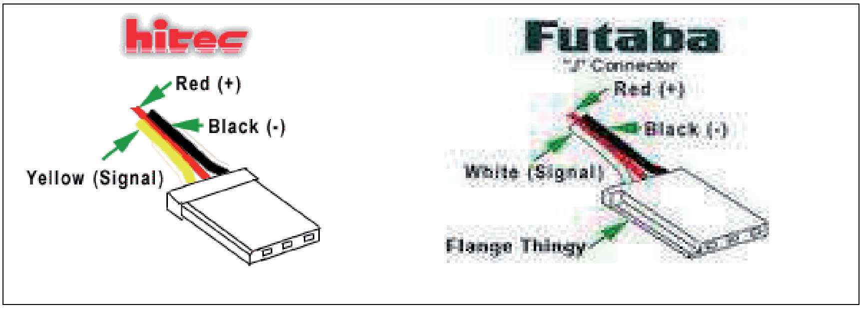


Figure 5.4: Wires configuration of the servo motors

I. The red-wire is used to provide positive supply to the motor.
II. Black wire is used to provide negative supply (ground) to the motor.
III. The third color wire is used for providing control signal to servo motor for its shaft position control.

The control signal is basically PWM signal. Time period of this control signal is needed to be 20ms, that is the PWM signal frequency is to be 50 Hz.

Pulse width of the signal will determine the shaft position (required position) of the servo motor. Range of PWM signal pulse width is different for different servo motors e.g. for Futaba servo motor it is between 1ms to 2.5ms.For Hitec hs422. The calculated pulse width needed for complete range (i.e. from 0 to 180 degree) is between 0.3ms to 2.6ms for the servo motor used in project. At 0.3ms pulse width motor shaft is at the zero degree (extreme lift) position. By increasing the pulse width from 0.3ms to 2.6ms the servo motor start motion in clock wise direction and if it is decreased from 2.6ms to 0.3ms the servo motor move in anticlockwise direction. Now to move motor shaft to a specific position a general formula has been developed that is implemented in microcontroller. OCR values are computed by utilizing the formula and the angles received from matlab. Microcontroller uses this OCR (output capture register) values and generates PWM signal with a pulse width according to required angular position. The PWM signal is shown in Figure-5.5.

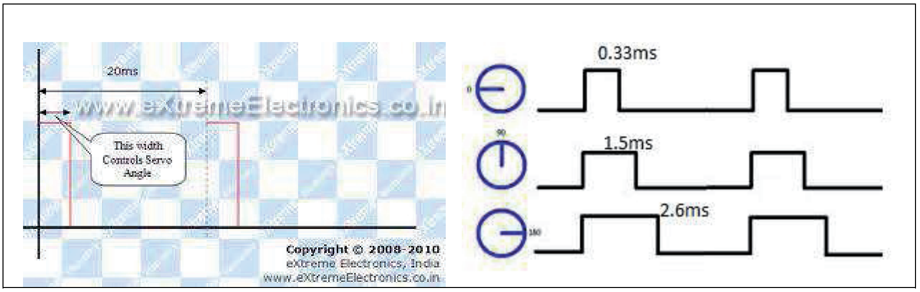


Figure 5.5: PWM signal for servo motor control

5.1.2 AVR Microcontroller board:

AVR microcontroller board consists of the following components:

i. Atmega16 (microcontroller)

ii. Two DB9 (9 pin connector)

iii. Max232

iv. 16x2 LCD

Atmega16

Atmega16 is a 40 pin microcontroller belonging to the Atmega family of microcontroller. As we have previously mentioned that we need a control signal for servo motor shaft position control. We have used atmega16 for that purpose to produce the control PWM signal according to the angle at which we want to move motor shaft.

While programming microcontroller for serial communication care is needed about the following things:

i. Baud rate

ii. Controller frequency

Baud rate

Baud rate is the number of bits transmitted per second. Whenever Serial communication is perform between computer and microcontroller it will need attention toward baud rate setting. If baud rate of computer and controller does not match no communication will happen or the data send from computer will not be correctly interpreted by the controller .e.g. if integer value 5 from computer some garbage value will receive to microcontroller. Different option of baud rates are available e.g. 2400, 4800, 9600 bits/sec etc. In this work baud rate is set to 9600 bits per second. Baud rate explanation is given in figure 5.6

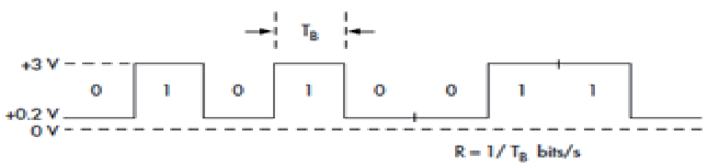


Figure 5.6: Baud rate of microcontroller

Operating frequency of microcontroller

Operating frequency of microcontroller plays very important role in generating of PWM signal for controlling servo motor. Microcontroller provides two options for selecting the operating frequency, one is its internal oscillator and the other one is the external crystal oscillator that can be attached to microcontroller when required. Internal oscillator of the microcontroller have four options i.e.1 MHz, 2 MHz, 4 MHz and 8 MHz, either of them can be used, depending upon requirement. If a frequency higher than 8MHz is required an external oscillator should be attach to microcontroller. External oscillator in range of 8MHz to 16MHz can be used for atmega16 .Atmega16 cannot support above 16MHz.

Internal or external oscillator selection depends upon the requirement. Internal oscillator is not that much stable as the external crystal oscillator .so for obtaining high accuracy external oscillator is used.

For controlling servo motor to a minimum possible degree, microcontroller should be operated at high frequency. At 8 MHz servo can be control to a minimum of 0.7 degree. While at 1 MHz minimum degree will be 5 Degree.

16x2 LCD

Liquid crystal display (LCD) is an electronic screen .16x2 LCD ha capacity that it can display 16 characters per line and there are two such lines. It has two types of registers i.e. data and command registers. Data registers are used to store the data to be display on LCD while the command registers are used to store command to LCD. A command is an instruction given to LCD for performing actions like clearing its screen, setting the cursor position, controlling display etc. LCD used is shown in figure 5.7.

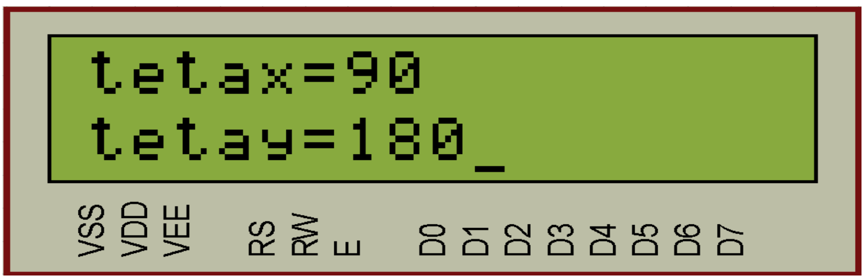


Figure 5.7: 16x2 LCD

Max232

MAX232 is an integrated circuit, first created by Maxim Integrated Products, that converts signals from RS-232 level to TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals. MAX232 is used in circuit where serial communication is to be performed with microcontroller. MAX232 is shown in figure 5.8.

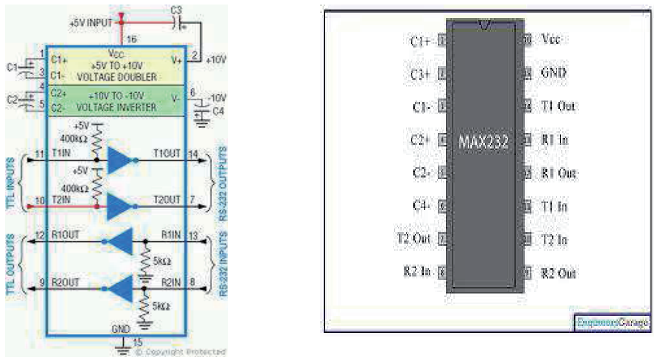


Figure 5.8: Max232 internal circuit and available package in market

DB9 / 9 pin connector

DB9 is nine pin connector used for connecting serial port cable from computer to microcontroller, for the purpose of serially transferring data from personal computer to microcontroller. It has 9 nine different pin whose names, purpose and configuration in shown above. DB9 is shown in figure 5.9.

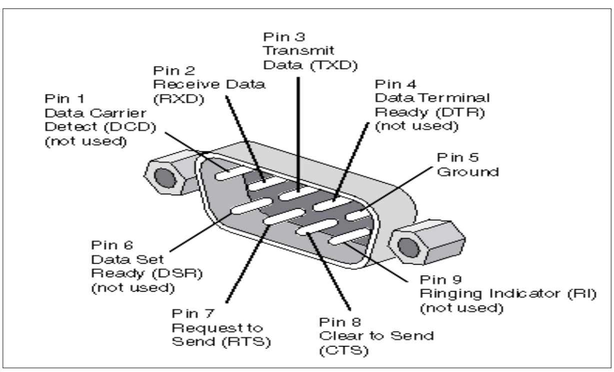


Figure 5.9: DB9/ Nine pin connector

5.1.3 Camera

For real time moving target detection and tracking camera is used like a vision sensor. From camera live video of the region where the target is present is captured for processing. For tacking live video Logitech HD Pro Webcam C920 is used. This camera has USB interface to computer and easy to use. This is a good quality webcam as compare to others webcams of this type. Video is taken through this camera and then it is processed in matlab using computer processor. Camera used is shown in figure 5.10.

Figure 5.10: Logitech HD Pro Webcam C920.

5.1.4 Power board

Servo motors are operated by 5v DC power supply. In order to provide 5v supply to servo motor a power supply board has been made. It consists of 5v voltage regulator (7805), an on/off button and connecting pins. Power is supplied to both motors from this board. There are three pins for each motor supply one for +ve, 2nd is ground and the third one is PWM pin from microcontroller.

5.1.5 Demo Frame

The whole hardware setup is being made on a wooden frame. It includes a board, a rectangular stand and two wooden tracks like a railway track. Board is has dimensions of 4x4 feet. For experimental purpose of detecting and tracking of moving objects an object will be move on this board. This board has been placed in front of camera. The camera and pan-tilt system has been fixed on a rectangular stand having a height of 1.5 feet. Pan-tilt system is placed below camera.

Rectangular stand is placed on the track. The purpose of the track is to move the stand forward and backward in order to decrease or increase distance between board and camera. The track is of length 4 feet. The whole setup is shown in figure 5.11.

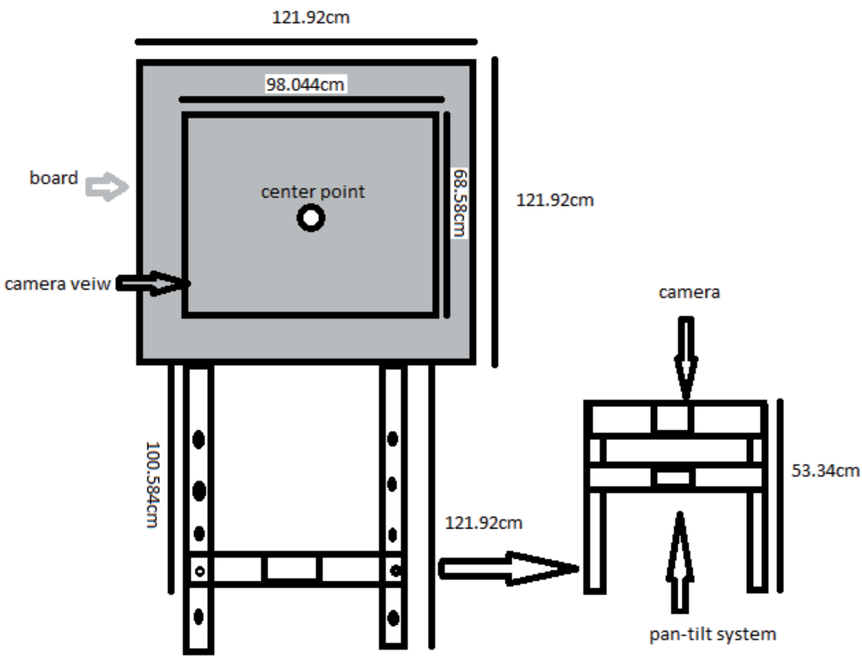


Figure 5.11: Demo frame

5.2 Target Locking

The tracked object is locked using control system and tracking function information. From the centroid information, find angles with reference to center of the frame (x_0, y_0). The angles calculated are then passed to microcontroller board, which directs motors motion accordingly.

Let the angles calculated currently are T1 and T2, where T1 is horizontal angle and T2 is vertical angle. Tx and Ty are previous horizontal and vertical angles respectively. Then motion angles are calculated using equations given below.

$$T_{mx} = T1 - Tx \qquad T_{mx} = T1 - Tx \tag{5.1}$$

$$T_{my} = T2 - Ty \qquad T_{my} = T2 - Ty \tag{5.2}$$

Where T_{mx} and T_{my} are horizontal motor angle and vertical motor angle respectively. These angles are passed to microcontroller board through serial communication. The operational flow diagram of complete interface is shown in figure 3.3.

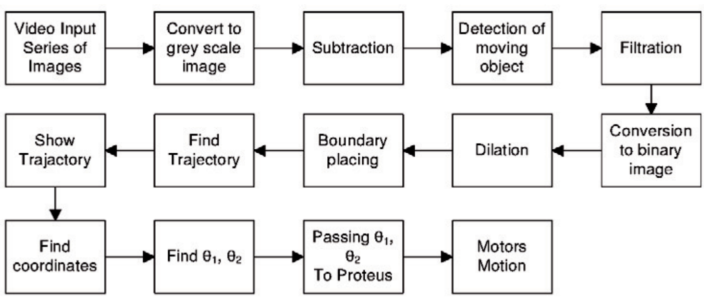


Figure 5.12: Operation Flow Diagram

5.2.1 Angles calculation

Tracking algorithm gives position of the desired moving object as output. Position obtained is compared with reference position (center of the frame), and angles are calculated. Two angles are calculated here i.e. horizontal angle and vertical angle. Horizontal angle shows motion of the Pan motor and vertical angle shows motion of the Tilt motor. Target is locked on the basis of motors motion. This method is described in figure 5.11: (x_0, y_0) is center of the frame, (x_1, y_1) is center position of the tracked object, D is distance between center of the frame and center of the camera, dx and dy are distances from reference position to the center of object along x-axis and y-axis respectively, T1 is the horizontal or Pan motor angle and T2 is the vertical or Tilt motor angle. Angles are calculated using equation 3 and 4.

$$T1 = \arctan\left(\frac{d_x}{D}\right) \tag{5.3}$$

$$T2 = \arctan\left(\frac{d_y}{D}\right) \tag{5.4}$$

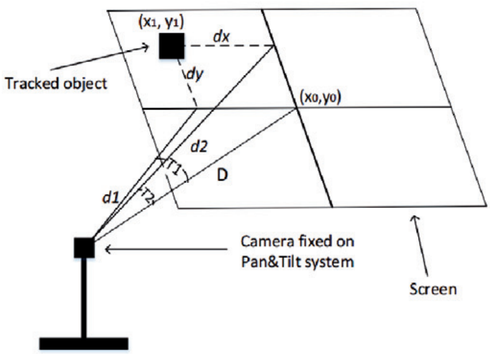


Figure 5.13: Angles calculations

5.2.2 Control system

Control system consists of three steps: serial communication, control signal and motors motion.

Serial communication

Angles calculated using eqn.3 and 4 are communicated serially to micro controller board. In simulation interface virtual serial ports and proteus are used. Virtual serial ports are created, the data (angles) are communicated serially through that ports and received on micro controller board generated in proteus. In hardware platform angles are communicated through serial port to micro controller board.

Control signal

Microcontroller board generates Pulse Width Modulation (PWM) signals for servo motors motion. PWM signals are generated using Output Compare Register (OCR) values. OCR values are obtained using eqn. 7 on the basis of angles received.

$$Y = 2.434 * X + 95.978 \tag{5.5}$$

Y is OCR value of microcontroller and X is the angle received serially. This is shown in figure 5.12.

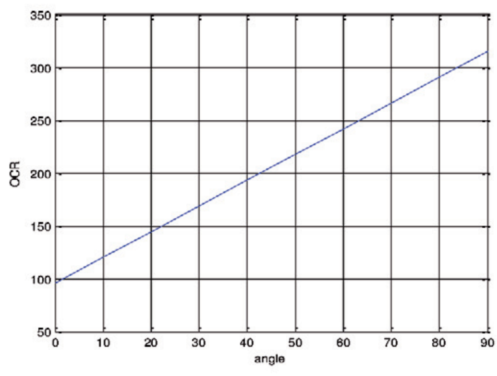


Figure 5.14: OCR calculation

Motor motion

The PWM signals generated are given as control input to servo motors (Pan& Tilt motors). Servo motors are moved to specific positions according to control signal. The camera is fixed on Pan& Tilt mechanism moves accordingly.

Chapter-06

Results Framework

In previous chapters algorithms are implemented to achieve the task of moving object detection and tracking system. Locking is achieved as an application. This chapter describes the results obtained from recorded videos and real time scenarios.

6.1 Recorded video Results

This section mainly focuses on results obtained from recorded video. Two different algorithms are implemented to obtain the desired results. A brief overview of the results is shown below.

6.1.1 Frame Differencing Algorithm

Frame differencing algorithm is widely used for motion analysis. It has very efficient results in detection of moving objects. Frame differencing algorithm depends upon reference frame and current frame, threshold and illumination changes. It gives centroid (center coordinates) of moving object and area of moving object as output. Using centroid information, we calculate trajectory of the desired tracked moving object. Angles of motors' motion relative to center position of the frame are calculated to lock the tracked object.

Some results derived on recorded video are described here which shows the efficiency and importance of FD algorithm. Figure shows tracking results of a tested movie named as 'ball.avi'. In figure 6.1, frame shows original object, 2nd frame shows the detected and tracked object. Then 10th, 43rd, 60th, 75th and 110th frames are shown, which all shows the tracked object and the last part of the figure shows the path or trajectory followed by object in motion.

Same results as above are shown in figure 6.2 only with changes in color. Applying pseudo coloring the results obtained are shown.

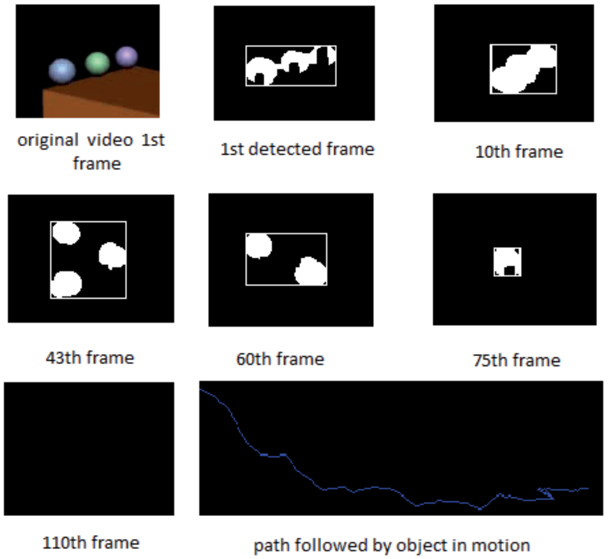


Figure 6.1: shows frame wise tracking results and path of tracked object inn black-white.

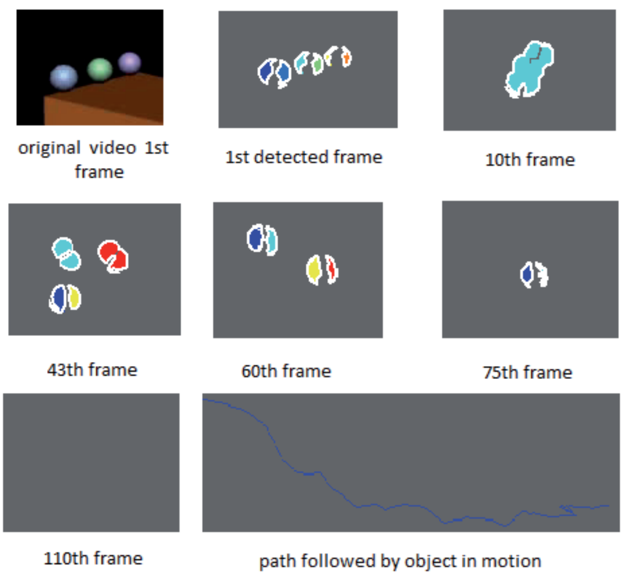
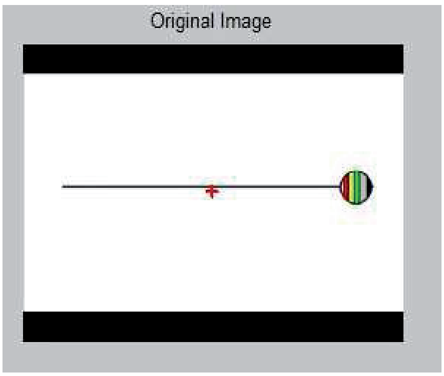
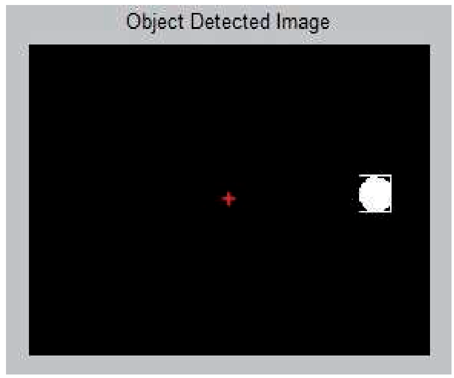


Figure 6.2: shows frame wise tracking results and path of tracked object using pseudo coloring.

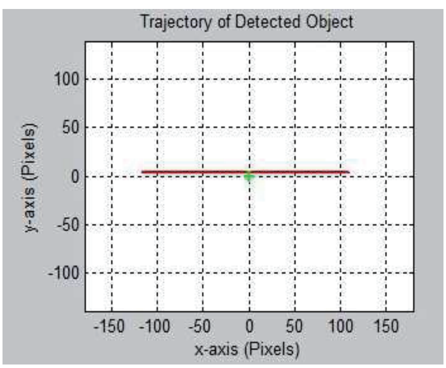
Further processing on tracked object leads us to some important information which are shown in figure. Figure 6.3 shows some information got from the algorithm results. (a) shows the original video frame by frame. (b) shows tracked object frame to frame. (c) shows path followed by the tracked object in cartesion coordinates. (d) shows distance of tracked object from center position of the frame. (e) shows the angular position of the tracked object with respect to center of frame and. (f) shows the combine image of all the images described above.



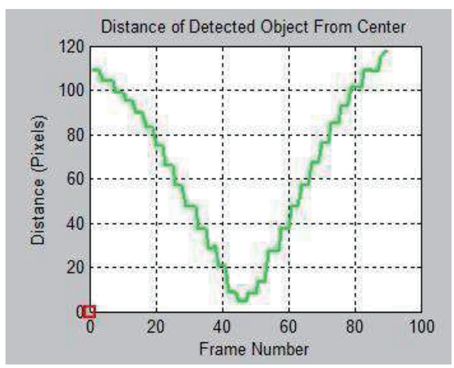
(a)



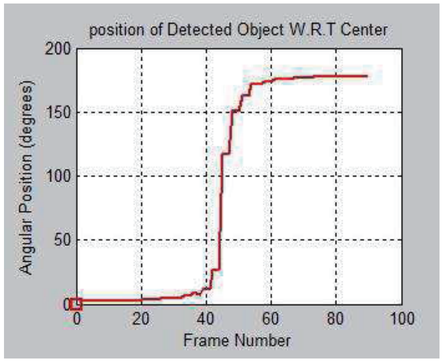
(b)



(c)



(d)



(e)

Figure 6.3: shows original video frames, tracked object video frames, path followed by tracked object, distance of tracked object from center position of frame and angular position of tracked object with reespect to center of frame separately.

6.1.2 Mean-Shift Algorithm

Mean shift algorithm is widely used for moving object tracking and is an efficient technique for moving object tracking. It can be implemented in two ways i.e. on the basis of simple mean and on the basis of statistical mean calculations of color in the target model and the candidate model.

The results obtained on the basis of simple mean are shown in figure. The figure 6.4 shows a frame wise tracking result. As we are using simple intensity mean calculations, so the tracking is not very efficient because as the intensity or light varies so it directly affects tracking process and the target loses.

This limitation can be overcome using traditional MS algorithm which uses Kernel function to calculate mean of the selected patch, and then uses Bhattacharyya coefficients for similarity measures. The new position is calculated and the template is updated. Figure 6.5 shows the results obtained from traditional MS algorithm. It shows the frame wise tracking results of MS algorithm.

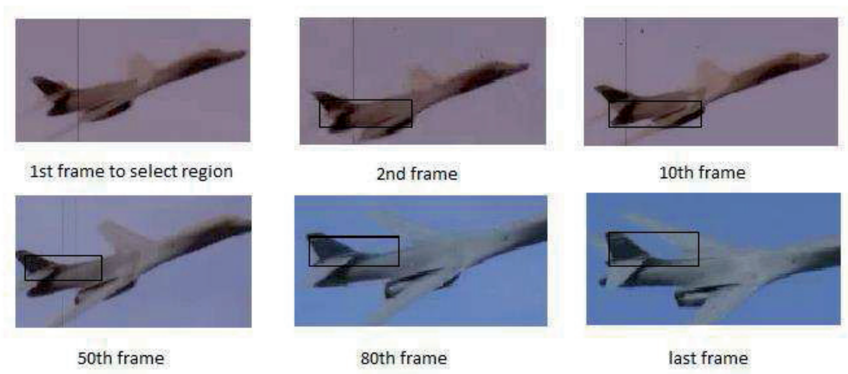


Figure 6.4: Frame wise tracking results of simple Mean-Shift algorithm.

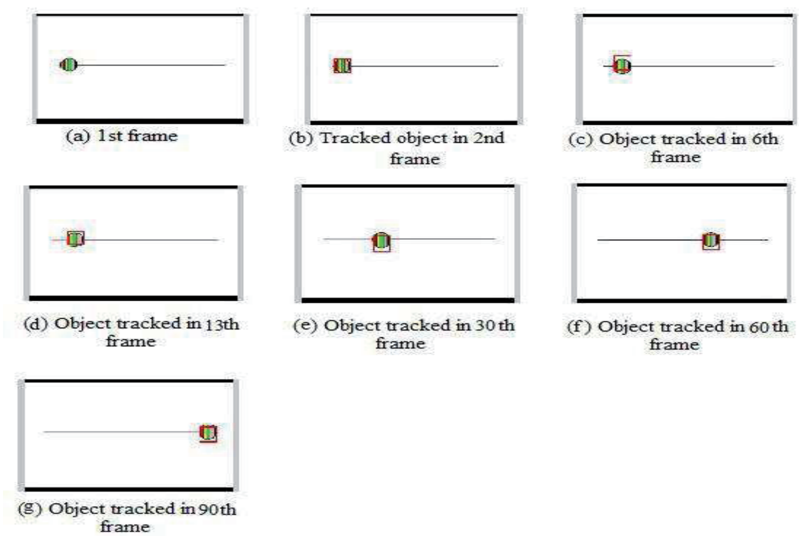
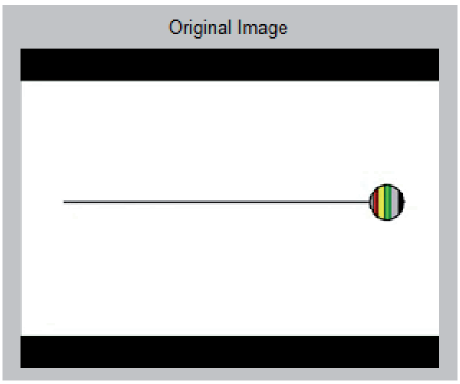
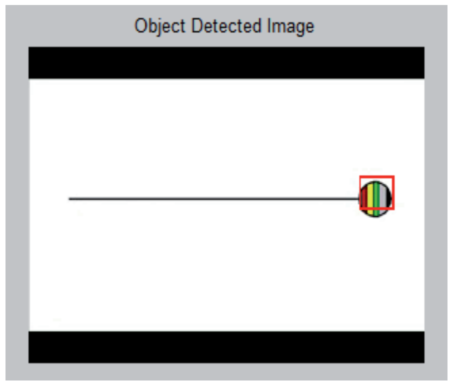


Figure 6.5: Frame wise tracking results of Kernel based Mean-Shift algorithm.

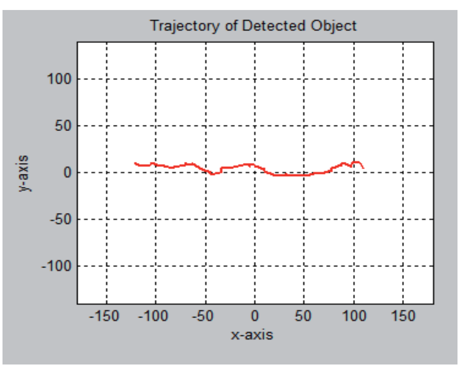
Further processing on tracked object leads us to some important information which are shown in figure. Figure 6.6 shows some information got from the MS algorithm results. (a) shows the original video frame by frame. (b) shows tracked object frame to frame. (c) shows path followed by the tracked object in cartesion coordinates. (d) shows distance of tracked object from center position of the frame. (e) shows the angular position of the tracked object with respect to center of frame and. (f) shows the combine image of all the images described above.



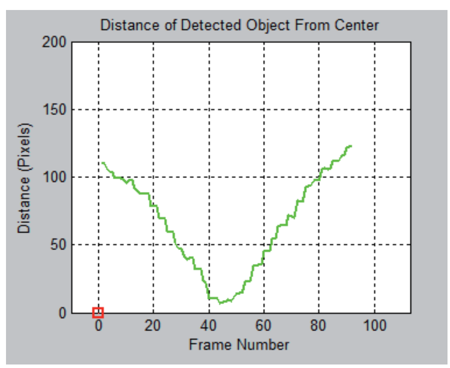


(a)

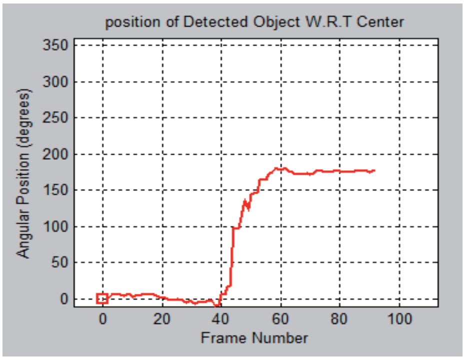
(b)



(b)



(d)

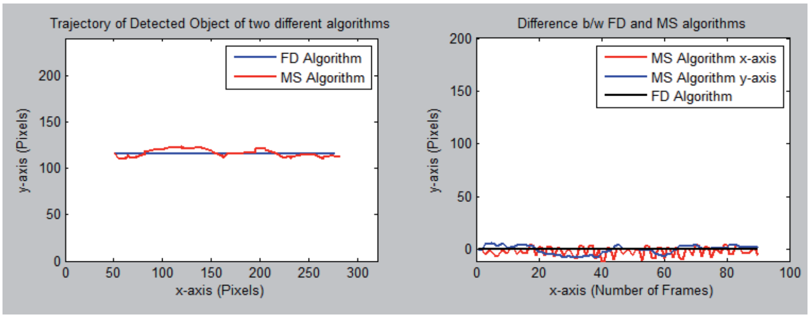


(e)

Figure 6.6: shows original video frames, tracked object video frames, path followed by tracked object, distance of tracked object from center position of frame and angular position of tracked object with reespect to center of frame separately.

6.1.3 Comparison between Frame Differencing and Mean-Shift algorithm

Moving target detection and tracking is very important task in analysis of moving bodies. To accurately and efficiently detect and track moving targets frame differencing and mean-shift algorithms are implemented. The results obtained from both the algorithms satisfy selection. Frame differencing is mainly used for detection purposes and mean-shift is widely used for tracking purposes. Here both detection and tracking is achieved using frame differencing and mean-shift algorithms separately. Figure 6.7(a) shows the trajectory of the detected in the same video processed over both algorithms and figure 6.7(b) shows the difference between the results or difference the trajectory coordinates. The figure shows that mean-shift is less accurate for detection purposes while frame differencing is more.



(a) (b)

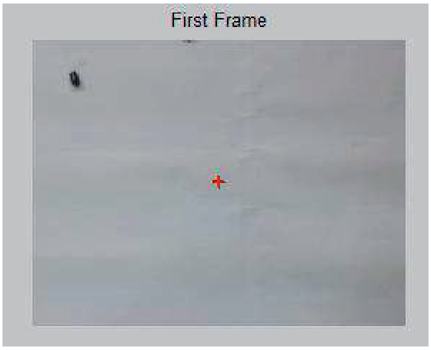
Figure 6.7: Comparison between frame differencing and mean-shift algorithm

6.2 Real time or Live video Results

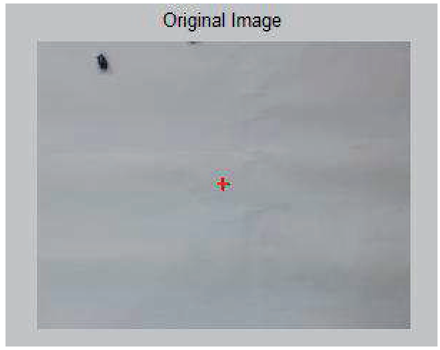
This section mainly focuses on results obtained from real time hardware. Two different algorithms are implemented to obtain the desired results. A brief overview of the results is shown below.

6.2.1 Frame Differencing Algorithm

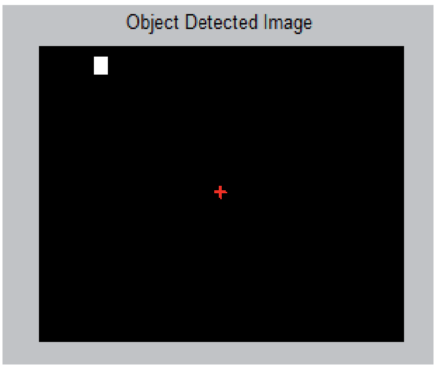
Frame differencing algorithm is widely used for motion analysis. It has very efficient results in detection of moving objects. Frame differencing algorithm depends upon reference frame and current frame, threshold and illumination changes. It gives centroid (center coordinates) of moving object and area of moving object as output. Using centroid information, we calculate trajectory of the desired tracked moving object. Angles of motors' motion relative to center position of the frame are calculated to lock the tracked object. Some results derived on real time or live video are described here which shows the efficiency and importance of FD algorithm. Figure 6.8 shows tracking results of a live video. Further processing on tracked object leads us to some important information which are shown in figure. (a) shows the start frame. (b) shows original object frame to frame. (c) shows tracked object frame to frame. (d) shows path followed by the tracked object in Cartesian coordinates. (e) shows distance of tracked object from center position of the frame. (f) shows the angular position.


First Frame

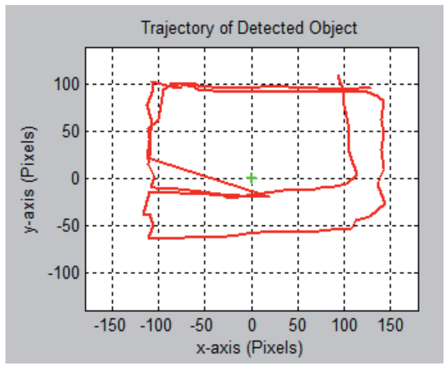
(a)


Original Image

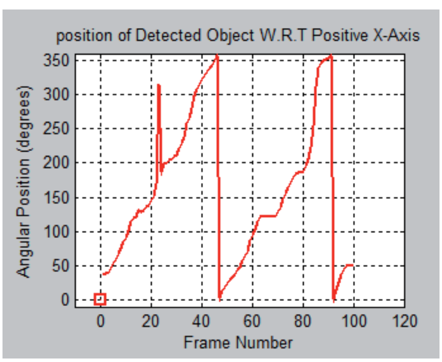
(b)


Object Detected Image

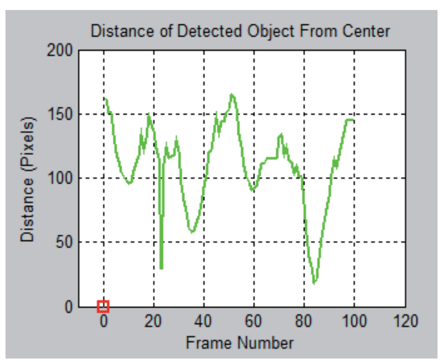
(c)


Trajectory of Detected Object
100
50
0
-50
-100
y-axis (Pixels)
-150
-100
-50
0
50
100
150
x-axis (Pixels)

(d)


position of Detected Object W.R.T Positive X-Axis
350
300
250
200
150
100
50
0
Angular Position (degrees)
0
20
40
60
80
100
120
Frame Number

(e)


Distance of Detected Object From Center
200
150
100
50
0
Distance (Pixels)
0
20
40
60
80
100
120
Frame Number

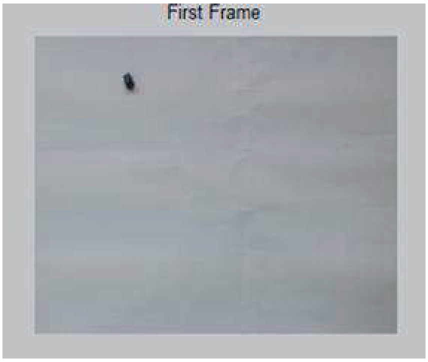
(f)

Figure 6.8: shows original live video frames, tracked object video frames, path followed by tracked object, distance of tracked object from center position of frame and angular position of tracked object with reespect to center of frame separately.

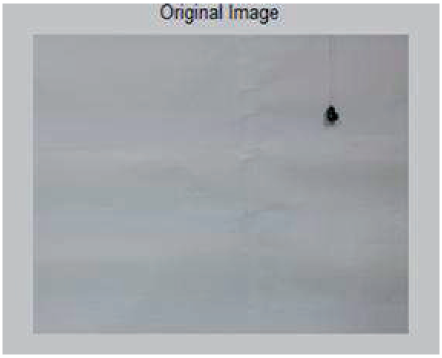
6.2.2 Mean-Shift Algorithm

Mean shift algorithm is widely used for moving object tracking and is an efficient technique for moving object tracking. It is be implemented on the basis of statistical mean calculations of color in the target model and the candidate model. The results obtained are shown in figure. Figure 6.9 shows a frame wise tracking result. MS algorithm uses Kernel function to calculate mean of the selected patch, and then uses Bhattacharyya coefficients for similarity measures. The new position is calculated and the template is updated.

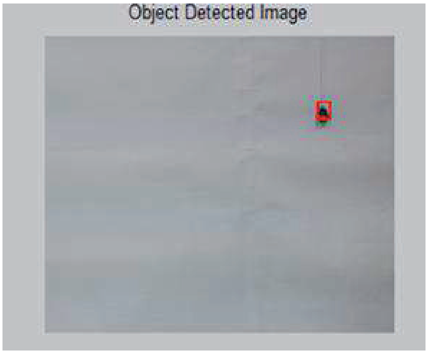
Further processing on tracked object leads us to some important information which are shown in figure. Figure 6.9 shows some information got from the MS algorithm results. (a) shows the start frame. (b) shows original object frame to frame. (c) shows tracked object frame to frame. (d) shows path followed by the tracked object in Cartesian coordinates. (e) shows distance of tracked object from center position of the frame. (f) shows the angular position of the tracked object with respect to positive x-axis.


First Frame

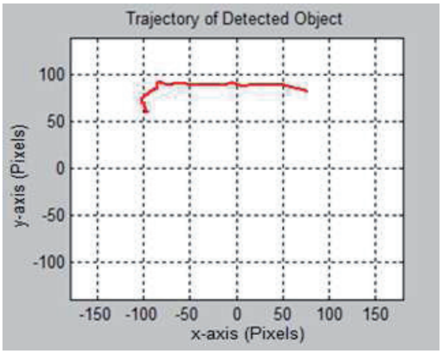
(a)


Original Image

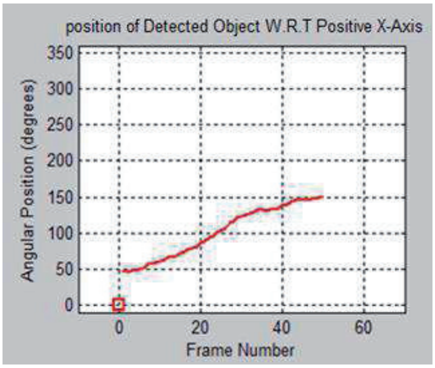
(b)


Object Detected Image

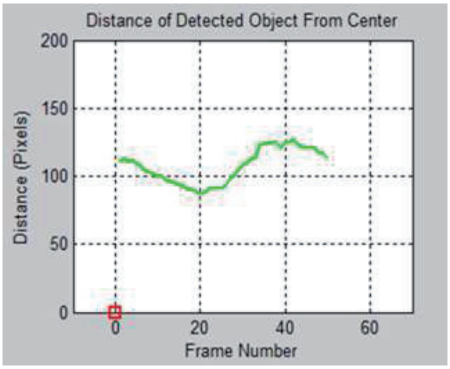
(c)


Trajectory of Detected Object
y-axis (Pixels)
x-axis (Pixels)
100
50
0
-50
-100
-150
-100
-50
0
50
100
150

(d)


position of Detected Object W.R.T Positive X-Axis
Angular Position (degrees)
Frame Number
350
300
250
200
150
100
50
0
0
20
40
60

(e)


Distance of Detected Object From Center
Distance (Pixels)
Frame Number
200
150
100
50
0
0
20
40
60

(f)

Figure 6.9: shows original live video frames, tracked object video frames, path followed by tracked object, distance of tracked object from center position of frame and angular position of tracked object with reespect to center of frame separately.

Chapter-7

Conclusion and Future Work

7.1 Conclusion

This system can successfully detect, track and lock the moving object in real time scenarios. This system is being implemented with two different algorithms i.e. Frame differencing and Mean Shift Algorithm.

The detection and tracking is taken out through Frame differencing algorithm and through Mean Shift Algorithm. In frame differencing two consecutive frames are compared and moving object is detected and tracked. After detection and tracking the moving object is locked. The system is implemented in two platforms. First one is simulation platform and second one is hardware platform. For simulation platform matlab is interfaced with Proteus through virtual serial port. The complete hardware model is designed in proteus and simulated. For hardware implementation Demo Frame is made on which Pan-Tilt system and camera are fixed. Microcontroller board is for motors motion control and through serial communication the objective of locking is achieved. In order to lock the detected object two angles are required for pan-tilt system are calculated. These angles are sent to AVR kit where pulses are generated to control motors motion. One motor is for horizontal motion and the second motor is for vertical motion. The laser light is fixed on motors and used to point out the moving object. This algorithm is used for just detection and tracking. This algorithm cannot recognize the detected object.

7.2 Future Work

This system is not capable of identification and recognition. This system is made for fixed background. This system can be implemented where the pixel value is constant or where the pixel value is changing with a constant pattern with respect to time.

The system can further be modified for identification. For example somebody wants to detect, track and lock just vehicles, humans or anything else. In this situation the system should be capable to identify the moving object, that whether this is a car or human body or anything else. This can be the future work of this project.

Our system can also be further modified for recognition purpose. The system can then be implemented in place where the recognition of every single person is needed. And the system will only lock that person which is not in the database of algorithm. So the recognition can also be a future work of this project.The system can be made environment free. If light effects changes or background changes so the system will not be affected. This is also a good point for future work.

Appendix A

A.1 Schematic Diagram of Power Board

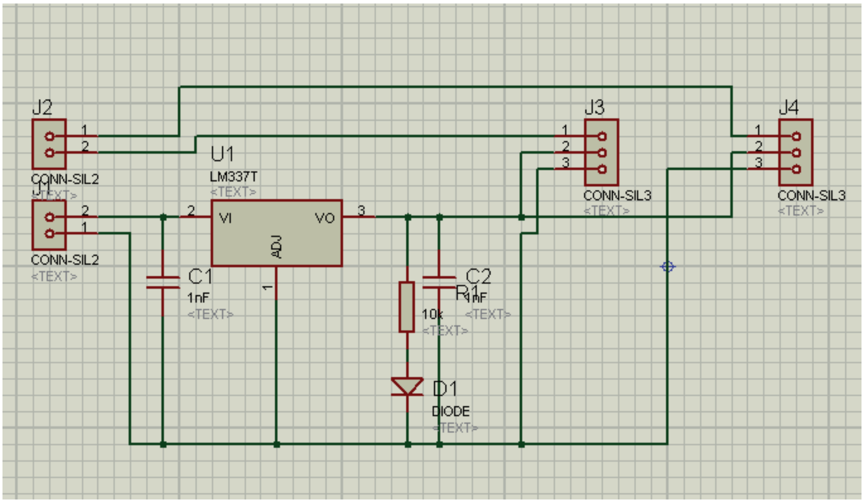


Figure: Schematic diagram of power board

A.2 Proteus Layout of Power Board

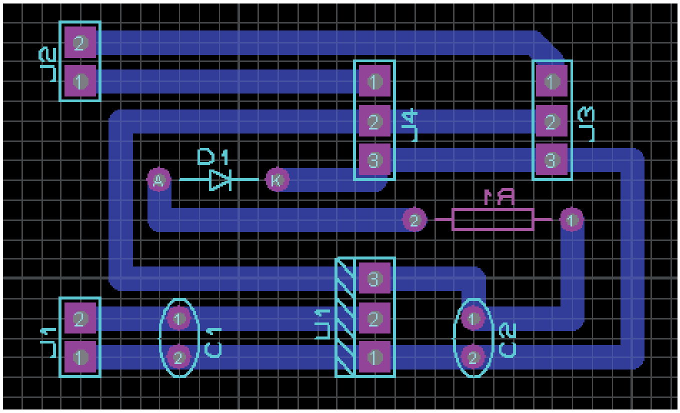


Figure: Proteus Layout of power board

A.3 Proteus Layout of Microcontroller Board

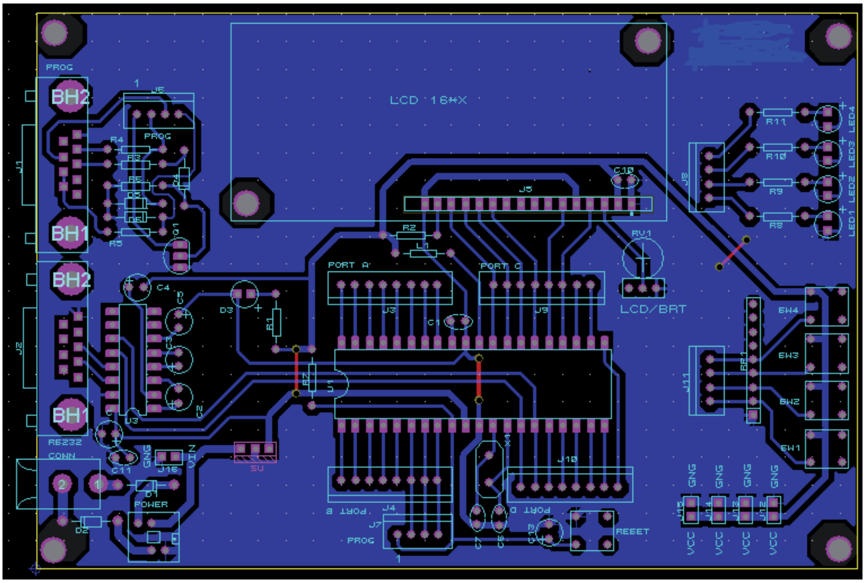


Figure A.3: Proteus Layout of Microcontroller Board

A.4 3D Layout of Microcontroller Board

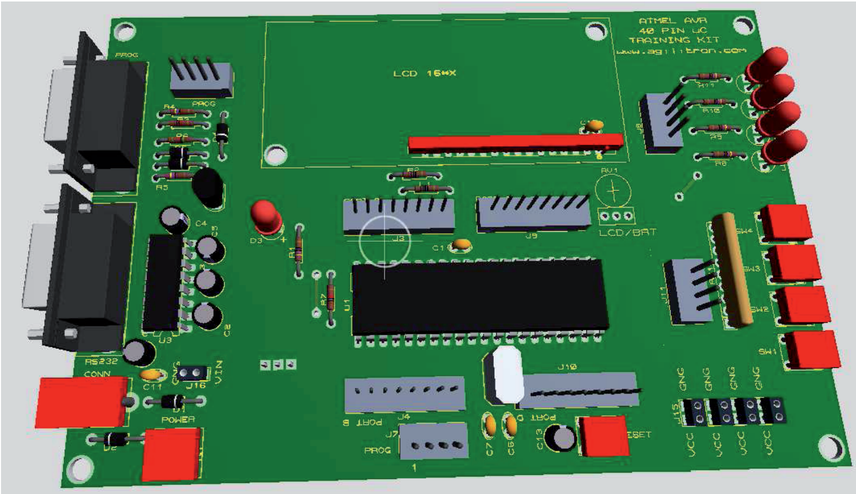


Figure A.4: 3D Layout of Microcontroller Board

Appendix B

B.1 Schematic Diagram of Power Board

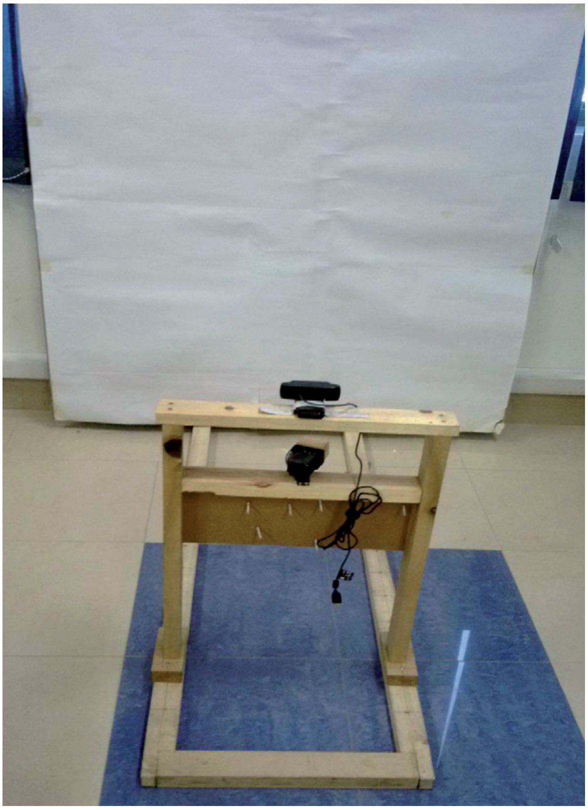


Figure B.1: Actual Image of Demo frame

Bibliography

[1] N. Prabhakar, etal "Object Tracking Using Frame Differencing and Template Matching," Research Journal of Applied Sciences, Engineering and Technology 4(24): 5497-5501, 2012.

[2] Amedome. M-D Kodjo, Y. Jinhua, "Real-time Moving Object Tracking in Video", International Conference on Optoelectronics and Microelectronics (ICOM), 978-1-4673-2639-1/12, 2012.

[3] PANDE R.P. etal, "detection of moving object with the help of motion detection alarm system in video surveillance," Journal of Signal and Image Processing, Volume 3, Issue 3, pp.-118-121, 2012.

[4] Hua-Jun Song and Mei-Li Shen, "Target Tracking algorithm based on Optical Flow Method using Corner Detection", Journal of multimedia tools and applications, volume 52, 121-131, march 2011.

[5] Kai Du etal, "MeanShift Tracking Algorithm with Adaptive Block Color Histogram," Special Fund for Basic Scientific Research in the Central University (CHD2010ZY012), 978-1-4577-1415-3/12, 2012.

[6] Jong Sue Bae and Taek Lyul Song, "Image Tracking Algorithm using Template Matching and PSNF-m", International Journal of Control, Automation, and Systems, vol. 6, no. 3, pp. 413-423, June 2008.

[7] Min Tang, "Edge Detection in Image Segmentation Based on Cellular Neural Networks", 978-1-4244-2902-8/09, 2009

[8] M. Wrioutius, J. Y. Ramel, and N. Vincent, "Selection of points for on-line signature comparison," in Ninth International Workshop on Frontiers in Handwriting Recognition, pp. 503–508, October 2004.

[9] D. Simon, "Kalman filtering with state constraints: a survey of linear and nonlinear algorithm," lET Control Theory and Applications, vol. 4, no. 8, pp. 1303-1318,2010.

[10] D. S. lang, S. W. lang, and H. Choi, "20 human body tracking with Structural Kalman filter," Pattern Recognition, vol. 35, no. 10, pp. 2041-2049,2002.

[11] Seok-Wun Ha, Yong-Ho Moon, "Multiple Object Tracking Using SIFT Features and Location Matching", International Journal of Smart Home Vol. 5, No. 4, October, 2011.

[12] David. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision 60(2), 91–110, 2004.