Speed Calculation Using Image Processing

Submitted in total fulfillment of requirements
For the degree of

Bachelors in Technology

by

Name: Sehab Mankada
Roll No.: 1414029

Name: Vidhi Chitalia
Roll No.: 1414074

Name: Nidhi Gada
Roll No.: 1414079

Guide: Prof. Sunayana Jadhav

DEPARTMENT OF INFORMATION TECHNOLOGY
K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch 2014-2018

Certificate

This is to certify that the Dissertation entitled **"Speed Calculation Using Image Processing"** is bonfide record of the dissertation work done by

1. Sehab Mankada
2. Vidhi Chitalia
3. Nidhi Gada

in the year 2017-18 under the guidance of Prof. Sunayana Jadhav of Department of Information Technology in partial fulfillment of requirement for the Bachelors in Technology degree in Information Technology Engineering of University of Mumbai.

Prof. Sunayana Jadhav	Prof. Sujata Pathak
Guide	Head Of Department

Dr. Shubha Pandit
Principal

Date : 28/5/2018

Place : Mumbai - 400077

K.J. Somaiya College of Engineering, Mumbai-400077
(Autonomous College Affiliated to University of Mumbai)

Department of Information Technology 2014-2018 Batch

Certificate of Approval of Examiners

We certify that this dissertation report entitled **Speed Calculation Using Image Processing** is bonafide record of project work done by

1. Sehab Mankada
2. Vidhi Chitalia
3. Nidhi Gada

This project is approved for the award of Bachelors in Technology Degree in Information Technology Engineering of University of Mumbai.

____

Internal Examiner

____

External Examiner

Date : 28/5/2018
Place: Mumbai 77

K.J. Somaiya College of Engineering, Mumbai-400077
(Autonomous College Affiliated to University of Mumbai)

Department of Information Technology **2014-2018 Batch**

Declaration

We declare that this written thesis submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in our submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

-- **Signature of the Student** **1414029** -- **Roll No.**	-- **Signature of the Student** **1414074** -- **Roll No.**
-- **Signature of the Student** **1414079** -- **Roll No.**	

Date : 28/5/2018

Place : Mumbai - 400077

Department of Information Technology 2014-2018 Batch

Table of Contents

Abstract

Recently it receives a great attention to reduce road accident and controlling traffic by limiting the speed of vehicles.Moving vehicle speed is usually determined by using Radar, Photo Radar, Drone Radar, LIDAR Speedometer Clocks, Average Speed Computers, Aircraft or Video frame methods, etc. Most of these methods are very expensive and difficult to integrate though provides the best accuracy.

Currently, computational of the optical flow of a sequence of images still remains a challenge in video processing. There are no specific techniques that can sufficiently generate an accurate and dense optical flow. Computing using local variable such as Lucas Kanade algorithm does not provide a good segmentation which indirectly affects the pattern of the optical flow obtained. In this project, two video based vehicle detection methods are presented namely Lucas Kanade and Kalman Filter which later is used for calculating the speed of those vehicles.

These methods consist of three stages: object recognition, noise removal and object tracking. Optical Flow Analysis in combination to segmentation of the motion is used by Lucas Kanade method. Gaussian Mixture Model is the method used for calculating the centroid of the tracked vehicle. Lastly, the results of both the experiments will be compared against their output to give the users the accurate speed of vehicles.

***K*eywords:** Video frame, Images, Optical Flow, Lucas Kanade, Kalman Filter, Speed, accuarcy, Vehicles

List of Figures

List of Tables

Chapter 1

Introduction

This chapter gives an overview of the project by providing information about the problem definition, motivation, scope, requirements.

1.1 Problem Definition

India is facing a lot of problems dealing with road traffic and we have come up with some ideas that could prove a boon to the road traffic. Our project deals with the technology of image processing in calculating the speed of moving vehicle. A comparison will be made between Lucas Kanade and Kalman Filter on the basis of accuracy of speed and their corresponding error rates shown by both the experiments.

1.2 Motivation

Expensive radar system which is being used now-a-days requires sophisticated technology which is not feasible to each and every individual. The system implemented here involves the differentiation of two algorithms based on their outputs which includes the efficient calculation of the speed of vehicles. The accuracy rate will be displayed which will help the vehicles save traffic and speed on highways.

1.3 Scope

Speed measurement is one of the crucial problems in traffic surveillance. So far, the field is dominated by radar and section speed measurements because they meet the methodological standards and requirements. To overcome these antiquated methods we have come across the image processing techniques that can be used to control the over-speeding vehicles by detecting their speeds.Here we will only focus on differential methods which are Lucas Kanade and Kalman Filter. Comparison is made based on the optical flow pattern, segmentation of the motion of the images and the processing time.

1.4 Salient Features

In the current project, we compare tracking of vehicles using two different algorithms (Lucas kannade and Kalman Filter) and with the help of these two algorithm we calculate speed of the moving object. In all previous projects one out of the two algorithm were used to provide the necessary result. By comparing these two algorithm, we obtain the optimal speed calculation from one of the best suited algorithms.

1.5 Organization of Project

Chapter 1 majorly talks about overview of the project selected by us and defines the motivation behind the project selected and the scope of the project.
Chapter 2 provides literature survey done to understand the topic better.
Chapter 3 provides the Software Project Management Plan(SPMP) for this project which defines the project management goals of the project and includes a description of the deliverable and deadlines.
Chapter 4 provides software requirement specification which deals with hard- ware, software, network and database requirements.
Chapter 5 software design specification deals with the architecture, data, user-interface and component design.
Chapter 6 deals with the test cases applied for the project and testing tools that are required for the same.
Chapter 7 implies the conclusion which can be drawn from the complete execution of the project, its advantages and its utility in real-life circumstances.
Chapter 8 talks about the implementation approach adopted while developing the system and all the other minute details related to the project.
Chapter 9 displays the results obtained at the end of project and the possible future prospects where the project can be made put to use.

Chapter 2

Literature Survey

This chapter describes about the survey which has gone through selecting the project, its usefulness to the masses and also different projects related to the one selected.

The world is facing big problem while dealing with road traffic. Costly radars and sensors are one of the reason we need to think about some alternate and cheaper solutions. Our project gives a solution to this problem using image processing techniques. The two algorithms viz. Lucas Kanade and Kalman Filter are being compared and an efficient algorithm is chosen based on the efficiency. It normally has following process
1. Object Detection
2. Object Tracking

In the year 2009, Lee Yee Siong , Siti Salasiah Mokri , Aini Hussain , Norazlin Ibrahim , Mohd Marzuki Mustafa proposed Motion Detection Using Lucas Kanade Algorithm and Application Enhancement (Lucas).
This paper they showed how to use the foreground detector and blob analysis to detect and count cars in a video sequence. It assumes that the camera is stationary and uses the gaussian mixture model(GMMs) to calculate the centroid.

In the year 2017,Mr.Mohammad Reduanul Haque proposed Vehicle speed determination from video streams using image processing.
This paper shows video based vehicle speed calculation method is presented, which has the capability of calculating the speed with higher accuracy but at relatively low cost. The method consists of three stages: object recognition, noise removal and speed calculation. The method shows satisfactory performance on standard data sets using Optical Flow Analysis.

In the year 2016, Mrithu A S proposed An Efficient Implementation of Video Based Traffic Analysis System(Kalman Filter Algorithm)
This paper shows that a video can be used to detect the vehicles, track and count them with high accuracy. The video stream is broken into n frames and a combination of running Gaussian average along with local binary pattern is used for vehicle detection.

For multiple object tracking, Kalman filtering is used. The system will count the number of vehicles passed through a particular area and determine the speed of vehicles passed. The experimental analysis have shown that the proposed system show high detection rate and is feasible.

In the year 2017, Shalaka S. Wardha , Prof. S. M. Deokar, Prof. S. S. Patankar, Prof. J. V. Kulkarni proposed “Development of Automated Technique for Vehicle Speed Estimation and Tracking in Video Stream ”.
The paper has presented a technique for vehicle speed measurement and vehicle tracking. Mathematical formulations are achieved for speed calculation based on the number of frames, frame rate and distance traversed by vehicle. This results into achieving the underlying accuracy that is required for calculating the speed of a vehicle.

In the year 2009, Lee Yee Siong implemented A Motion Detection Using Lucas Kanade Algorithm and Application Enhancement .
This paper described an algorithm to evaluate the real time speed of vehicles from the known camera calibration parameters using image processing techniques. The frame differencing technique is applied on the moving vehicle, the object tracking procedure is executed and speed is calculated using the displacement of the centroids. These centroids helps to calculate the distance covered by a vehicle of a particular frame in real time environment.

Mr. Jesse Scott proposed A Kalman Filter Based Video Background Estimation
This paper used sequence of images to estimate the speed of the vehicle by an uncalibrated camera where the parameters are known prior to the execution. Sequencing of the images allows to acquire centroid of a vehicle which in turns assists to estimate the speed of the vehicle. Estimation of speed doesn’t require precise calibration but some common inference which reduce complex structure.

Chapter 3

Software Project Management Plan

In this document we have discussed the Software Project Management Plan. This document discusses the process model we have used for development. It also discusses roles and responsibilities along with the team members assignments. We have enumerated list of tasks, constraints and the resources needed for completion of the project scheduled for development.

3.1 Introduction

3.1.1 Project Overview

The advancement of Intelligent Transportation Systems (ITS) requires high quality traffic information in real-time. Traffic data can be taken from different sensors such as loop detectors, pneumatic sensors etc. As an alternative, a camera based data collection system in which the output of the cameras installed along the roadways are processed. This processed data can be used to analyze the speed of the vehicle with the help of two algorithms. Here we will only focus on differential methods which are Lucas Kanade and Kalman Filter. Differentiating among the above two algorithms, the most efficient one will be chosen.

3.1.2 Project Deliverables

The system offers the following modules

1. **Dataset of vehicular images**
   The list of documents needed in this dataset are the images which can be inserted. User wont be able to see or have an access to the filesystem thus no user module present here. Technical module involves steps taken to retrieve the data stored for further usage.

2. **Dataset of videos**
   The list of documents needed in this section will have the videos of vehicles

taken in different environments. These videos will be stored as datasets into the file system. Technical module involves the steps taken to retrieve the data for further use.

3. **Speed Calculator**
   The list of documents needed in this section are of the algorithms which are going to be used i.e. Kalman Filter and Lucas Kanade. The system manual will have to specify the window size of the object and from that the parameters will be taken for calculations. Technical module involves calculation of the speed with the parameters specified.

3.2 Project Organization

3.2.1 Software Process Model

The model being followed for the implementation of our project is the SPIRAL MODEL, a risk-driven process model generator. A combination of Waterfall and Iterative model where each phase begins with a design goal and ends with the client reviewing the progress. Spiral model is quite flexible with the new addition of functionality to the project at later stages and the risk analysis of the new functionality is calculated simultaneously without any interruption to the ongoing cycle modelled. Below is the diagram of the Spiral Model depicting its variant stages of execution.

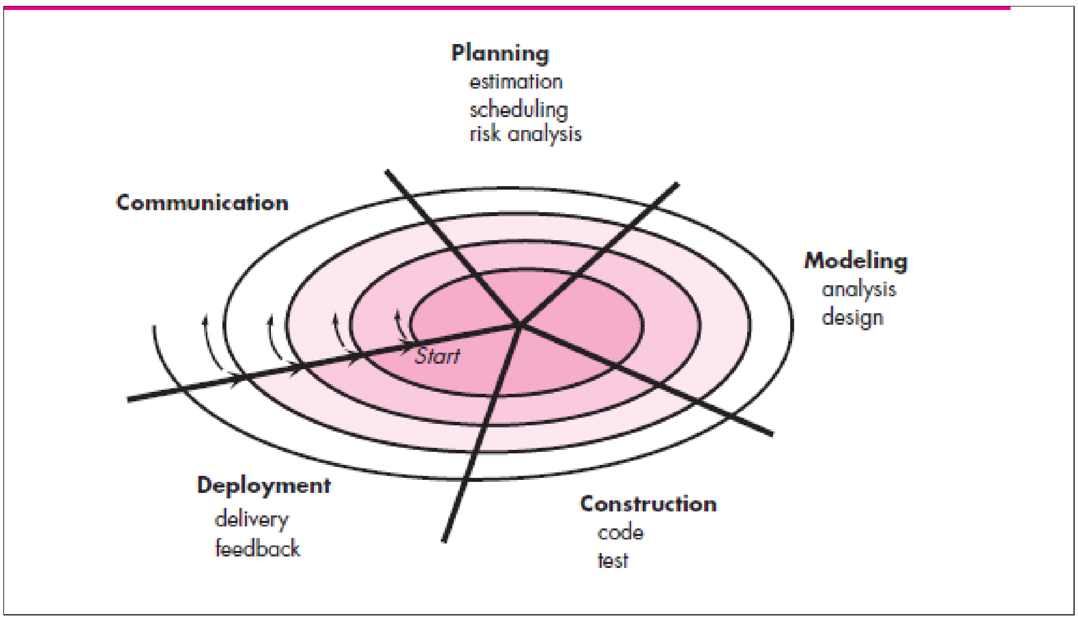


Figure 3.1: The Spiral Model

3.2.2 Roles and Responsibilities

Names	Responsibilty
Sehab Mankada Vidhi Chitalia Nidhi Gada	Models,designs and creates the database and tables used by the software.
Sehab Mankada Vidhi Chitalia Nidhi Gada	Has authority to manage the project which includes leading the planning and the development of all project deliverables managing the budget and schedule and all project management procedures(scope management,issues management,risk management etc)
Sehab Mankada Vidhi Chitalia Nidhi Gada	Understands the business requirement and design a solution that will meet the business need.Determines the overall model and framework for the solution,down to the level of designing screen,reports programs and other component.
Sehab Mankada Vidhi Chitalia Nidhi Gada	Responsible for actual building of the solution.
Sehab Mankada Vidhi Chitalia Nidhi Gada	Ensures that the solution meets the business requirements and that is free of error and defects.

Table 3.1: Roles and Responsibility

3.2.3 Tools and Technology

1. **Operating System**
   Windows 7, 8, 10

2. **Platform**
   MATLAB 7.8/9.2

3. **Database**
   MATLAB File System

3.3 Project Management Plan

3.3.1 Task 1-Requirement Gathering

Description
The document specifies the requirements of the software in detail like hardware requirements, DB requirements, Software requirements etc.
Deliverable and Milestones
The deliverable is an SRS document. After being reviewed by the project guide and finalized, the revised SRS has becomes a milestone.
Resources needed
We used Latex for documentation. Internet has been the basic requirement.
Risks and Contingencies
The SRS has to be revised when there was any change in the requirement

3.3.2 Task 2- Feasibility Study

Description
The document is a layout of the services to be offered to the civilians of the society and how they will be delivered.
Deliverable and Milestones
Requirement analysis, Design Document and source codes are the deliverable.After being reviewed by the project guide and finalized, these revised documents become a milestones.
Resources needed
We used Latex for documentation. Internet has been the basic requirement.
Risks and Contingencies
Changing requirements, incomplete requirements and lack of resource tools are the risks involved.

3.3.3 Task 3- Project Details

Description
The document is a brief of the whole project which involves documentation of all the modules involved in the project.
Deliverable and Milestones
SRS, SPMP,STD and SDD are the deliverable. After being reviewed by the project guide and finalized, these revised documents become a milestones.
Resources needed
We used Latex for documentation. Internet has been the basic requirement.
Risks and Contingencies
Changing requirements, incomplete requirements and lack of resource tools are the risks involved.

3.3.4 Task 4- Scope

Description
The document is an expansion of the thoughts we put in while thinking about the project- where it will be used.
Deliverable and Milestones
SRS and SPMP are the deliverable. After being reviewed by the project guide and finalized, these revised documents become a milestones.
Resources needed
We used Latex for documentation. Internet has been the basic requirement.
Risks and Contingencies
Changing requirements, incomplete requirements and lack of resource tools are the risks involved.

3.3.5 Task 5- Software Requirement Specification

Description
The document is a summary of the project requirements, software product features, software attributes and the tools and technologies.
Deliverable and Milestones
The document specifies the in depth explanation of the above components.After being reviewed by the project guide and finalized, these revised components in the document become a milestones.
Resources needed
We used Latex for documentation. Internet has been the basic requirement.
Risks and Contingencies
Changing requirements, incomplete requirements and lack of resource tools are the risks involved.

3.3.6 Task 6- Coding Module 1

Description
The document specifies the design and coding of the module 1.
Deliverable and Milestones
The document will have the full code with the expected output.This document will be reviewed by the guide and finalized after which it becomes a milestone.
Resources needed
MATLAB
Risks and Contingencies
The code can be caught with bugs which will hamper the normal execution flow of the code.

3.3.7 Task 7- Verification Module

Description
This task was carried out throughout the project. As interface was developed for the user to choose various options and for displaying the output result.

Deliverable and Milestones
As each feature was finalized it was integrated with the interface. A feature added to the interface after completion will become the deliverables. The final output after integration of all features will become a milestone.
Resources needed
MATLAB
Risks and Contingencies
This task can show the output result only after the speed is calculated .

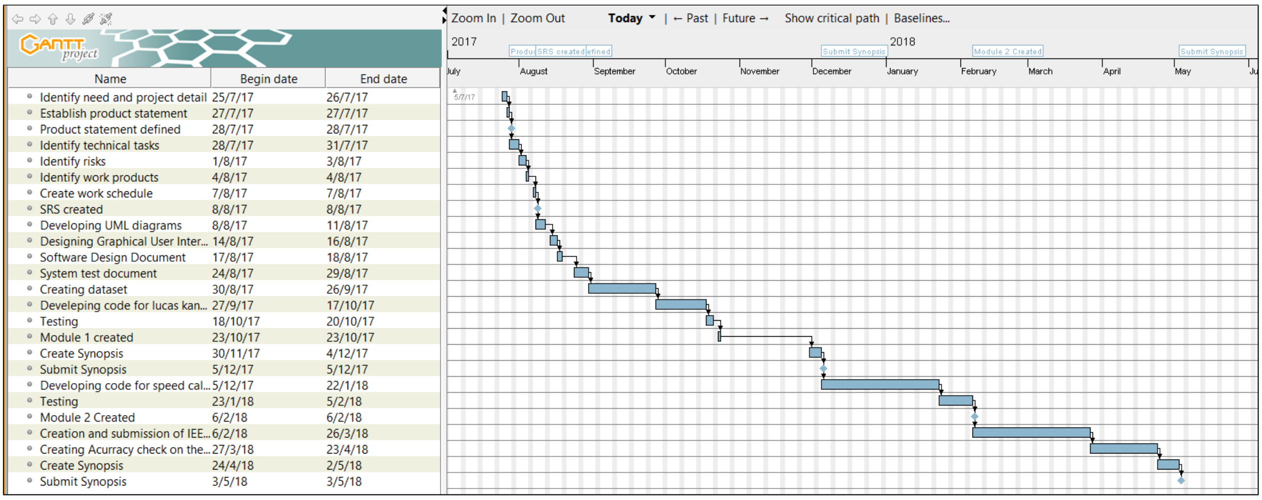
3.3.8 Task 8-Software Design Document

Description
This document consist of detailed design of software consisting of architectural diagram, class diagram, flowchart and other diagrams needed to clarify the flow of information and working of product.
Deliverable and Milestones
Deliverable is the document made by team members. After being reviewed by project guide, the revised document becomes a milestone.
Resources needed
Rational rose, Microsoft office, Internet and Browser was also required
Risks and Contingencies
The design for the software had to be changed when there was a change in the requirement for the final product.

3.3.9 Task 9- Software Test Document

Description
After all the above tasks are completed, the product will be able to calculate the speed and process it accordingly.
Deliverables and Milestones
The deliverables are a list of features not working as expected, changes required.The final interface after incorporating the changes and resolving errors becomes a milestone.
Resources needed
Testing Tools, Test cases
Risks and Contingencies
This task can be carried out only when all above tasks are completed.

3.4 Timetable



Name	Begin date	End date
Identify need and project detail	25/7/17	26/7/17
Establish product statement	27/7/17	27/7/17
Product statement defined	28/7/17	28/7/17
Identify technical tasks	28/7/17	31/7/17
Identify risks	1/8/17	3/8/17
Identify work products	4/8/17	4/8/17
Create work schedule	7/8/17	7/8/17
SRS created	8/8/17	8/8/17
Developing UML diagrams	8/8/17	11/8/17
Designing Graphical User Inter...	14/8/17	16/8/17
Software Design Document	17/8/17	18/8/17
System test document	24/8/17	29/8/17
Creating dataset	30/8/17	26/9/17
Develeping code for lucas kan...	27/9/17	17/10/17
Testing	18/10/17	20/10/17
Module 1 created	23/10/17	23/10/17
Create Synopsis	30/11/17	4/12/17
Submit Synopsis	5/12/17	5/12/17
Developing code for speed cal...	5/12/17	22/1/18
Testing	23/1/18	5/2/18
Module 2 Created	6/2/18	6/2/18
Creation and submission of IEE...	6/2/18	26/3/18
Creating Acurracy check on the...	27/3/18	23/4/18
Create Synopsis	24/4/18	2/5/18
Submit Synopsis	3/5/18	3/5/18

Figure 3.2: Gantt Chart

Chapter 4

Software Requirement Specification

This chapter discusses about the requirements of the software in detail. In this chapter, we will discuss about the functional and non-functional requirements, different interfaces required and the protocols used. Also it will give details about the features that the software will cater.

4.1 Introduction

4.1.1 Product Overview

The advancing of Intelligent Transportation Systems (ITS) requires high quality traffic information in real-time. Traffic data can be taken from different sensors such as loop detectors, pneumatic sensors etc. Although they provide accurate and reliable data, their usage is limited because of their limited coverage, implementation and maintains cost and higher possibility for breakage. As an alternative for sensor based data collection, there is camera based data collection in which the output of the camera installed along the roadways are processed. This processed data can be used to analyze the speed of the vehicle with the help of two algorithms. Differentiating among the above two algorithms, the efficient one will be chosen.

4.2 Specific Requirements

4.2.1 External Interface Requirements

1. **System Interface**
   (a) Calculated Speed
   (b) Moving Vehicles
   (c) Window Frame
2. **Hardware Interface**

The selection of hardware is very important in the existence and proper working of any software. When selecting hardware, the size and requirements are also important. Good Resolution Mobile camera is needed for a better capture of photo.

3. **Software Interface**
   (a) Operating System : Windows 7/8/10 32 bit 64 bit
   (b) Back- End : MATLAB 7.8/9.2
4. **Communication protocols**
   (a) Processor : i3/i5/i7, +2 GHz
   (b) RAM : 4 GB
   (c) Hard Disk Drive : 20 GB

4.2.2 Software Product Features

Functional Requirements

1. **Detecting moving vehicles**
   The module requires to detect the moving vehicles under the readings of the cameras installed at the traffic signal. The camera installed will be a high definition camera as image with the best pixels a will be captured through it. While for video streaming, a good camcorder of higher pixel value will be used to record the moving vehicles.
2. **Tracking moving vehicles**
   The next module to be implemented is the tracking of the moving vehicles. After successfully capturing the images and recording the video, a special attention is needed to be given to the movement of the vehicles moving swiftly. A window size needs to be defined to track the vehicle under those considerations, if the vehicle is not visible then tracking will be difficult and in such case the window size needs to be altered.
3. **Speed calculation**
   The module which calculates the speed of moving vehicles by capturing images and also by video streaming. Speed is thus assessed by using the normal Speed formula. The actual speeds of the vehicle will then be compared with the calibrated readings to find out the best one among the two.
4. **Specified limited speed for moving vehicles**
   This module determines the speed limits of all moving vehicles found on the road. Different roads will have different speed limits and so the threshold speeds for all the vehicle type will be defined previously in the database. These values will then be compared with the calibrated readings to find out whether the vehicles have violated the specified threshold speed.

4.2.3 Software System Attributes

1. **Reliability**
   The two algorithms are reliable only upto certain extent i.e 60 percent and 70 percent respectively. The data for the images will be viable upto 1GB and the dataset for video streaming will be done at a time which will be supported upto 10MB.

2. **Availabilty**
   As the system contains a mammoth of data in terms of images and videos and processing them to get the data will take some time, it will be available for use to just one customer at a time.The system will be available for use 24x7 regardless of the recovery period which can occur due to some failures in the system.

3. **Security**
   Security refers to the overall completeness, accuracy and consistency of data. All the data regarding the speeding vehicle requires special attention by the developer so it's advisable to store them safely and with great security. Incase of the breach of security, it can be indicated by the absence of alteration between two instances, meaning the data is intact and unchanged.

4. **Maintainability**
   Datastore should be well taken care as the main modules like image captures and video streams are all stored in it. Maintainability is one of the important constituent of the software system as this will be the central module where all the results will be shown after the execution of the two algorithms. So, datastore will require high maintenance.

5. **Performance**
   Performance of the system is of utmost importance and no leniency should be granted in this regard. The system response time should be fast and minimum time lag is expected. It should handle errors efficiently. Lastly, the error rate of the speeds of the vehicles calibrated should boil to the minimum for efficient results.

4.2.4 Database Requirement

MATLAB supports the file system where all the files and folders can be stored on the target computer. The functions that work with it must have files that always accept the full paths to those as file inputs. If we don't specify the full path, then MATLAB looks for files in the current folder first and then on folders on the search path. The file system consists of the following things:

1. **Vehicle Images/Video Streams**
   The images are captured from the normal mobile camera but should have good resolution. The burst shot mode on the camera will allow us to click multiple images of a single moving vehicle. Similarly, videos of vehicles are captured and stored in a file system which can be used while the coding module is being implemented.

2. **Threshold Speeds**
   Speeds of different road environment and vehicle types will need to be added. In real life too vehicles moving on normal streets have different speeds compared to when they are moving on expressways and highways. This will notify us that threshold speeds will be different for all the road environment along with the vehicles.

3. **Speed Calculator**
   Speed calculation is the important module considered in the implementation. Speed of various vehicles will be calculated with the two algorithms and displayed in this chart.

4. **Accuracy and Error Measurement**
   The error rates of the speed of vehicles will be calculated and their accuracy's will be displayed alongside in a table. No algorithms proposed have 100 percent accuracy and so to combat with the maximum efficacy- the accuracy and error measurement will be the attribute to look out for in the database as this will tell us which one produces the least error measurement and is the most efficient.

Chapter 5

Software Design Description

This chapter discusses about the design overview along with the requirement traceability matrix. We have specified different diagrams required for the completion of the project. The flow diagrams and the ER diagrams are drawn and showcased for better understanding the project.

5.1 Introduction

5.1.1 Design Overview

In this section, the steps taken to calculate the speed of a moving vehicle are explained. The algorithms i.e. Kalman Filter and Lucas Kanade are examined. The parameters such as Optical Flow Vector, Pixels captured inside the Window,Centroid Matrix storage etc. are considered. The input (Picture, Video) to the system is analyzed and its output is compared with the error rate to check its efficiency.

5.1.2 Requirement Traceability Matrix

A traceability matrix is created by associating requirements with the work products that satisfy them. The configuration management plan is to show how changes will be tracked and controlled. Traceability is a key part of managing change. It ensure completeness that all lower level requirements come from higher level requirements and that all higher-level requirements are allocated to lower ones. Traceability also provides the basis for test planning

Required Component	**System Interface**	**Database**	**Error Calculation**	**Result**
Detecting moving vehicles	X		X	X
Tracking moving vehicles	X		X	X
Speed for Kalman Filter		X	X	X
Speed for Lucas Kanade		X	X	X

Table 5.1: Requirement Tracebility Matrix

5.2 System Architecture Design

5.2.1 Chosen System Architecture

Database-Centric Architecture

Database-Centric Architecture or Data-Centric architecture is related to software architectures in which database play a crucial role. In this the data model precedes with the implementation of any given application and will be around for long after it is gone.

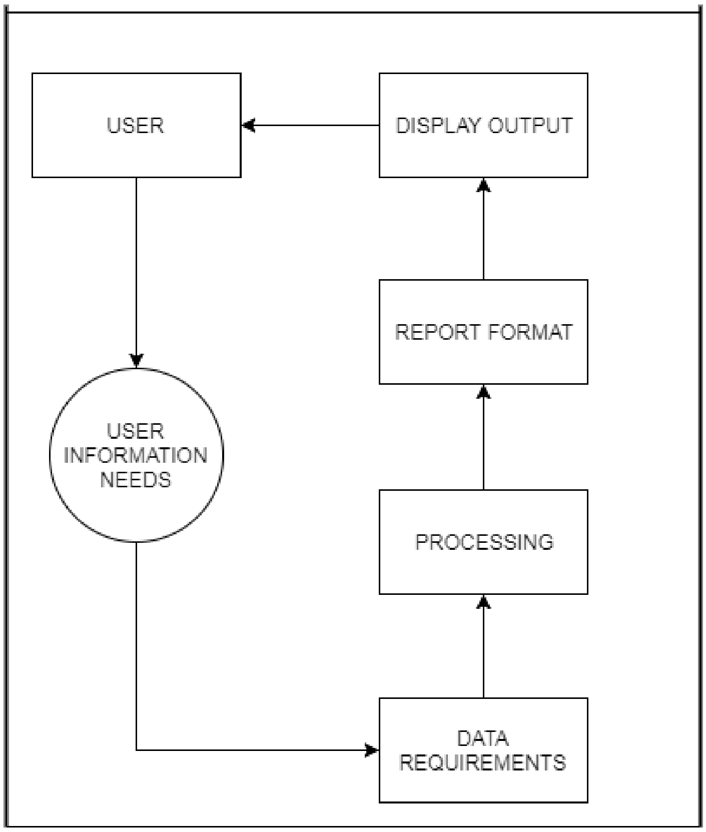


Figure 5.1: Data-Centric Architecture

ER Diagram

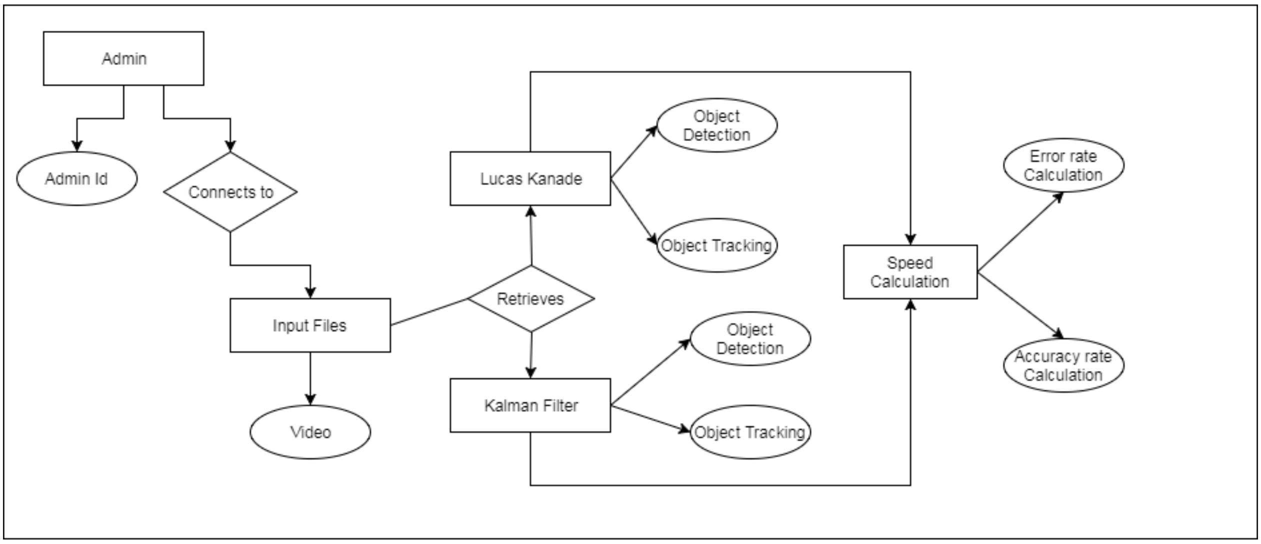


Figure 5.2: ER Diagram

5.2.2 Discussion of Alternative Designs

Event-Driven Architecture (EDA)

Event-Driven Architecture (EDA), is a software architecture pattern promoting the production, detection, consumption of, and reaction to events. EAD is a style of application architecture can be implemented in any language, which can improve responsiveness, throughput and flexibility. Also, EDA is most effective when existing related stored data is made available.

5.2.3 System Interface Description

It is a stand-alone application which calculates the speed of the moving vehicles.

5.3 Algorithmic Approach

Lucas Kanade

The proposed algorithm is presented here. Before any operation a scene should be selected from a static camera. Our test movies are selected from a normal camera. Some pre-processing operations have to be done for making the scene ready to process. Due to the camera's auto white balance and the effect of sudden environment intensity changes, the mean of every frame is calculated on gray-scale format. The optical flow estimation is the essential part of the algorithm which is executed next. Filtering speckle and general external noises induced due to weather conditions is one of the most important sections of the procedure. Median filter is used in our framework. During filtering operation, some holes are created in the frames. To fill these holes and prevent detection mistakes morphological closing is implemented. Now the motion objects Optical Flow Based Moving Object Detection and Tracking for Traffic Surveillance are detected, but many of them are not interested. The pedestrians or waving flags are the example of these unwanted motions. Blob analysis helps us to cluster objects which cannot be cars based on blob sizes Bounding boxes around the vehicle.

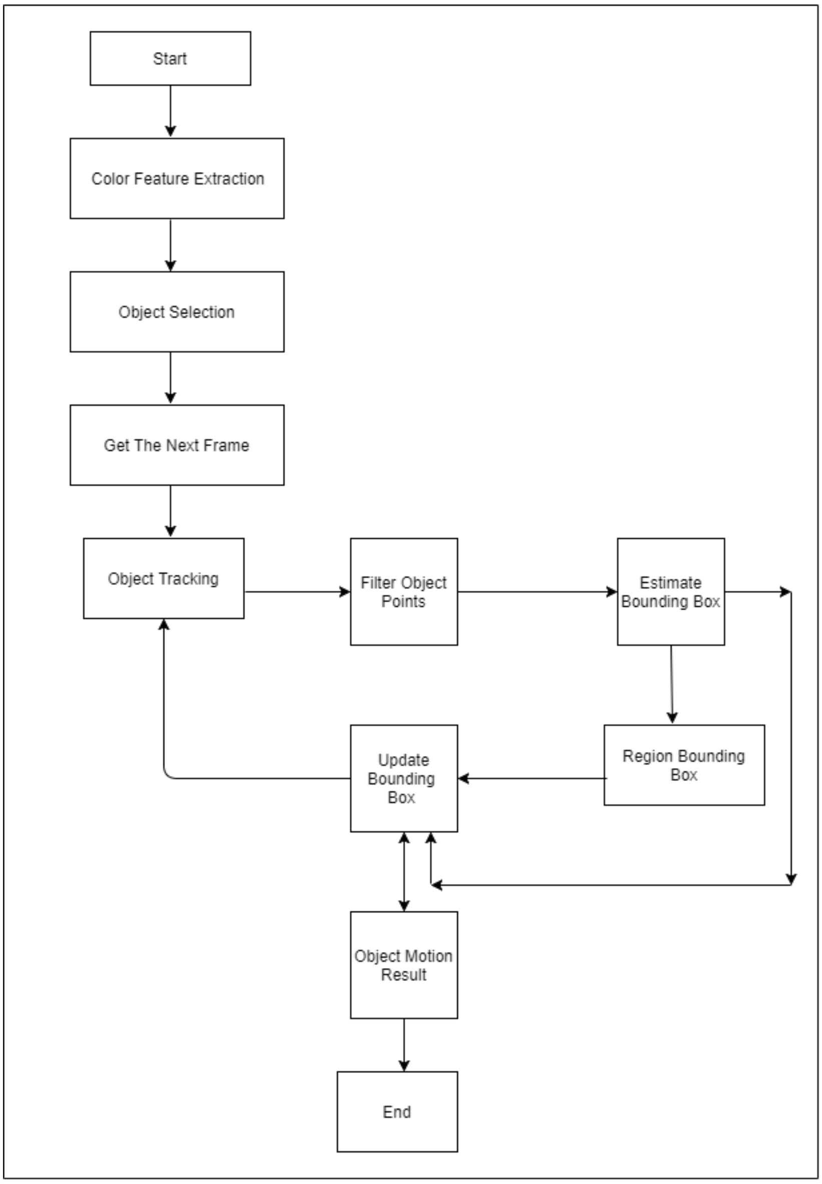


Figure 5.3: Lucas Kanade

Kalman Filter

In the proposed algorithm, object tracking is performed by predicting the object's position from the previous information and verifying the existence of the object at the predicted position. Secondly, the observed likelihood function and motion model must be learnt by some sample of image sequences before tracking is performed. The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process in several aspects: it supports estimations of past, present, and even future states, and it can do the same even when the precise nature of the modelled system is unknown. The Kalman filter estimates a process by using a form of feedback control. The filter estimates the process state at some time and then obtains feedback in the form of noisy measurements.

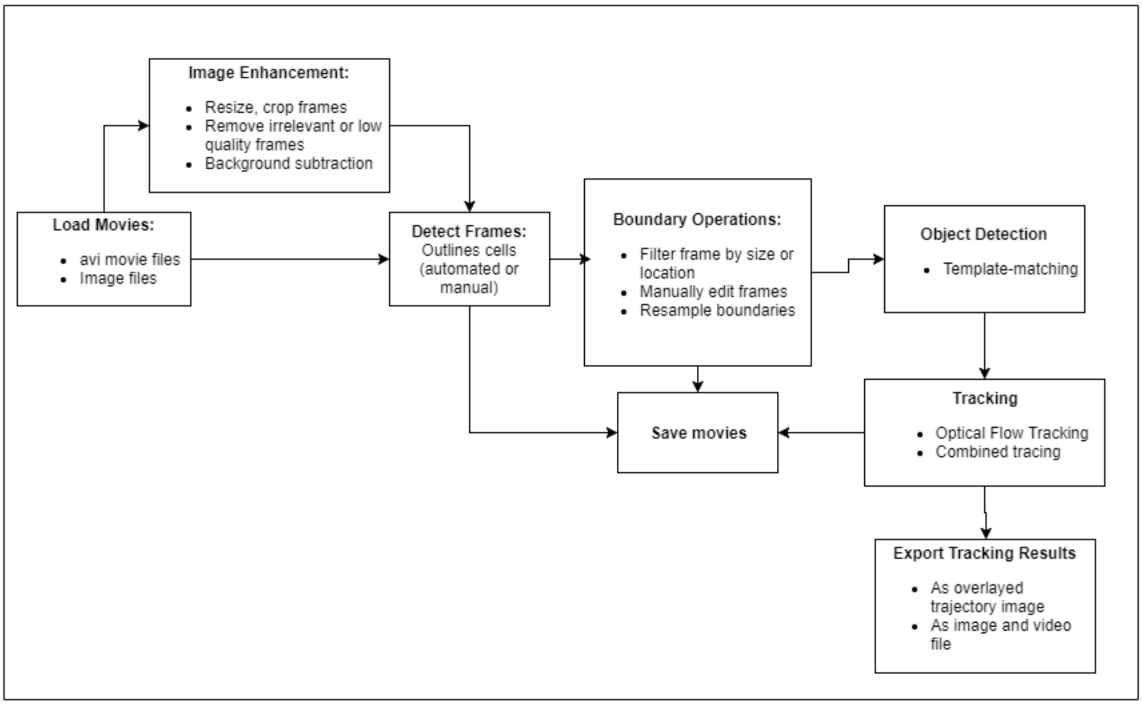


Figure 5.4: Kalman Filter

Chapter 6

Software Test Document

This chapter provides information about the tasks which are tested and the successful results of the testing are used to calculate the speed of moving vehicles with least error results.

6.1 Introduction

6.1.1 System Overview

The system involves calculating the speed of moving vehicles by capturing images and videos of moving vehicles. Images and videos will help us stimulate the speed of the vehicle. This is done by applying two algorithms named Lucas Kanade and Kalman Filter. Various parameters will be considered and the one's which give the minimum error rate will be selected and showcased.

6.1.2 Test Approach

Tests will be conducted to check the efficiency of the software. The reason for this test is to check for any errors and limitations. A list of various planned test are as follows.

System Testing

The entire system will be tested as per the requirements to ensure that there is proper interaction with the database, there is proper input given to the software and that the output generated is relevant.

White Box Testing

White box testing focuses on the program control structure. In this project, when the user selects the vehicle, it's data stored in database is retrieved and the calculations performed yields an accurate result is checked.
White Box Testing (also known as Clear Box Testing, Open Box Testing, Glass Box

Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs.

Performance Testing

Performance test will be conducted to check whether the software is reliable and whether it gives the output within a given time. Test will also be conducted to check whether the algorithms gives an exact result.

Black Box testing

Black box testing is a method of software testing that's tests the functionality of application as opposed to its internal structures of workings (see white box testing).Specific knowledge of application's code/internal structure and programming knowledge in general is not required. The tester is only aware of what the software is supposed to do, but not how i.e. When he/she enters a certain input, he/she gets a certain output; without being aware of how the output was produced in the first place. Test cases are built around specifications and requirements i.e. what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements and design to derive test cases.

Unit Testing

In Unit testing, attention is diverted to individual modules, independently to one another to locate errors in coding and logic. Each module is independently checked for errors and these errors are to be rectified to ensure the effectiveness of performing its respective functions.
In our project each units/modules (speed calculation, locating the vehicle, comparison of accuracy of the speed) were tested separately. The idea is that if each individual component is tested and works correctly. Also, while testing in each and every module many bugs were detected and resolved individually without affecting other modules workflow.

Integration Testing

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. In this project, we have used two algorithms viz. Kalman Filter and Lucas Kanade which calculates the speed of a moving vehicle with video and pictures respectively. Integration testing is conducted to expose defects in the interfaces and inter-action between integrated components. Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested.
The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. Test cases are constructed to test whether all the components within modules interact correctly.The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages.

6.2 Test Plan

6.2.1 Test Plan Objectives

- Testing is the process of executing the program the program with the intention of finding an error.
- A good test is one that has high probability of finding an as yet undiscovered.
- A successful test is that which uncovers as-yet-undiscovered error.

6.2.2 Features To Be Tested

1. Lucas Kanade Algorithm
2. Kalman Filter Algorithm
3. Juxtaposition between the algorithm

6.2.3 Features Not To Be Tested

- Implementation of the algorithm

6.2.4 Testing Tools and Environment

MATLAB provides testing tools which contains various frameworks like Function-based and class-based unit tests,Extend-unit,performance,mocking testing framework.

6.3 Test Cases

6.3.1 Check Connection-TC1

Test Case Description
Check the working of file system and successful implementation of the project software.
Input
Start the Application
Expected Output
Application runs properly without any error.
Actual Output
Application working properly
Pass or Fail Criteria
Pass

6.3.2 Uploading Moving Vehicle Images-TC2

Test Case Description
Inserting the images of the moving vehicles.
Input
Images
Expected Output
The file system showing images of the vehicles with their specifications.
Actual Output
The database showing images of the vehicles with their specifications.
Pass or Fail Criteria
Pass

6.3.3 Saving images in the file system-TC3

Test Case Description
The captured images are inserted in the file system.
Input
Image of a Vehicle
Expected Output
The images are secured in the file system.
Actual Output
The images are saved successfully and are available for the purpose of retrieval.
Pass or Fail Criteria
Pass

6.3.4 Detecting a Moving Vehicle-TC4

Test Case Description
The window selected for taking pictures should have some movement of vehicles which can be captured.
Input
Vehicular Image
Expected Output
Moving Vehicle is detected.
Actual Output
The vehicle is detected within the window frame.
Pass or Fail Criteria
Pass

6.3.5 Video Selection Testing- TC5

Test Case Description
The best video needed for the speed calculation is selected from various videos.
Input
File system is used for video selection.

Expected Output
The video is been selected and is successfully played.
Actual Output
A particular video is selected and further proceeds with the speed calculation.
Pass or Fail Criteria
Pass

6.3.6 Speed Calculation for Lucas Kanade Algorithm-TC6

Test Case Description
The Lucas–Kanade method assumes that the displacement of the image contents between two nearby instants (frames) is small and approximately constant within a neighborhood of the point p under consideration which will calculate the speed.
Input
Nearby instants (frames).
Expected Output
The calculated speed is displayed.
Actual Output
The calculated speed is displayed.
Pass or Fail Criteria
Pass

6.3.7 Speed Calculation for Kalman Filter Algorithm-TC7

Test Case Description
The Kalman Filter method optimally uses a series of measurements observed over time containing statistical noise and other inaccuracies to calculate the speed.
Input
Videos
Expected Output
The calculated speed is displayed.
Actual Output
The calculated speed is displayed.
Pass or Fail Criteria
Pass

6.3.8 Error Calculation-TC8

Test Case Description
The error will be generated after the calculation of the speed and it's accuracy rate which indeed will showcase the discrepancy rate in the algorithm.
Input
Parameters considered for calculating speed.
Expected Output
Error rate will be known.

Actual Output
The error rate will be made known in proper manner.
Pass or Fail Criteria
Pass

Test Case ID	Test Case	Description	Input	Expected Output	Actual Output	Pass/-Fail
1	Check Connec-tions	Successful implementation of the project software	Start the Application	Application runs properly	Application working properly	Pass
2	Uploading Moving Vehicle Images	Inserting images of moving vehicles	Images	The file system showing list of images and their speci-fications	The database showing images in a proper format	Pass
3	Saving images in the file system	The captured images are inserted in the file system	Image of a Vehicle	The images are secured in the file system	The images are saved success-fully	Pass
4	Detecting Moving Vehicle	Visibility of the vehicle is necessary in the window frame	Vehicular Image	Moving Vehicle is detected	The vehicle is detected within the window frame	Pass
5	Video Selection Testing	The best video needed is selected from file system	File system is used for video selection	The video is been selected and is suc-cessfully played	Video is selected and proceeds with the next step	Pass
6	Speed Calculation for Lucas Kanade Algorithm	The algorithm calculates sped between two nearby instants	Nearby instants (frames)	The calculated speed is displayed	The calculated speed is displayed	Pass
7	Speed Calculation for Kalman Filter Algorithm	Calculates the speed of the vehicles with processing the video frames	Videos	The calculated speed is displayed	The calculated speed is displayed	Pass
8	Error Calculation	The error is calculated on looking at the output of the algorithm	Parameters considered for calculating speed	Error rate will be known	The error rate will be known in proper manner	Pass

Table 6.1: Test Cases

Chapter 7

Conclusion

From the results that we obtained, combining the computation of partial derivative in lucas kanade algorithm with smooth filtering, the processing time decreases in a better optical flow.The best parameter to represent each of the algorithm is concluded based on the factors which are segmentation, motion vector and the processing time. In lucas kanade algorithm the optimized window size will be of [25,16].
From the experimental results,we have found that the system works well with good accuracy as the error rate is very low. We hope that this system will find extensive applications in real time traffic management field and many other purposes.
After executing both the algorithms(Lucas Kanade and Kalman filter), we conclude that Kalman filter algorithm is more accurate than lucas kanade for calculating the speed of the vehicles.

Chapter 8

Implementation

This chapter exhibits the implementation details about all the approaches which have been employed throughout the development of the project.

There are two algorithms used named Lucas Kanade and Kalman Filter which have been used throughout the execution of the project. But along with these two algorithms there are small other techniques that have been adopted.

Optical Flow

It is used to compute the motion of pixels of an image sequence. Optical flow is the distribution of the apparent velocities of objects in an image. By estimating optical flow between video frames, we can measure the velocities of objects in the video. In general, moving objects that are closer to the camera will display more apparent motion than distant objects that are moving at the same speed. It provides a dense (point to point) pixel correspondence. Correspondence problem determines where the pixels of an image at time *t* are in the image at time *t+1*.

Horn Schunck Method

This is the most fundamental optical flow algorithm. It assumes smoothness in the flow over the whole image. This method tries to minimize the distortions in flow and prefers solutions which show more smoothness. The motion associated to each pixel (x,y) of an image *I* can be modeled as: $I(x,y,z) = I(x+u\delta t, y+v\delta t, t+\delta t)$

Segmentation/Centroid Method

Segmentation means decomposition of the image under study into its different areas of interest. Particle segmentation is used to separate the target (object of interest) from background, when target is not fully visible. The pixel intensities are discretized into gray levels along with the segmentation that can be done in two steps:

1. The gray level image is transformed into binary image using lower and upper threshold limits of the target. These thresholds of target can be determined through the pixel intensity and its surroundings.

2. The pixels which fall within this target layer limits (called pixel detection) are computed.

Gaussian Mixture Model

Gaussian mixture models (GMM) are composed of k multivariate normal density components, where k is a positive integer. Each component has a d-dimensional mean (d is a positive integer), d-by-d covariance matrix, and a mixing proportion. Mixing proportion j determines the proportion of the population composed by component j, j = 1,...,k.

Execution Images of Lucas Kanade Algorithm

The figure below shows a snapshot of a particular video frame. Here the original video frame (RGB) is converted to gray-scale. While executing the algorithm we come across the unwanted noise in the background. When we try extracting the background all these will come into the picture. To minimize these noise in the video frame we use filtering methods. A foreground is created to show all these things in the frame.

Figure 8.1: Lucas Foreground

The figure below depicts the background which is refurnished by using the filtering methods. Filtering is the method which is used to extract the background from these noise which creates holes in the frames. To fill these holes and get the appropriate output we use Median Filtering that avoids the morphological mistakes while executing. The foreground created is cleaned to its maximum limit which is given by defining the intensity.

Figure 8.2: Lucas Clean Foreground

The figure down shows the detection of all the cars in the entire video. Bounding boxes (blue color lines) around the objects have been created to depict the detection. The shadow of the cars, white lines etc have all been detected in the video due to the intensity issues in the filtering methods.

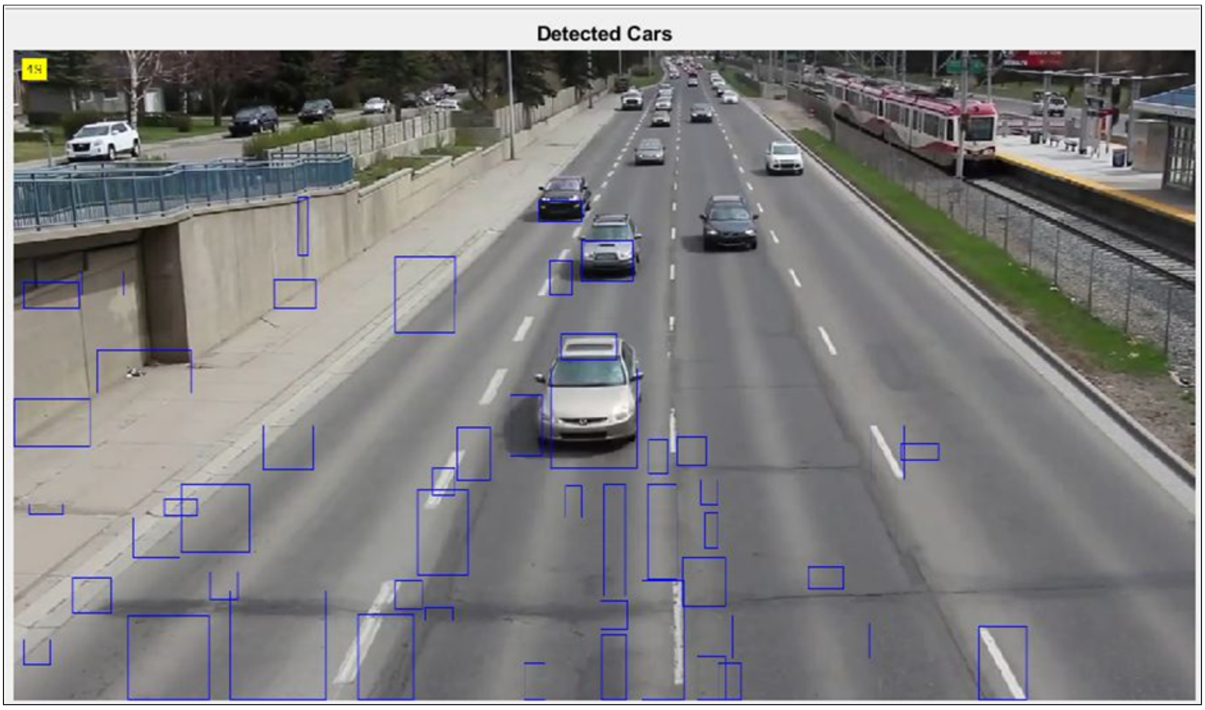


Figure 8.3: Lucas Detected Cars

Below is the snapshot taken from the video which shows the speed of the vehicles in kmph. The yellow circle is the one which shows the speed of the vehicles. Lucas Kanade algorithm is applied for the detection and tracking of the vehicles in the frame. Later the speed of the vehicle is calculated using the speed distance time which is as follows: Speed = Distance/ Time

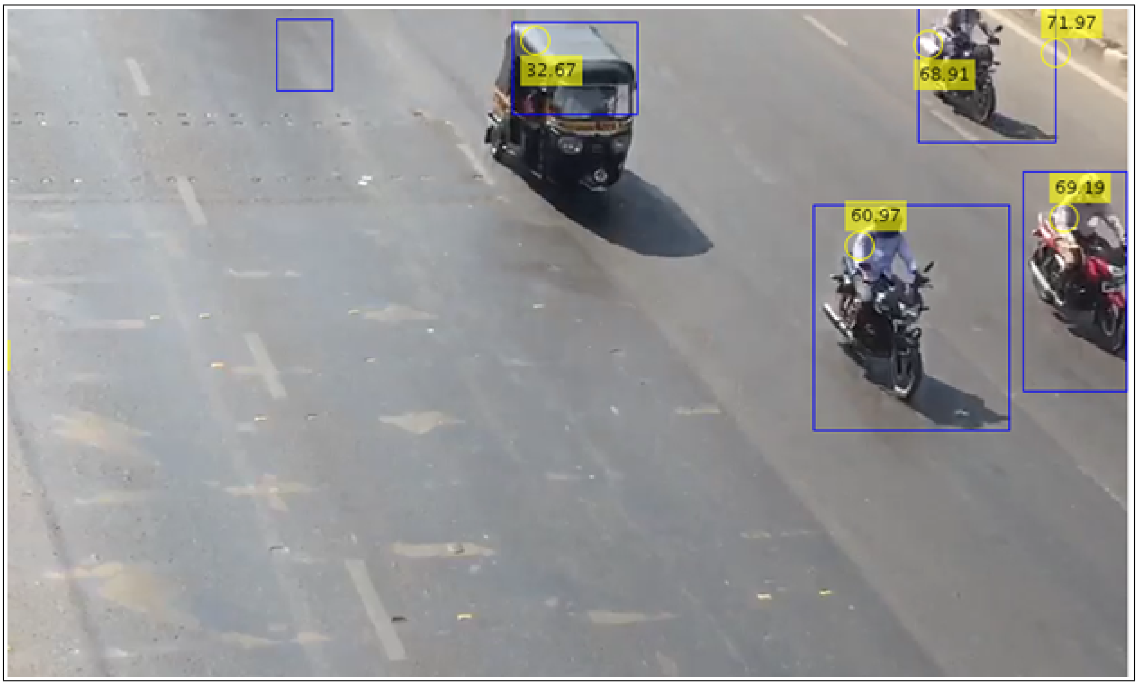


Figure 8.4: Speed

Execution Images of Kalman Filter Algorithm

The figure below shows the detection of vehicles in the gray scale. The video frame captured is converted to grayscale so that there is effective detection of the vehicles in the frame. In this algorithm, the foreground detection is omitted as the inherent feature of the Kalman Filter algorithm is to provide the smoothness to the background detection while removing the noise in the surroundings. Here, the vehicles are detected by bounding boxes (yellow lines) around them. No filtering methods applied externally due to the accurateness of the algorithm in detecting of moving objects.

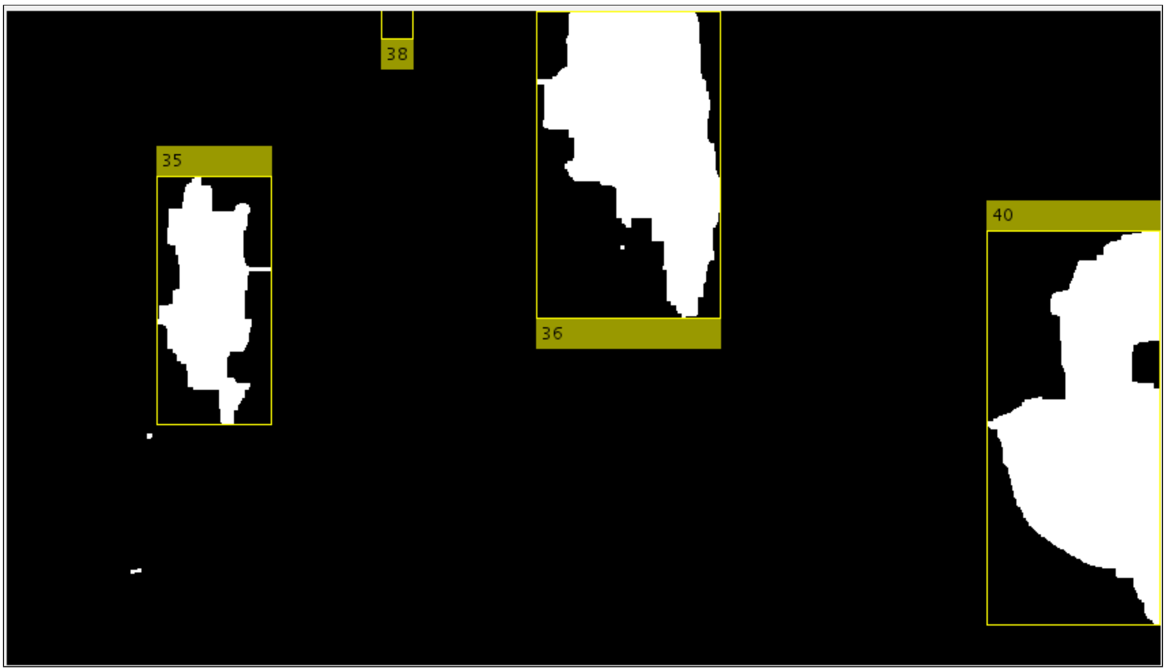


Figure 8.5: Kalman Foreground

The figure below shows the tracking of the vehicles in the video frame. The video is played and the vehicles are tracked. The tracking takes place within the window of the screen till the time the vehicle is in sight. The bounding boxes are made around the vehicles which accurately tracks the vehicle motion. Kalman filter is a technique of detection of the objects where only the objects are tracked and the background or the morphological noise is avoided completely, giving the effective results.

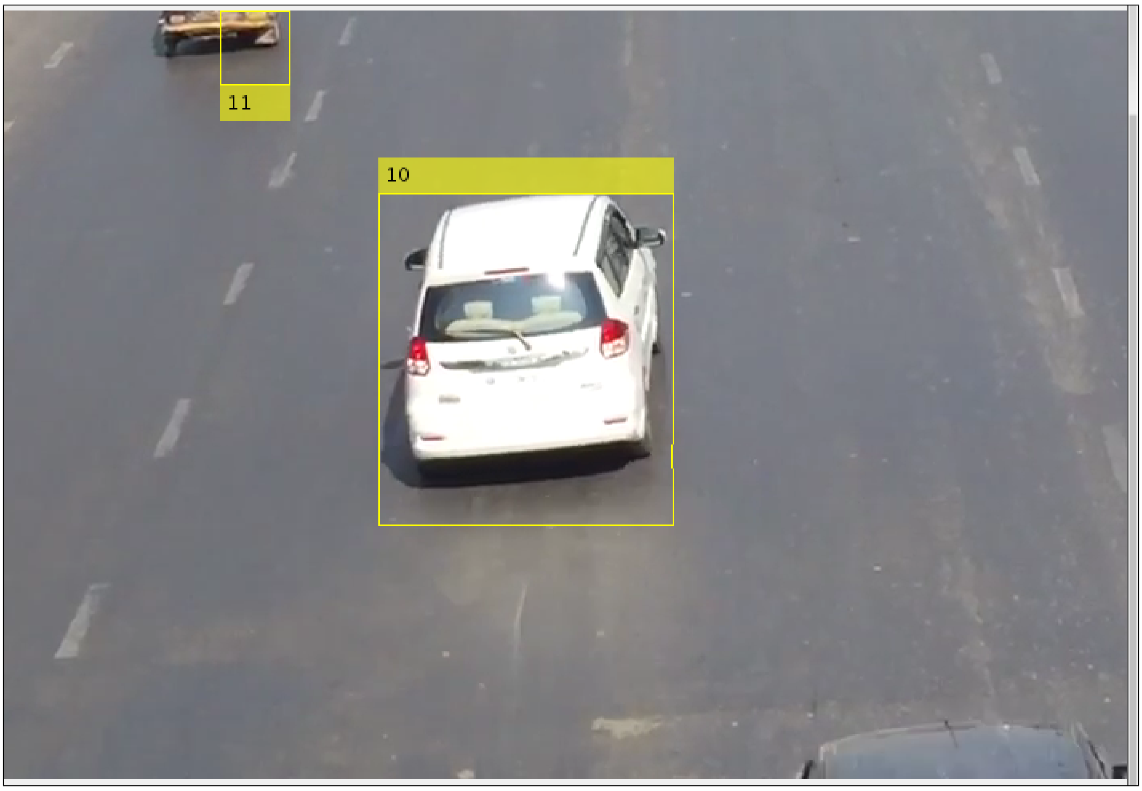


Figure 8.6: Kalman Vehicle Detection

The figure below is a snapshot of all the speeds which have been calculated. The Kalman filter algorithm is used for detection and tracking of algorithms which is later used for speed calculation. The input to the speed calculation module is so clean and viable that the error percentage is minimized and hence the calculation of speed is pretty accurate. The video is played in the background and the calculated speed is displayed in the table format.

```
Speed
  94.1291
Speed
  85.5000
Speed
  62.9025
Speed
  40.3051
Speed
  34.4025
Speed
  28.5000
Speed
  28.5000
Speed
  40.3051
```

Figure 8.7: Kalman Speed Calculation

Chapter 9

Results and Future Discussions

9.1 Result

Table of the result obtained during the entire process

Live Video	**Car**	**Speed By lucas Kannade (kmph)**	**Error(%)**	**Speed By Kalman Filter(kmph)**	**Error(%)**
1	1	57.33	25	57.7	20
	2	62.00	11	36.1	8
	3	51	10	26.12	9
2	1	59.89	18	31.57	16
	2	77.30	11	36.98	10
	3	62.00	15	78.46	13
3	1	40.98	30	159	25
	2	54.67	20	78.46	19
	3	77.21	11	56.92	13

Table 9.1: Result

9.2 Future Scope

This system centers around computing the speed of vehicles by figuring their accuracy, efficiency and unwavering quality. For better outcomes, we can broaden the above proposed system by building a database having the individual personal details of the drivers, their vehicles and the vehicle number plate enrolled under the legislature. Utilizing the Speed Estimation module from our framework alongside the new system to be ordered The Vehicle Number Plate Detection; we can force a levy to the drivers breaking the speed limits in the city. This new system will relax the administration's concern of pedestrian safety which in turn will end the infringement of traffic laws by the drivers.

We can utilize the vehicle tracking system at the toll booth where if a vehicle does not pay the sum, at that point, the cameras will catch the picture of the number plate, register with the database and by this we can punish the vehicle proprietor by sending an online reminder. As for security the vehicle detection technique can be utilized to discover the vehicle if the burglary has happened. By getting the pictures of the vehicle, the police can get the data of the vehicle owner from the database and in this manner enabling them to discover the vehicle and catch the hoodlum.

References

[1] L. Bohang, L. Qingbing, C. Duiyong, S. Hailong, ”Pattern Recognition of Vehicle Types and Reliability Analysis of Pneumatic Tube Test Data under Mixed Traffic Condition”, Second International Asia Conference on Informatics in Control Automation and Robotics ISSN: 1948-3414, pp. 44-47, 2010.

[2] S.P. Patil, M.B. Patil, ”Moving Vehicle Detection: A Review”, International Journal of Computer Applications, vol. 87, no. 15, February 2014.

[3] Yanfeng Chen, Wu Qingxiang, ”Moving Vehicle Detection Based on Optical Flow Estimation of Edge”, International Conference on Natural Computation (ICNC), 2015.

[4] N. Chintalacheruvu, V. Muthukumar, ”Video Based Vehicle Detection and Its Application in Intelligent Transportation Systems”, Journal of Transportation Technologies, vol. 2, pp. 305-314, 2012.

[5] J. L. Baron, D. J. Fleet, and S. S. Beauchemin, ”Performance of Optical Flow Technique,” IJCV 12:1, pp. 43-77, 1994.

[6] B. D. Lucas, Kanade. ”An Interative Image Registration Technique with an Application to Stereo Vision,” DARPA Image Understanding Workshop, pp 121-130, 1981.

[7] Ehsan Nateghinia, Hadi Moradi, ”Video-Based Multiple Vehicle Tracking at Intersections”, Proceeding of the 2nd RSI/ISM International Conference on Robotics and Mechatronics, Oct 15-17, 2014.

[8] Xie Lei, Zhu Guangxi, Wang Yuqi, X.U. Haixiang, Zhenming, ”Real-time Vehicles Tracking Based on Kalman Filter in a Video based ITS”, International Conference on Communications Circuits and Systems, vol. 2, 2005.

[9] S. Siang Teoh, T. Brunl, ”A Reliability Point and Kalman Filterbased Vehicle Tracking Technique”, International Conference on Intelligent Systems, pp. 134-138, 2012.

[10] Archita Tah1 , Sudipta Roy , Prasenjit Das and Anirban Mitra1 “Moving Object Detection and Segmentation using Background Subtraction by Kalman Filter ” Indian Journal of Science and Technology, Vol 10(19), DOI: 10.17485/ijst/2017/v10i19/95427, May 2017.

[11] Mohammed Reduanul, Md. Golam Moazzam, Saiful Islam, Rony Das, Mohammed Shorif Uddin, “Vehicle Speed Determination from Video Streams using Image Processing ”, 2016 International Workshop on Computational Intelligence (IWCI) 12-13 December 2016, Dhaka, Bangladesh.

[12] Pratishtha Gupta, G N Purohit,Manisha Rathore, “Estimating Speed of Vehicle using Centroid Method in MATLAB ”,International Journal of Computer Applications (0975 – 8887) Volume 102– No.14, September 2014.

[13] Asif Khan, Imran Ansari, Dr.Mohammad Shakowat Zaman Sarker and Samjhana Rayamajhi, “Speed Estimation of Vehicle in Intelligent Traffic Surveillance System Using Video Image Processing”, International Journal of Scientific and Engineering Research, Volume 5, Issue 12, December-2014

[14] H. A. Rahim, U. U. Sheikh, R. B. Ahmad and A. S. M. Zain, “Vehicle Velocity Estimation for Traffic Surveillance System”, World Academy of Science, Engineering and Technology International Journal of Transport and Vehicle Engineering Vol:4, No:9, 2010

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my guide Prof. Sunanaya Jadhav for her exemplary guidance, monitoring and constant encouragement throughout the course of this project.
I also take this opportunity to express a deep sense of gratitude to Head of the department, Prof. Sujata Pathak for her cordial support, valuable information and guidance, which helped me in completing this task through various stages.
I am obliged to staff members of K. J. Somaiya College of Engineering, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Date: 28/5/2018

Sehab Mankada (1414029)
Vidhi Chitalia (1414074)
Nidhi Gada (1414079)