VISION BASED VEHICLE LOCALIZATION FOR INFRASTRUCTURE ENABLED AUTONOMY

A Thesis

by

DEEPIKA RAVIPATI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Peng Li
Co-Chair of Committee,	Sivakumar Rathinam
Committee Members,	Pierce Cantrell
	Rabi Mahapatra
Head of Department,	Miroslav M. Begovic

May 2019

Major Subject: Computer Engineering

ABSTRACT

Primary objective of this research is to devise techniques to localize an autonomous vehicle in an Infrastructure Enabled Autonomy (IEA) setup. IEA is a new paradigm in autonomous vehicles research that aims at distributed intelligence architecture by transferring the core functionalities of sensing and localization to infrastructure. This paradigm is also promising in designing large scalable systems that enable autonomous car platooning on highways. A reliable camera calibration technique for such an experimental setup is discussed, followed by the technique for 2D image to 3D world coordinate transformation. In this research, information is received from: (1) on-board vehicle sensors like GPS and IMU, (2) localized car position data derived from deep learning on the real-time camera feeds and (3) lane detection data from infrastructure cameras. This data is fused together utilizing an Extended Kalman Filter (EKF) to obtain reliable position estimates of the vehicle at 50 Hz. This position information is then used to control the vehicle with an objective of following a prescribed path. Extensive simulation and experimental results are also presented to corroborate the performance of the proposed approach.

DEDICATION

To my sister, my parents and my professors.

ACKNOWLEDGMENTS

I would like to thank the Department of Electrical and Computer Engineering and the Department of Mechanical Engineering of Texas A&M University for providing resources and constant support for the completion of my thesis.

I would like to express my sincere gratitude to my advisor Dr. Sivakumar Rathinam for his constant support, guidance and patience. His expertise, time, resources and consistent advice helped to complete the thesis in timely manner. I am thankful to Dr. Swaminathan Gopalswamy for his helpful insights towards IEA.

I am always thankful to Dr. Peng Li, Dr. Pierce Cantrell and Dr. Rabi Mahapatra for serving on my thesis committee.

I am extremely grateful to my CAST team Abhishek, Kenny and Tyler for their time and constant support during testing at Rellis campus.

I am fully indebted to my sister Mounika Ravipati and my parents without whom I could not have reached this milestone. I would like to thank my friends Jyothsna, Dileep, Vamsi and Amrita for being there with me all the time.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Sivakumar Rathinam of the Department of Mechanical Engineering, Professor Peng Li and Professor Pierce Cantrell of the Department of Electrical and Computer Engineering and Professor Rabi Mahapatra of the Department of Computer Science and Computer Engineering.

The experimental setup required for testing has been constructed and assembled by Kenny Chour, Abhishek Nayak and Tyler Marr of the Department of Mechanical Engineering.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

The equipment and other resources required for conducting this research were provided by Connected Autonomous Safe Transportation (CAST) lab of the Department of Mechanical Engineering.

NOMENCLATURE

IEA	Infrastructure Enabled Autonomy
ADAS	Advanced Driver Assistance Systems
AEB	Auto Emergency Braking
SA	Situation Awareness
MSSP	Smart Multi Sensor Pack
RSU	Road-Side Unit
HIL	Hardware-In-Loop
GPS	Global Positioning System
IMU	Inertial Measurement Unit
EKF	Extended Kalman Filter
ROS	Robot Operating System
MMSE	Minimum Mean Squared Error
LDW	Lane Departure Warning
BSD	Blind Spot Detection
RMSE	Root Mean Squared Error

LIST OF FIGURES

FIGURE Page

TABLE OF CONTENTS

Page

1. INTRODUCTION

Records show that annually 1.25 million people lose their lives in road accidents and nearly 50 million people are injured [1]. Most of the deaths are caused due to preventable driver errors. Autonomous or highly intelligent and aware vehicles have the potential to minimize this number considerably. Also, such vehicles have promising benefits in terms of safe roads, fuel and traffic efficiency [2]. Therefore, autonomous vehicles received significant attention in recent years. However, building an autonomous vehicle with perception and decision making capabilities exceeding human abilities is not easy. According to National Fatality Analysis Reporting Systems (FARS), a human driver commits a fatal error only once in roughly 88 million miles [3]. Hence, developing a safe autonomous vehicle with such precision has always been the greatest challenge.

1.1 History of Autonomous Vehicles

Roboticists have always contemplated the possibility of building mobile robots that are as capable and intelligent as humans. Autonomous mobile robots find applications in various areas like space explorations, search and rescue operations, domestic robotics, medical surgeries, etc. Self-driving or autonomous vehicles are no different than mobile robots.

Research on autonomous vehicles started in the 1500s even before motorized automobiles spawned the market. Leonardo da Vinci was the earliest to design a self-propelled cart powered by coiled springs. Since then, the field of intelligent and autonomous vehicles took many curves resulting in different precursors on its way. The level of autonomy has also progressed in the same direction. Starting with no autonomy or level 0, research advanced through shared autonomy with driver assistance, conditional autonomy and then to full or level 5 autonomy. Ultimately, a fully autonomous and self-sufficient vehicle was developed at Carnegie Mellon University's Navlab in the 1980s. This trend has continued [4] [5].

DARPA's Grand Challenge [6] [7] in 2004 sparked further research by providing a unique and modern platform for testing state-of-the-art techniques and further innovation in this field. The

task for the challenge was to design a driverless car that could travel 150 miles in the Mojave desert. Although none of the candidates could reach the goal, the event was considered successful as it spurred interest and opened avenues for further innovations. This led to the huge success of 2005 Grand Challenge, which mainly focused on endurance of vehicles in deserts and avoiding stationary obstacles.

Since the DARPA's Grand Challenge 2005, many key players in the industry like Mercedes Benz, Bosch, Nissan, Toyota and Google, have accelerated their research and released their prototypes of self-driving cars. Presently, these prototypes are being tested on actual roads and have proven that the autonomous vehicles can give desirable performance in challenging environments. The failure of physical components like an engine or a transmission in the vehicle have become very rare. As a result, the focus of automakers shifted to enhance safety and help humans make better driving decisions. As a result, many automakers have started to equip most of their luxury cars with Advanced Driving Assistance Systems (ADAS) like Auto Emergency Braking (AEB), parking assist, blind spot detection among others.

1.2 General Architecture of an Autonomous Vehicle

An autonomous vehicle should be able to achieve the following:

1. sense its surroundings using the available sensors, termed as Direct Perception [8]
2. localize itself with respect to the environment
3. make decisions according to received perceptions and situation awareness
4. control actuators or Drive-by-Wire (DBW)

This can be visualized in the Figure 1.1. This reduces to the classical Sense, Plan, Act robotic paradigm. In addition to the above, an autonomous system is expected to make sensitive decisions in critical and hypothetical situations as a human does, in order to minimize risk.

Sensing and Perception are pivotal in determining the contextual and situation awareness information. Sensors are crucial as they interact with the world on behalf of the vehicle and measure

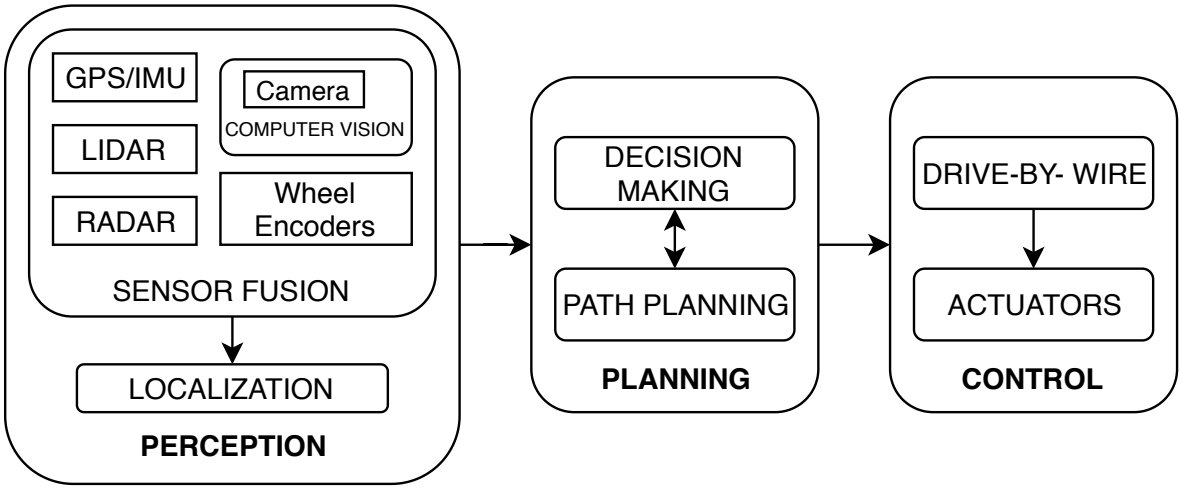

Figure 1.1: General architecture of an autonomous vehicle

some attribute of the world. Figure 1.2 depicts a typical autonomous vehicle prototype coupled with an expensive array of sensors - Camera, Global Positioning System (GPS), Inertial Measurement Unit (IMU), Laser Detection and Ranging (LiDAR), Wheel Encoders, Radio Detection and Ranging (RADAR) and more. Thanks to the recent advancements in hardware technologies, all the sensors today are available with great range and resolution.

The availability of high resolution sensors resulted in high quality images and point clouds, whose size grow in the order of gigabytes per minute. Hence, there was a significant need for computation requirements. With the advent of Graphics Processing Units (GPU) and multi-core CPUs, high computation power became readily available. This ensured the near perfect pedestrian and vehicle recognition an achievable feat. Also, huge point clouds and high resolution video streaming can be processed in seconds using state-of-the-art computer vision and deep learning techniques to extract maximum information obtained through sensing. This information is used for localization and mapping to understand the surroundings. This gives the vehicle, a reliable dependable decision making capability.

Another direction in the autonomous research is learning based driving [9]. Self-supervised learning based terrain roughness estimation and speed selection [10] [11] have shown good progress. Deep learning algorithms to predict the steering wheel angle based on steering wheel data is an-

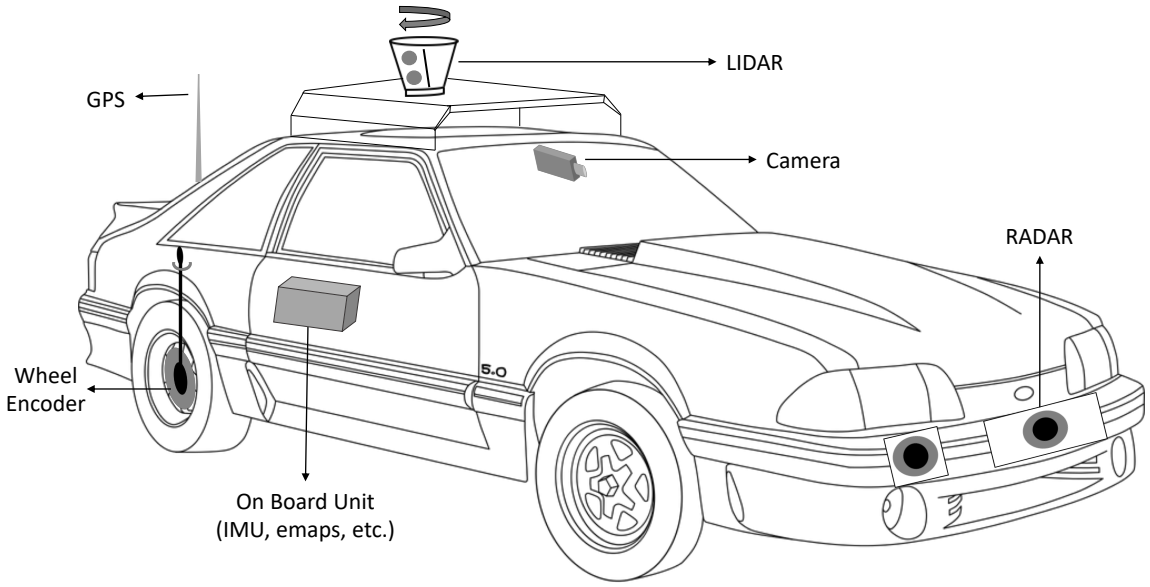

Figure 1.2: Vehicle installed with all sensors that solve localization

other interesting research. Learning by demonstration has also fetched good results in this direction.

Despite great efforts and research, the penetration of autonomous vehicles into roads has been very slow. In the 5 levels of autonomy starting with level 0, no commercial vehicle has successfully crossed level 3 autonomy. Level 5 or fully autonomous vehicles are still under research and testing. This is mainly because the current prototypes demand automobile manufacturers to bear the responsibility and liability in providing perception and situation awareness capabilities to the vehicle. A study on the driving functionality gives useful insights in realizing an autonomous vehicle design. As described in [12], driving functionality can be decomposed into three parts:

1. Situation Awareness (SA) synthesis using one or more sensors to develop a contextual and self-awareness for the vehicle and driver,

2. Driving Decision Making (DDM), which defines the desired motion based on the self-awareness information

3. Drive-by-Wire (DBW), which generates control signals for the actual motion of the vehicle.

The penetration of autonomous vehicles could be achieved faster if these responsibilities can be redistributed among automobile manufacturers, third party ADAS systems in the car and infrastructure. This report sheds light on the new paradigm of autonomous vehicles, namely, Infrastructure Enabled Autonomy (IEA), detailed in the next chapter.

1.3 Organization

Chapter 2 focuses on IEA architecture, its advantages and challenges. This chapter also presents a simulation study made on IEA to evaluate its feasibility of scaling and distribution.

Chapter 3 discusses about the estimation of the vehicle world coordinates from 2D image coordinates. Since the cameras are mounted at a height of 25-30 meters, conventional techniques of camera calibration might not work with IEA setup. Hence, a calibration technique has been devised and evaluated.

Chapter 4 discusses lane detection on IEA set up and localization based on lane detection. Cameras mounted on RSUs are subject to sway as they are mounted at heights. This sway usually corrupts the calibrated camera parameters. The offset by which the vehicle is away from the center of the lane could be used as a control parameter, instead of entirely relying on the camera transformed coordinates. Hence, an image processing pipeline using Hough Transform will be discussed that has been used for IEA lane detection.

Chapter 5 details on Kalman filter based sensor fusion, to fuse the measurements received from the lane detection, camera data and vehicle on-board sensor data (GPS/IMU), to give a fine localization of the vehicle.

2. INFRASTRUCTURE ENABLED AUTONOMY

It has been discussed in the earlier chapter that the responsibility of providing all the three driving functionalities to the vehicle, namely, Situation Awareness (SA), Driving Decision Making (DDM) and Drive-by-Wire (DBW) is the main hindrance of the autonomous vehicles' production and their penetration in everyday society. Infrastructure Enabled Autonomy (IEA) is a new template of transportation that can provide a solution to this problem by transferring the primary responsibility of localization from the vehicle to infrastructure [13]. In other words, sensors embedded within the infrastructure can sense the surroundings and provide sufficient localization and SA information to drive the vehicle from start to destination.

Whether it is a human driven or an autonomous vehicle, getting updated with the current location information can significantly aid the decision making. Often in metropolitan cities, due to high buildings, GPS signals become unreliable. Such GPS denied environments might leave the driver with no neighborhood information. Offloading such responsibility to infrastructure could truly improve the driving experience and aid to Driving Decision Making (DDM).

2.1 IEA Architecture

IEA architecture can be realized by devising and constructing special traffic corridors with Road-Side Units (RSU) placed on either side of the road. These RSUs carry the sensors required to localize the vehicles going on the road and monitor the traffic. This sensor package is called Smart Multi-Sensor Package (MSSP). Along with MSSP, there are specialized SmartConnect devices, which aim to establish wireless connectivity between the vehicles and MSSPs. These enable the communication between the vehicle and infrastructure, thereby, transmitting information necessary for its localization. The vehicle is also installed with a SmartConnect device, making it possible to receive information from MSSPs. Driving Decision Making (DDM) is implemented locally in the vehicle leveraging the SA information received, followed by DBW. Figure 2.1 depicts a typical IEA enabled traffic corridor.

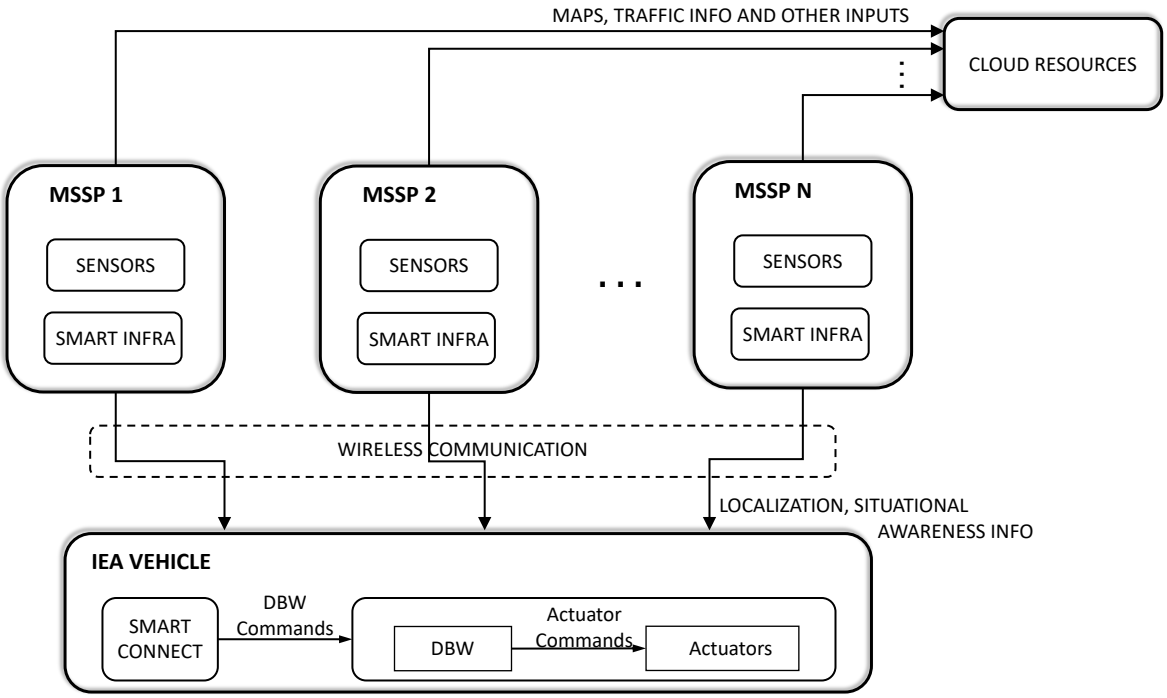

Figure 2.1: IEA architecture visualizing on a special traffic corridor

As a result, this architecture leaves the automakers with the only responsibility of Drive by Wire, which they can produce with high reliability. This way the liabilities can be re-distributed and this in turn accelerates the production and deployment of self-driving cars on roads in the near future. Evidently, there are many research challenges in the realization of IEA architecture with respect to scale, distribution, cost and complexity that need to be addressed. Hence, it is ideal to evaluate the architecture prior to the real time realization and testing.

2.2 IEA Simulation Setup and Study

Having acknowledged the concept of Infrastructure Enabled Autonomy and its challenges, a simulation environment imitating IEA has been developed. Developing a simulation environment which incorporates the discussed features is challenging because of its complexity and scale. As the number of RSUs increase, computational requirements of simulation increases. Hence this simulation architecture is designed as a hybrid and distributed Hardware-in-the-Loop (HIL) model as explained in [12] and is discussed briefly below. Although several hardware-In-the-Loop sim-

ulation setups exist in the literature, this architecture is unique of its kind to accommodate key features of IEA. The simulation environment has been set up with the help of Robot Operating System (ROS) and Gazebo.

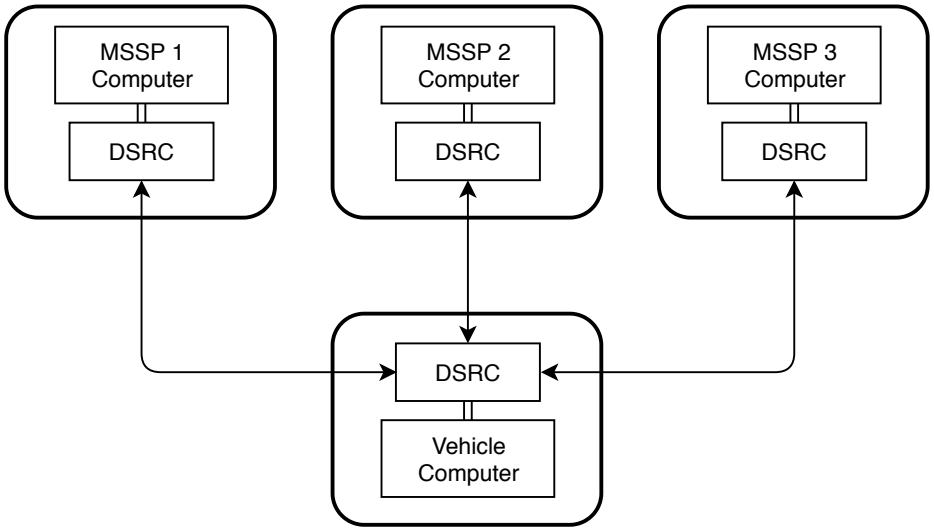

Figure 2.2: Hybrid simulation architecture for IEA

The architecture of the simulation is shown in figure 2.2. The simulation set up consists of multiple computers, one named as the vehicle computer which has the standard vehicle dynamics set up and the remaining are called MSSP computers, which have the MSSPs setup. In other words, MSSP computers simulate the camera streaming and act like MSSPs containing cameras, mounted on Road-side units. Alongside, a Dedicated Short Range Communication (DSRC) device is coupled with each computer to establish communication between the vehicle and MSSP computers. DSRC is a wireless communication technology used for vehicle to vehicle communication. It emerged as a result of the dedicated spectrum of 75 MHz in 5.9 GHz frequency band allocated for Intelligent Transportation Systems. The DSRC device takes the role of special SmartConnect device discussed earlier in the IEA architecture.

In this simulation setup, DSRC devices are used to serve two purposes. One is to send the vehicle localization information from MSSP to the vehicle, which is the same role as in actual

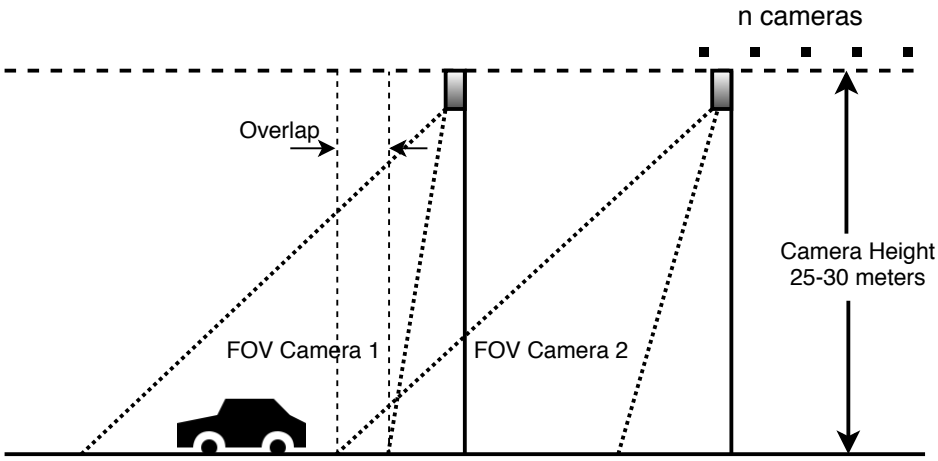

Figure 2.3: IEA corridor visualization

realization of the system. The other is to send the actual location of the vehicle between the MSSP computers to simulate the exact position of the vehicle in their own simulation environments. This is essential as DSRCs are the only way all the computers are connected. This enables the vehicle to pass by each MSSP computer and the vision algorithms localizes the car in the image frame and transforms the image coordinates into actual world coordinates. In this simulation set up, camera streaming and vehicle dynamics are simulated. DSRC communication, vision processing and camera transformations run in real-time. This is the only difference between the actual realization and the simulation study.

The goal of the simulation is to make the car travel from a start point to a desired end point, passing through an IEA traffic corridor consisting of 3 MSSPs. A series of image processing operations take place to get the position estimate of the car from images and then to transform them to real world coordinates. The simulation environment is simple without any occlusions and hence, background subtraction algorithm is used to detect any presence of vehicle in the camera streaming. Background subtraction algorithm identifies any notable change in the intensities of the current image with respect to the previous image. This algorithm detects any new object in the frame or motion of an already existing one. When the vehicle enters the field of view of the camera,

background subtraction extracts its position coordinates. These coordinates are used to trigger Tracking-Learning-Detection (TLD) algorithm [14], that keeps the track of the vehicle. TLD is the most widely used tracking algorithm. Its unique tracking, detection and learning pipeline makes it one of the fastest tracking algorithms, that can output the coordinates of the vehicle in an image with high frequency.

Once the vehicle is localized in the 2D image frame, the coordinates are used to estimate the vehicle position in the actual world. All the required information about the cameras are known before hand, which includes the camera position, focal length, orientation with respect to the world and the height at which the camera has been placed. Hence, camera matrix can be modeled to obtain intrinsic and extrinsic matrices and perform a perspective transformation to estimate 3D world car coordinates from 2D. The details‘ about the camera modeling and projections are elaborated in chapter 3. For now, the equations are presented to show the relationship between the 2D and 3D coordinates as discussed in [15]

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r11 & r12 & r13 & t1 \\ r21 & r22 & r23 & t2 \\ r31 & r32 & r33 & t3 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.1}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} R|T \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.2}$$

where K represents the intrinsic matrix or the geometric properties of camera; it is a $3\text{x}4$ matrix, R represents the Rotational matrix or the orientation of camera with respect to X, Y, Z axes of the

world; it is a 3x3 matrix and T represents the location coordinates of the camera; it is a 3x1 vector. Using the back projection [15] to a known height, world location of the car is obtained and is used by the controller for navigation towards the destination.

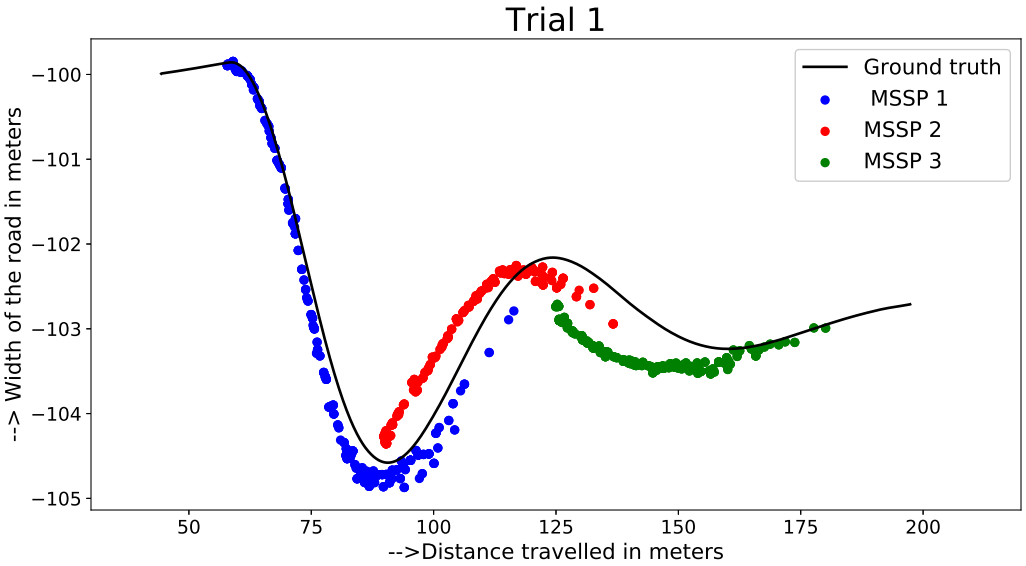

Figure 2.4: True position and localized vehicle position by MSSP camera plotted at vehicle velocity 6m/s

Figure 2.4 shows the simulation results of the 2D to 3D world mapping, which is processed using equation 2.1. The projection made a localization close to the true value. The results also show that the vehicle location has been transmitted between MSSPs and vehicle, and ultimately the vehicle reached the goal which is towards the end of third MSSP. One more key takeaway from the simulation results is about the distance between two MSSPs. It can be observed from figure 2.5 that as the vehicle moves far from the MSSP, error between the estimated value and the true value increases. It has been found from multiple trials that the ideal distance between MSSPs for this setup would be 90-100 meters, with an overlap. The results obtained from the simulation were promising and they advocate the actual realization of an IEA corridor.

The realization of IEA corridor is also a similar setup. In the experiment field, the cameras

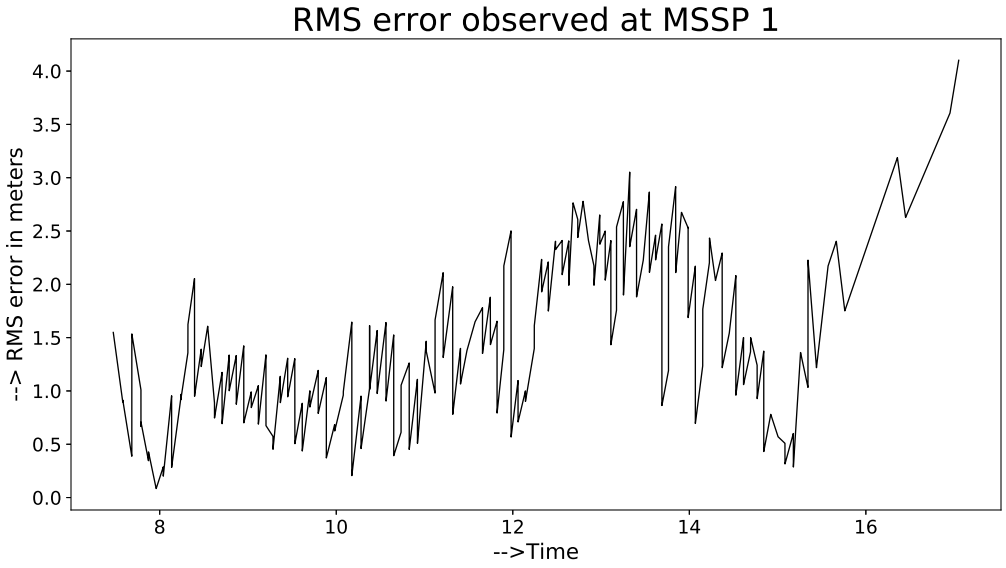

Figure 2.5: RMS error of the position estimates observed on MSSP 1

are mounted on RSUs at a height of 25-30 meters, with a field of view of nearly 80 meters. Each RSU is also set up with a DSRC sensor to enable the wireless communication. The vehicle is set up with a computer, DSRC sensor, low resolution GPS/IMU sensors and one high resolution GPS sensor which serves as ground truth to compare against the vehicle's position derived from algorithms described in this research. Except for these, there are no additional sensors installed in the vehicle that aid localization. The vehicle relies on the localization data received from the cameras installed in the infrastructure, which is basically 2D image frame information. A series of images to world coordinate transformations are applied to extract the world coordinates and rest of the setup is same as the simulation.

3. 3D POSITION ESTIMATION AND CAMERA MODEL

3.1 Homogeneous Coordinate System

As explained in earlier chapters, one of the key challenges pertaining to IEA architecture is to localize the vehicle in 3D world coordinates, given the 2D co-ordinates with respect to image frame. This is particularly interesting as it reduces to the standard problem of pose estimation. Solving vision problems involving pose estimation requires prior knowledge on camera parameters and camera model used. As we deal with three different frames of reference nearly camera frame, image frame and world frame, there is a need to define every transformation in a mathematical way. Consider a 3D translation and we are dealing with Euclidean Coordinate system. Let $P\,[X_1, Y_1, Z_1]$ be any point. Now, if the origin is translated to $[t_x, t_y, t_z]$ then the translated coordinates of P can be written as

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{3.1}$$

This translation can be alternately written as a matrix multiplication form thus allowing Linear Transformation as follows:

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} \tag{3.2}$$

The above representation of a 3D coordinate to a 4D vector $[X, Y, Z, 1]$ lets the transforma-

tion to be represented as a matrix. This method is called homogeneous coordinates representation. Representing image coordinates and world coordinates in homogeneous coordinate system allows common geometric operations like scaling, rotation, translation and perspective projection to be implemented as matrix operations. In other words, any operation we apply to camera like translation or rotation, we can associate a matrix with it. Similarly scaling can be represented as

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} X_1 \times S_x \\ Y_1 \times S_y \\ Z_1 \times S_z \end{bmatrix} \implies \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} \tag{3.3}$$

A similar concept of homogeneous coordinates and matrix operations is used to establish a relationship between the image coordinates and the world coordinates. Projection of world coordinates into image coordinates require camera modeling and calibration. This is because the angle at which the camera is located (rotation matrix), the position of camera (translation vector) and its focal lengths (camera properties) highly determine how the world frame gets captured in the image frame.

Camera parameters can be represented in two matrices called Intrinsic and Extrinsic camera parameters. Camera calibration is the method employed to model all these parameters. The procedure on how calibration is done for the experiments is detailed later in this chapter. As mentioned, any transformation to matrix has to be compensated by a matrix multiplication. Therefore, the relation between camera coordinates (not image coordinates) and world coordinates can be given as

$$P_c = TRP_w = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r11 & r12 & r13 & 0 \\ r21 & r22 & r23 & 0 \\ r31 & r32 & r33 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.4}$$

where, P_c represents the camera coordinates and P_w represents the world coordinates. Image and camera coordinates are linked by one more matrix which considers the scaling due to focal lengths and skew of the axes. Hence a relation can be drawn between image coordinates and world coordinates as

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r11 & r12 & r13 & t1 \\ r21 & r22 & r23 & t2 \\ r31 & r32 & r33 & t3 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.5}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} R\,T \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.6}$$

where $[x, y, 1]$ is the homogeneous image coordinate (location of a pixel) and $[X, Y, Z, 1]$ is the corresponding world coordinate. K is the intrinsic camera matrix and it represents the internal properties of the camera. R is the rotational matrix and T is the translation vector or the location coordinate of the camera.

Intrinsic Matrix describes the geometric properties of a camera, like the focal length and skew

along the axes. It conveys how the pixel coordinates of the image are linked with corresponding camera reference frame coordinates. Extrinsic matrix describes the position of the camera with respect to world and its orientation (direction it is pointing to). Together, the resulting matrix is called Camera Matrix.

3.2 Camera Calibration

Camera calibration is a technique of finding intrinsic and extrinsic parameters of the camera that is used for imaging processes. This helps in modeling the correspondence between image sensor measurements and the 3D world. Meticulous calibration is pivotal for many applications like 3D interpretation of images, 3D reconstruction and robot manipulator design.

3.2.1 Traditional Calibration Method

Checkerboards are often used to perform camera calibration and compute the parameters required. Their repeating grid structures provide natural interest points; they are local features of any image which are invariant to many transformations. The corners of each square in the checkerboard act as an interest point. The main idea behind checkerboard calibration is that the world coordinate system is set to one corner of the checkerboard. This means all the points on the board now lie on the same plane, making the Z coordinate zero. Hence, the extrinsic matrix now consists of 4 columns with the third column zeros.

The dimensions of the checkerboard are known prior. This makes it possible to collect a handful of image coordinates through interest point detection and corresponding world coordinates through the real dimensions of each square. The rest follows from the standard linear algebraic operations and matrix partitions applied on equation 3.6. This way intrinsic and extrinsic matrices are obtained. Direct Linear Transformation (DLT) [16] and Multi-plane Calibration [17] are two classical methods that employ checkerboards for calibrating cameras.

3.2.2 Camera Calibration for IEA

In the context of IEA, cameras are mounted on road-side units at a known height, usually between 25 to 35 meters. Hence, the calibration is done for the mounted cameras. This follows

the same idea explained above. On the test field, few world coordinates are captured using GPS and corresponding image coordinates are marked on the image. It must be noted that all the world points would be coplanar. For example, if m points are collected on the ground, corresponding m image coordinates are collected on the image. The Z coordinates remain the same for all the world points collected. Now, consider the equation 3.6

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} R\,T \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.7}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} R'\,T' \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.8}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = R' \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T' \tag{3.9}$$

$$\begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \\ 0 \end{bmatrix} = R' \times \begin{bmatrix} X_1 - X_2 \\ Y_1 - Y_2 \\ 0 \end{bmatrix} \tag{3.10}$$

$m - 1$ such difference vectors exist. Taking an inverse of world coordinates matrix, leaves R', a 2x2 matrix.

3.2.3 Transformation of Image Coordinates to World Coordinates

Now that the cameras are calibrated, when the vehicle enters the field of view of any camera, it is localized in the image using a deep learning algorithm called Yolo (You Only Look Once) [18] and the corresponding x, y image coordinates are obtained. These image coordinates are being transformed to world coordinates to localize the vehicle with respect to world. Consider equation 3.10. The (x, y) image coordinate is subtracted from the nearest known point among all those considered for calibration. The corresponding world coordinate is also subtracted on the left hand side. Rest of the steps involve simple linear algebraic operations.

Let $[x_l, y_l]$ be the localized coordinates in the image corresponding to the vehicle. Let $[X_k, Y_k]$ be the known world coordinates corresponding to $[x_k, y_k]$ image coordinates. Now, from equation 3.10,

$$\begin{bmatrix} x_l - x_k \\ y_l - y_k \\ 0 \end{bmatrix} = R' \times \begin{bmatrix} X - X_k \\ Y - Y_k \\ 0 \end{bmatrix} \tag{3.11}$$

$$\implies R'^{-1} \begin{bmatrix} x_l - x_k \\ y_l - y_k \end{bmatrix} + \begin{bmatrix} X_k \\ Y_k \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} \tag{3.12}$$

This way the transformation of image to world coordinates is done, thus localizing the vehicle in real world.

3.3 Results

The experiment field has been calibrated with the method described in the above section and equation 3.10. When the vehicle enters the field of view of the camera, the localized 2D image coordinates are transformed to real world coordinates as described in equation 3.12. The following figures 3.1 and 3.2 show the results of the transformed coordinates in the actual field.

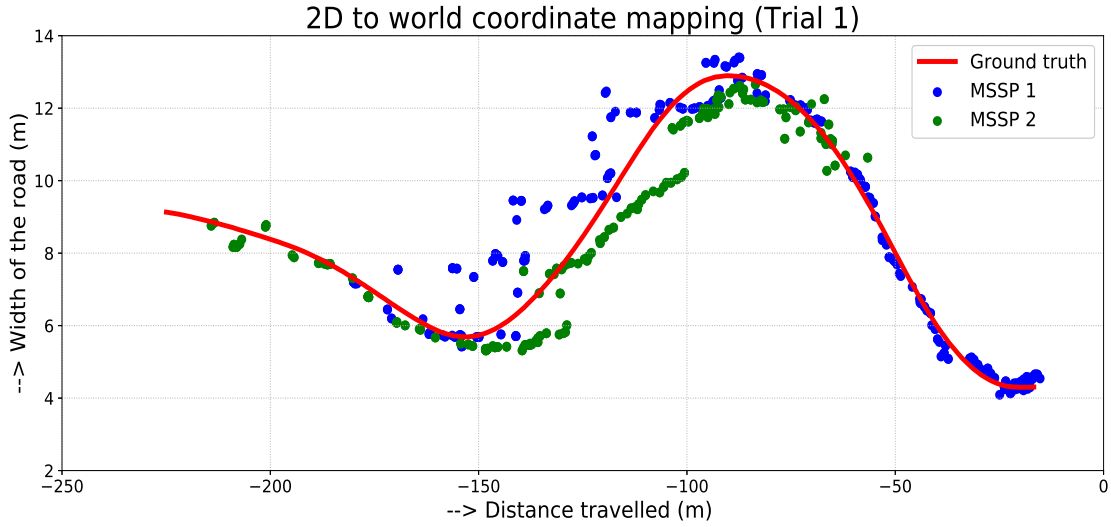

Figure 3.1: Results for the image coordinates to the real world coordinates transformation, trial 1

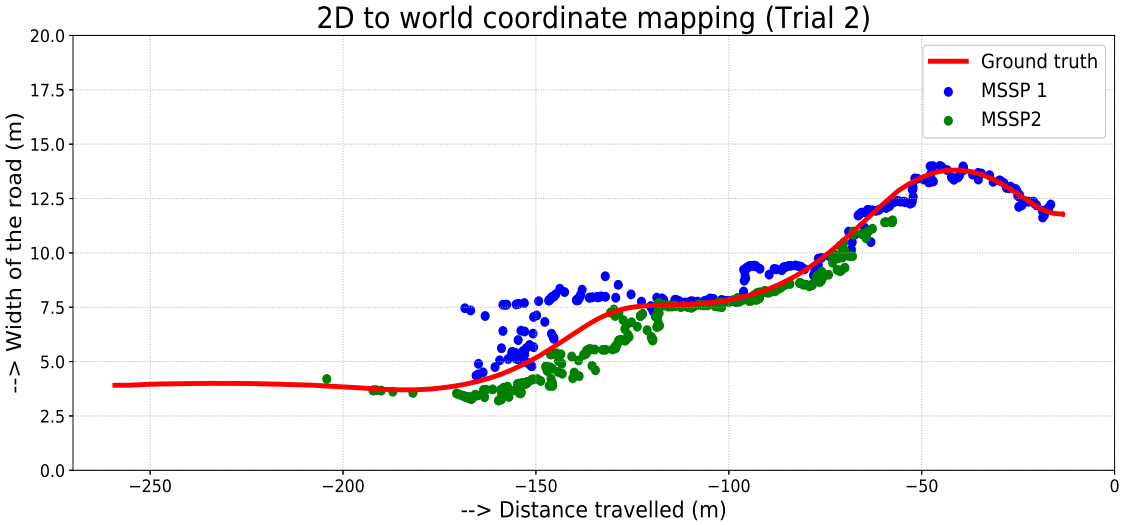

Figure 3.2: Results for the image coordinates to the real world coordinates transformation, trial 2

4. THE LANE DETECTION

4.1 Lane Driving and Importance of Lanes

Many countries including the United States have lanes marked on highways and corresponding rules established on lane keeping and changing. Lane based driving makes the commute time and traffic efficient. Lane keeping and changing can sometimes cost lives due to missed attention. These techniques also characterize a human driving skill which can be offloaded to machines. As a result, Lane Departure Warning (LDW) and Blind Spot Detection (BSD) systems have emerged in recent years. Many automakers like Nissan Motors, Toyota and Mercedes made these safety systems available in their vehicle models in the early 2000s.

There are three types of lane departure warning systems currently available:

1. systems that only warn the driver if the vehicle is moving away from the lane
2. systems that warn and if no action is taken, take necessary steps to ensure the vehicle stays within the lane
3. systems that take over steering and keep the vehicle centered in the lane

Lane keeping techniques mentioned above can be implemented for IEA but with a goal of achieving a precise localization and control. As the cameras are mounted at a good height of 22 - 30 meters, they are subjected to sway. This sway corrupts the calibration of cameras making the world coordinate transformations inaccurate. The offset by which the vehicle is away from the center of the lane can be used as a feedback to control the car orientation and keep the vehicle centered on the lane.

Lane detection can be considered a classical image segmentation problem of identifying lane areas and non-lane areas. Many traditional methods as well as deep learning techniques are available in literature [19] [20]. The work in [21] by Xiao et. al talks about the lane detection using Conditional Random Fields (CRF) applied on LiDAR data. The development of high definition

LiDARs led to many LiDAR based road detection techniques. As many research organizations have supported further innovation by providing the datasets like KITTI [22], thousands of labeled training images became available to feed the data hungry deep neural networks. Given the advancements of hardware and computing ability, deep learning algorithms on lane detection have set new benchmarks beating all the state-of-the-art techniques.

Considering IEA set up explained in Chapter 2, a typical camera feed view looks like figure 4.2.

Computation power is one of the key challenges of IEA. As deep learning techniques require intensive computing power, considering the facts that the cameras are stationary and offer fixed field of view, simple state-of-the-art techniques are employed to detect the lane markings on IEA corridor. An image processing pipeline shown in Figure 4.1 has been used for the lane detection in IEA. Each frame of camera streaming undergoes this pipeline.

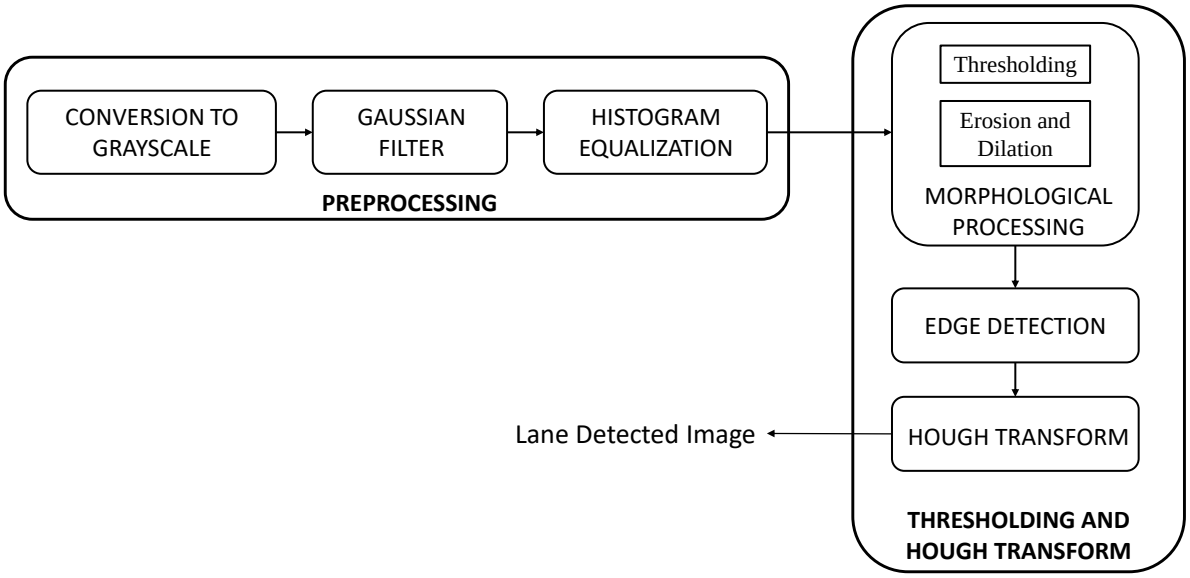

Figure 4.1: Lane detection pipeline

4.2 Lane Detection Pipeline

4.2.1 Image Preprocessing

The preprocessing step is done to enhance few features before proceeding to the actual detection. Image's color depth is reduced to grayscale, with pixel intensities ranging from 0 to 255, as it makes the processing easier and computationally efficient. The processed image is then passed through a Gaussian filter. This removes any unwanted and high frequency noise components of the image. As this is a low pass filter, the resulting image would be blurred. Now the image frame is ready for further processing.

4.2.2 Adaptive Histogram Equalization

Since the image intensities highly depend on the lighting conditions, some areas of the image may be brighter than other areas. A close view on the histogram of the image shows how the pixel intensities are distributed throughout the image. A good image can be described with the fact that the pixel intensities are equally distributed over all the regions of histogram in the range [0, 255]. Histogram equalization method is used to improve the overall contrast of the image. It is the normalization of intensities of the image and thereby flattening the histogram. Instead of considering global contrast, the image is divided into blocks, say 8 x 8, and each block is equalized independently. This is called Adaptive Histogram Equalization [23] and this step renders proper contrast to the image for further processing. Figure 4.3 shows the enhanced image using adaptive histogram equalization.

Following histogram equalization, thresholding operation is performed, converting the grayscale image to a binary image. The typical threshold value would be 180 and it is accurate enough to capture the lane markings. Morphological operations namely erosion and dilation are performed on the image repeatedly to remove any tiny noise pixels that arise due to hard thresholding. The resulting image is then given as an input to canny edge detector, which outputs the edges of the image, highlighting the lane markings on the binary image. Now this image is fed to a line fitting algorithm to get the mathematical model for the lanes.

Figure 4.2: IEA camera view

Figure 4.3: Histogram equalized image

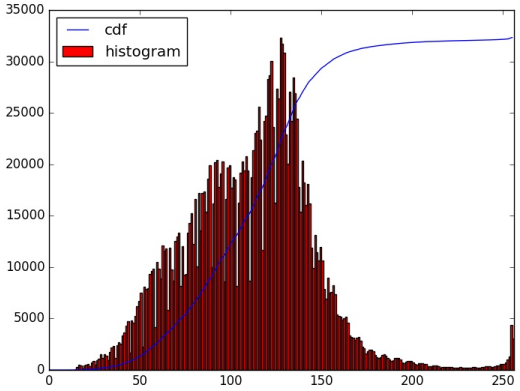

Figure 4.4: Histogram of the original image

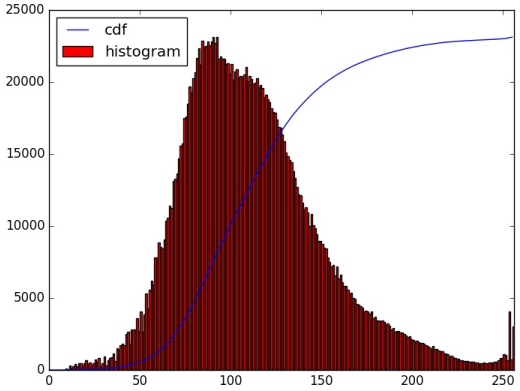

Figure 4.5: Histogram of the enhanced image

Lane detection boils down to a curve or line fitting problem after the preprocessing and thresholding step. The image resulting from the above process is a binary image with lane marked pixels highlighted.

Line fitting can be done using three methods:

1. Least square fit, which aims at reducing the mean squared error between all the points and the fitted line

2. Random Sampling and Consensus (RANSAC), which iteratively chooses two points at random to draw line and it quits when the error between the current solution and all other points is less than a threshold value

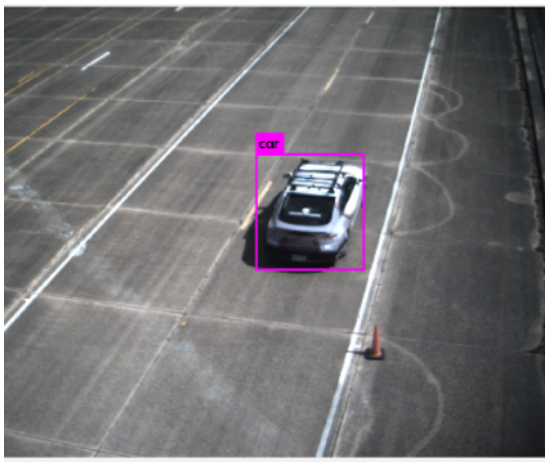

Figure 4.6: IEA camera view

Figure 4.7: Grayscale image

Figure 4.8: Adaptive Histogram Equalization

Figure 4.9: Thresholded image

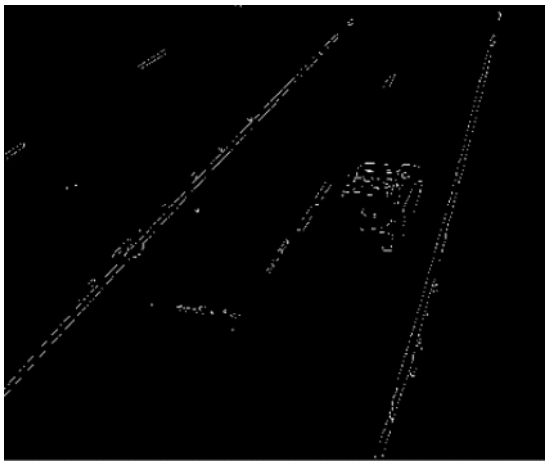

Figure 4.10: Edge detection

Figure 4.11: Lanes detected

3. Hough Transform is another line fitting algorithm and in an image consisting multiple lines, it decides which points belong to which line based on polling. It is a feature extraction method, simple to use that can segregate points belonging to their respective lines

In the entire test set up of IEA, lanes were straight and hence, Hough transform has been used to detect lanes. This method can be extended for curved roads by replacing the Hough transform step with the least square fit of a second degree polynomial and following same pipeline.

4.2.3 Brief Introduction to Hough Transform

Fundamental principle driving Hough transform is the voting procedure in the parameter space. This can be elaborated as follows.

The standard form of straight line with slope m and intercept c can be formulated in Euclidean co-ordinate system as

$$y = mx + c \tag{4.1}$$

Re-arranging the equation,

$$c = -mx + y \tag{4.2}$$

$$c = -m(x_i) + y_i \tag{4.3}$$

It can be observed that equation 4.3 is the equation of line in (m, c) coordinate space. This is the principle of Hough transform. Hough transform takes each edge point in the image (after edge detection), transforms it into the format shown in equation 4.3 and accumulates all such points belonging to a line. If there are multiple lines, the accumulator collects those points separately. This process is called polling. If the number of such points is greater than the mentioned threshold, the algorithm finds local maxima in that (c, m) space. If the polling count does not cross the threshold required to call it a line, then the accumulator is discarded.

The equations of the lanes drawn on the image shown in figure 4.11 are now known. The (x, y) location of the vehicle is obtained from the deep learning algorithm. The distance between the

vehicle and the center of the line is calculated as the perpendicular distance from a point to the line. This serves as a control parameter to keep the vehicle at the center of the lane. This serves as one of the measurements to the sensor fusion discussed in the next chapter.

5. SENSOR FUSION AND KALMAN FILTER

5.1 Introduction

Sensor fusion is the process of combining sensory data obtained from disparate sensors such that the uncertainty in the resulting information would be minimal and can enhance the system performance. Most of the autonomous vehicles today have an expensive array of sensors installed as discussed in the Chapter 1 and they employ multi sensor fusion techniques to obtain reliable localization and situation awareness information.

There are many techniques available in literature about the multi sensor fusion like alpha filter, complementary filter and Kalman filter. Alpha filter is a first order filter that simply updates from measurements and predicts the states on time propagation.

$$\hat{x}_k = (1 - \alpha)\bar{x}_k + \alpha\tilde{x}_k \tag{5.1}$$

where $\hat{x}_k$ is the updated state from measurement $\tilde{x}_k$ and $\bar{x}_k$ is the predicted state. This approach filters out noise but is not very rigorous or optimal. Kalman filter [24] is the most widely used sensor fusion technique because of its robust model and reliability. It is a recursive estimator and is the optimal Minimum Mean Square Estimate (MMSE) filter, given the details of the correctness of measurements and system. Similarly, complementary filters are a simpler form of Kalman filters and are employed for sensor fusion. However, complementary filter also is not an optimal solution for a modeled random process.

In the IEA experimental set up explained earlier, the vehicle is installed with low resolution Global Positioning Systems (GPS) and Inertial Measurement Unit (IMU) sensors. Along with these measurements, location details are also estimated from cameras in the MSSPs embedded in the infrastructure as discussed in Chapter 3. Hence, the vehicle receives information from multiple sensors and it needs to be synchronized and fused to reduce the uncertainty of the position. The fusion technique employed here is Kalman filter.

5.2 Kalman Filter

Kalman filter, also known as Linear Quadratic Estimation (LQE) gives one of the best fits to observed measurements, given a modeled random process. It estimates the states following a two step procedure, prediction and measurement. If it is known prior about the noise model of the measurements and noise model of the system through their covariance matrices, Kalman filter can give an optimal statistical estimate of the data. In general, Kalman filter is expected to operate on linear systems represented as below

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{5.2}$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{5.3}$$

where

$\mathbf{x}$ is the State Vector

$\mathbf{y}$ is the Output Vector

$\mathbf{u}$ is the Input Vector

$\mathbf{w}$ is the Process Noise Vector

$\mathbf{v}$ is the Measurement Noise Vector

$\mathbf{F}$ is the State System Matrix

$\mathbf{G}$ is the Input System Matrix

$\mathbf{H}$ is the Observation Matrix

Since Kalman filter is recursive, the process starts with the prediction of state and error covariance matrix and then updates with the measurements received. The a priori are calculated with the prediction step as follows

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{Q}_k\mathbf{u}_k \tag{5.4}$$

$$\hat{\mathbf{P}}_{k|k-1} = \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k^T + \mathbf{Q}_k\mathbf{u}_k \tag{5.5}$$

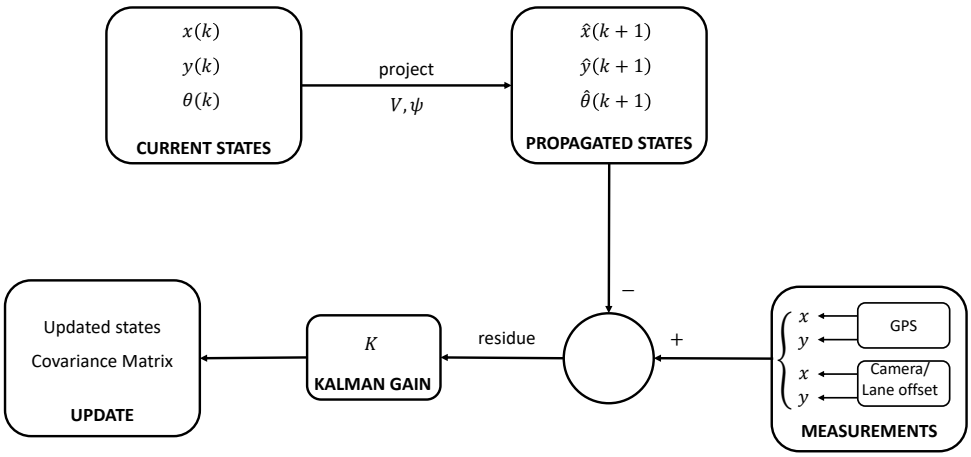

Figure 5.1: Kalman filter block diagram

where $\hat{\mathbf{x}}_{k|k-1}$ is the state estimate and $\hat{\mathbf{P}}_{k|k-1}$ is the error covariance matrix estimate.

When the system receives the measurements, it calculates the Kalman gain and thus corrects the system by calculating posteriori estimates of state and error covariance matrix.

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{5.6}$$

The new corrected state matrix becomes

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}) \tag{5.7}$$

Similarly, the error covariance matrix of the state is corrected as

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \tag{5.8}$$

$\mathbf{R}$ and $\mathbf{Q}$ matrix are called Measurement Noise and Process Noise Covariance Matrices respectively. These are the matrices that convey the uncertainty in measurements and process. $\mathbf{R}$, $\mathbf{Q}$

and $\mathbf{P}$ are chosen through trial and error methods when the randomness of the model is not known prior.

5.3 Vehicle Dynamics and Extended Kalman Filter

Vehicle in IEA is modeled as a Constant Heading and Velocity model (CHCV). The model is presented with the equations below.

$$\dot{x} = Vcos\theta \tag{5.9}$$

$$\dot{y} = Vsin\theta \tag{5.10}$$

$$\dot{\theta} = \psi \tag{5.11}$$

where x, y are position coordinates, θ is the heading angle, ψ is the yaw rate of the vehicle and V is the velocity of the vehicle. Though this is a non-linear system, it is differentiable. The states to be estimated are x, y, θ, while ψ and V are the inputs to the system. The system can be represented as a discrete time system; the state estimates after time Δt can be represented as below.

$$x_{k+1} = x_k + \Delta t V cos\theta_k \tag{5.12}$$

$$y_{k+1} = y_k + \Delta t V sin\theta_k \tag{5.13}$$

$$\theta = \theta_k + \Delta t \psi_k \tag{5.14}$$

Since the system is non-linear and differentiable, an Extended Kalman Filter (EKF) is used to model this non-linear system. This is same as the Kalman Filter except that, Jacobian is calculated at every point to linearize the system. The system can be represented as

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \tag{5.15}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{5.16}$$

For our vehicle dynamics, Jacobian of state function f with respect to the states is calculated as

$$F = \begin{bmatrix} 1 & 0 & -\Delta t V sin(\theta) \\ 0 & 1 & \Delta t V cos(\theta) \\ 0 & 0 & 1 \end{bmatrix} \tag{5.17}$$

At every instant of time, the system predicts the vehicle using this Jacobian and updates the states and covariance matrix when the measurements are received. Along with the position measurements (x, y) received from camera and GPS, lane offset, which is the distance by which the vehicle is off from the center of the lane, is calculated (using lane detection mentioned in chapter 4) and is passed as a measurement to the system, as a function of x and y. Distance from (x_i, y_i) from a line is given by

$$d = \frac{y_i - x_i tan(\alpha) - y_0}{sec(\alpha)} \tag{5.18}$$

where $tan(\alpha)$ is the slope of the line passing amid of the lane and y_0 is its intercept. Hence, the measurement vector becomes $[d_i, x_i, y_i]$, leaving the Jacobian of measurement matrix as

$$H = \begin{bmatrix} -sin(\alpha) & cos(\alpha) & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{5.19}$$

Using this measurement matrix and the system matrix, Kalman filter is operated and location of the vehicle is estimated.

5.4 Results

The plots shown below present the results of the algorithms discussed till now. They include 2D to world coordinate transformation, Kalman filter outputs, error analysis and yaw rate of the vehicle from two different trials. Figures 5.5, 5.9 shows the error analysis with the RMS error observed at both the MSSPs and in the filtered output with respect to time. It can be observed that

the filtered output has the lowest RMS error when the vehicle is in the field of view of the cameras. This is because the Kalman filter gets frequent updates from cameras as well as GPS. Also, it can be observed from figures 5.2, 5.6 that as the vehicle moves away from the field of view of the camera, the 2D to world transformed coordinates become erroneous. Hence, this observation can be pivotal in deciding the repeating distance between MSSPs.

Figures 5.4 and 5.8 shows how the Kalman filter corrects any erroneous initialization of heading angle.

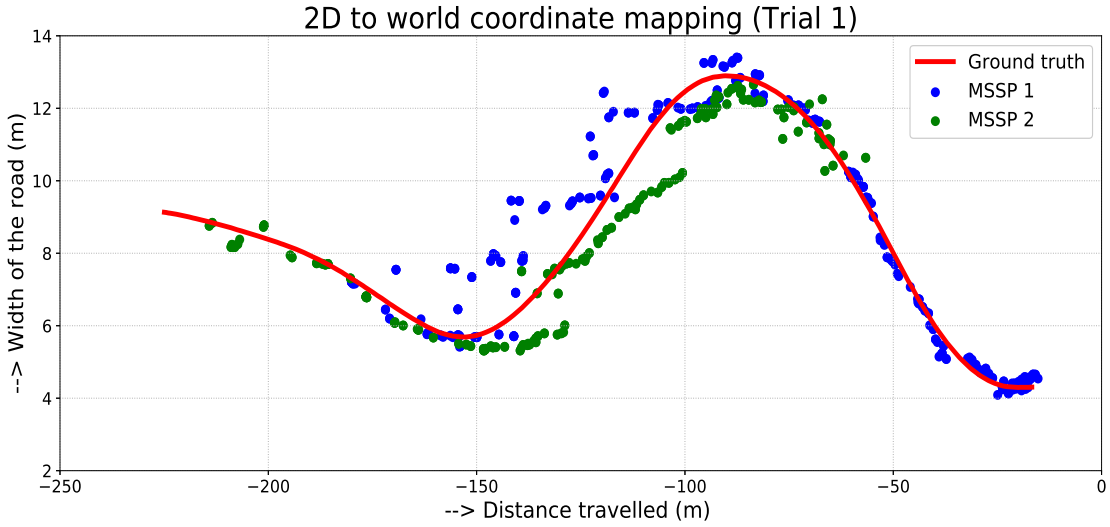

Figure 5.2: Plot showing the calibration and world position estimates using 2 MSSPs, trial 1

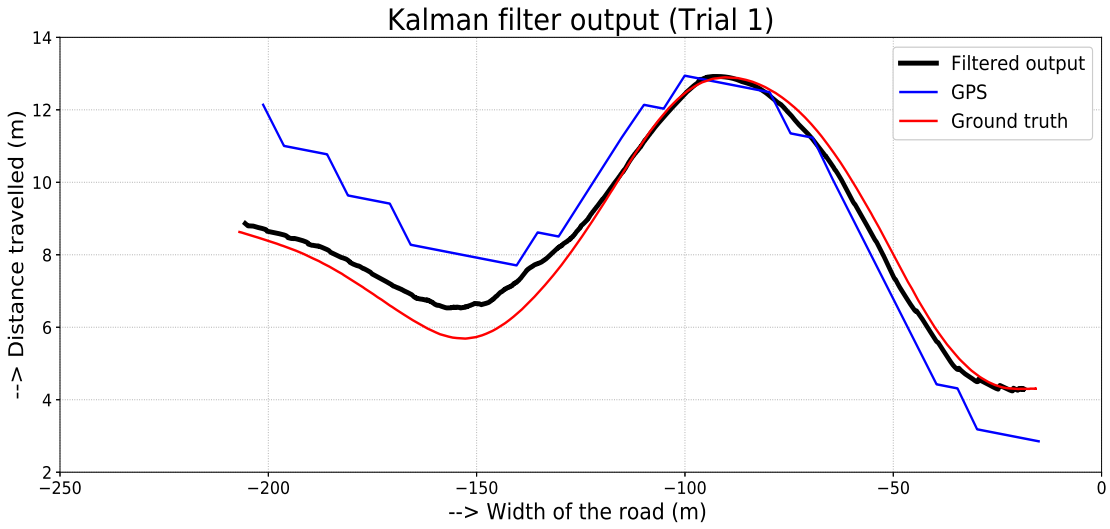

Figure 5.3: Result showing vehicle localization using Kalman estimate, trial 1

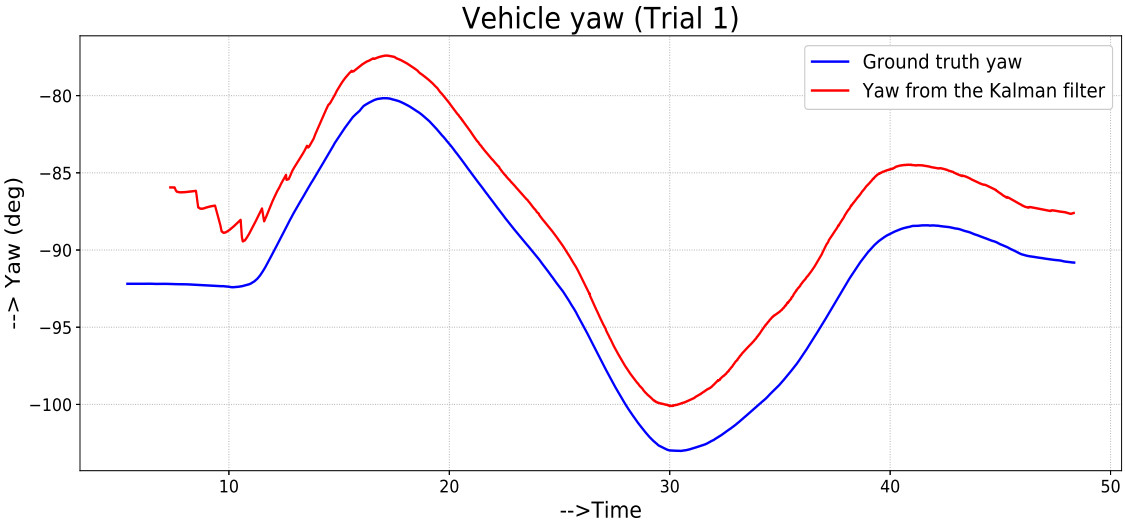

Figure 5.4: Vehicle yaw compared to the ground truth, trial 1

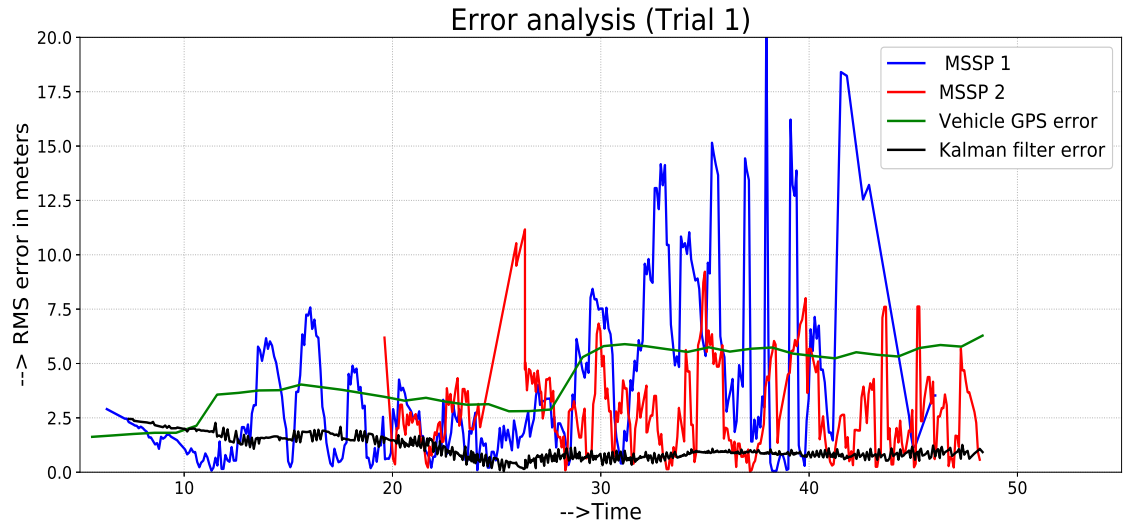

Figure 5.5: RMS error observed at MSSPs and in the Kalman filter output, trial 1

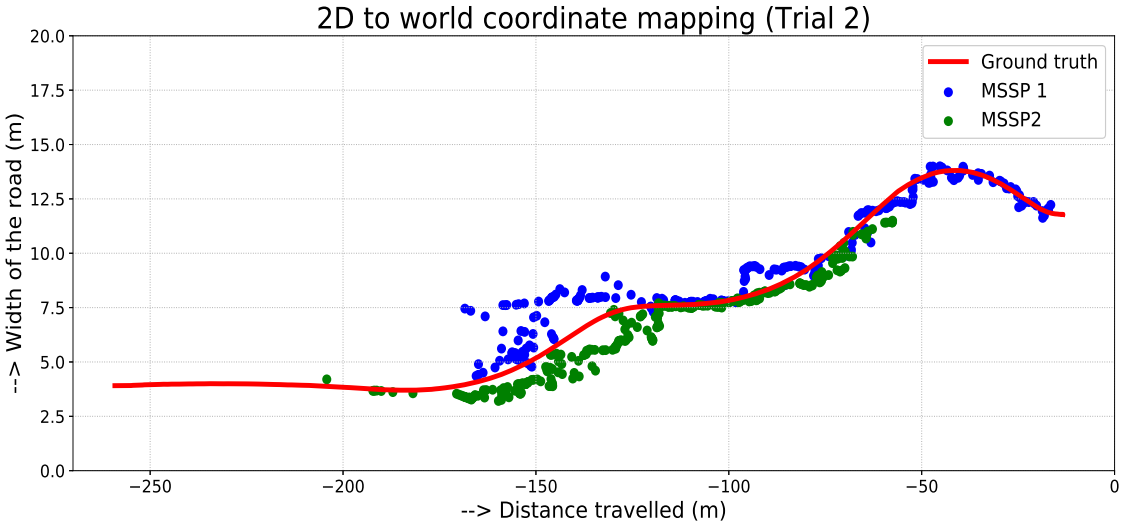

Figure 5.6: Plot showing the calibration and world position estimates using 2 MSSPs, trial 2

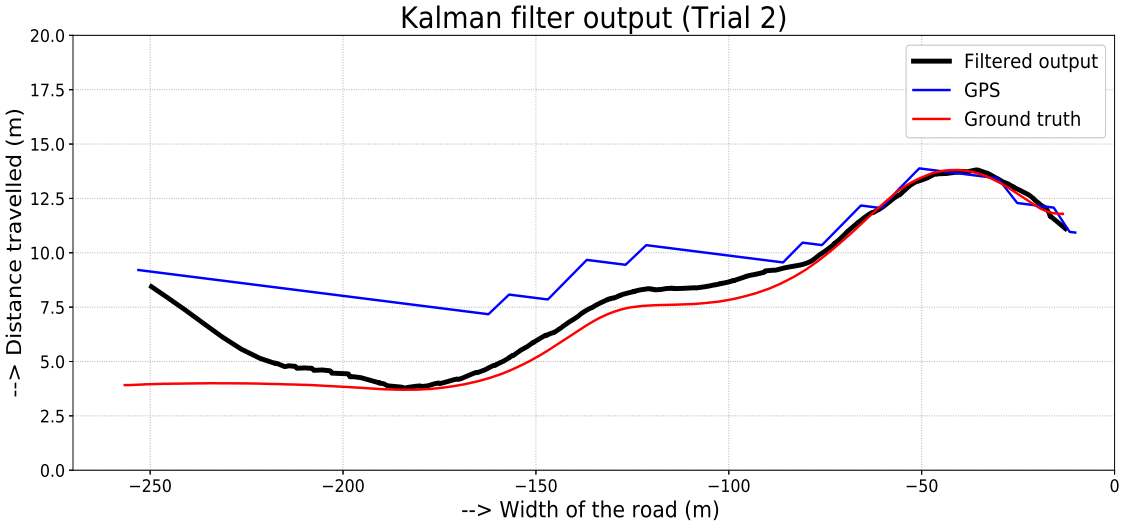

Figure 5.7: Result showing vehicle localization using Kalman estimate, trial 2

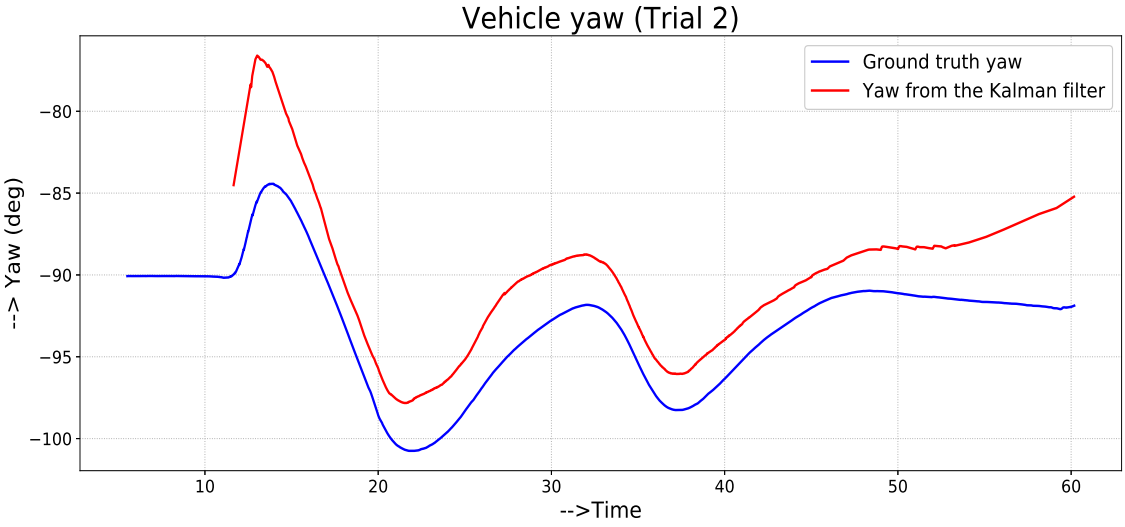

Figure 5.8: Vehicle yaw compared to the ground truth, trial 2

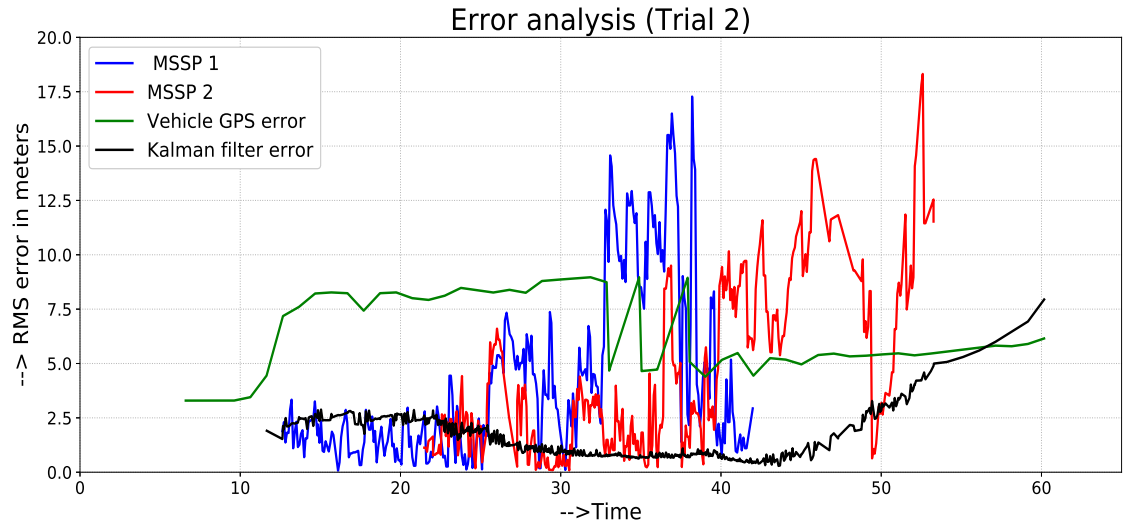

Figure 5.9: RMS error observed at MSSPs and in the Kalman filter output, trial 2

6. CONCLUSION

The main objective of this research is to develop a proof of the concept of Infrastructure Enabled Autonomy architecture. Through distributed hardware in loop simulations and real time realization, localization of the vehicle based on camera sensor has been studied. In this research,

- an efficient camera calibration method for IEA cameras and then to transform the localized 2-D image coordinates of the vehicle to 3-D world coordinates has been devised.
- A robust lane detection algorithm has been implemented. Using the vehicle's position with respect to the center of the lane and the GPS/IMU information on-board the car, an Extended Kalman Filter (EKF) has been developed to obtain the position of the car at a higher rate.
- Simulation and experimental results are presented to corroborate the performance of the proposed algorithms.

The research outcomes shed light on several other aspects of IEA. It can be observed that IEA can detach itself from the global reference frame and can direct the traffic based on its local reference frame. Hence, IEA can be scaled efficiently in GPS denied environments with limited hardware.

A neural network can be implemented to estimate the world coordinates of the vehicle, given 2D image coordinates.The network can be trained as part of calibration and the weights can be used for the coordinates transformation. More advanced and complex planning subsystems and navigation algorithms can be designed to handle dense traffic efficiently. Adaptive algorithms that can learn the driving decisions based on the past situation awareness information can be experimented. Continued research and realization of IEA corridors can help reap the benefits of autonomous transportation in the near future.

REFERENCES

[1] W. H. Organization, "Global Health Observatory Data." http://www.who.int/gho/road_safety/mortality/traffic_deaths_number/en/. [Online; accessed 12-Nov-2018].

[2] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015.

[3] N. H. T. S. Administration, "Fatality analysis report." https://www-fars.nhtsa.dot.gov/Main/index.aspx. [Online; accessed 14-Feb-2019].

[4] M. H. Hebert, C. E. Thorpe, and A. Stentz, *Intelligent unmanned ground vehicles: autonomous navigation research at Carnegie Mellon*, vol. 388. Springer Science & Business Media, 2012.

[5] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE expert*, vol. 11, no. 2, pp. 19–27, 1996.

[6] K. Iagnemma and M. Buehler, "Editorial for journal of field robotics - special issue on the darpa grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 655–656, 2006.

[7] M. Buehler, K. Iagnemma, and S. Singh, "Editorial," *Journal of Field Robotics*, vol. 25, no. 8, pp. 423–424, 2008.

[8] R. Murphy, R. R. Murphy, and R. C. Arkin, *Introduction to AI robotics*. MIT press, 2000.

[9] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[10] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, "Self-supervised monocular road detection in desert terrain.," in *Robotics: science and systems*, vol. 38, Philadelphia, 2006.

[11] D. Stavens and S. Thrun, "A self-supervised terrain roughness estimator for off-road autonomous driving," *arXiv preprint arXiv:1206.6872*, 2012.

[12] A. Nayak, K. Chour, T. Marr, D. Ravipati, S. Dey, A. Gautam, S. Gopalswamy, and S. Rathinam, "A distributed hybrid hardware-in-the-loop simulation framework for infrastructure enabled autonomy," *arXiv preprint arXiv:1802.01787*, 2018.

[13] S. Gopalswamy and S. Rathinam, "Infrastructure enabled autonomy: A distributed intelligence architecture for autonomous vehicles," *arXiv preprint arXiv:1802.04112, accepted IEEE IV 2018*, 2018.

[14] Z. Kalal, K. Mikolajczyk, J. Matas, *et al.*, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, p. 1409, 2012.

[15] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[16] A. De la Escalera and J. M. Armingol, "Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration," *Sensors*, vol. 10, no. 3, pp. 2027–2044, 2010.

[17] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, 2000.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[19] J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 690–703, 2017.

[20] J. Kim and M. Lee, "Robust lane detection based on convolutional neural network and random sample consensus," in *International Conference on Neural Information Processing*, pp. 454–461, Springer, 2014.

[21] L. Xiao, B. Dai, D. Liu, T. Hu, and T. Wu, "Crf based road detection with multi-sensor fusion," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pp. 192–198, IEEE, 2015.

[22] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[23] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.

[24] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.