Automatic Image Based Positioning System

OMSRI KUMAR AEDDULA

Department of Applied Signal Processing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden
This thesis is submitted to the Department of Applied Signal Processing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering with Emphasis on Signal Processing.

Contact Information:
Author:
OMSRI KUMAR AEDDULA
E-mail: aeom16@student.bth.se

Supervisor:
Irina Gertsovich

University Examiner:
Dr. Sven Johansson

Department of Applied Signal Processing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
Abstract

Position of the vehicle is essential to navigate the vehicle along a desired path without any human interference. Global Positioning System (GPS) loses significant power due to signal attenuation caused by construction buildings. A good positioning system should have both good positioning accuracy and reliability. The purpose of this thesis is to implement a new positioning system using camera and examine the accuracy of the estimated vehicle position on a real-time scenario.

The major focus of the thesis is to develop two algorithms for estimation of the position of the vehicle using a static camera and to evaluate the performance of the proposed algorithms.

The proposed positioning system is based on two different processes. First process uses center of mass to estimate the position, while the second one utilizes gradient information to estimate the position of the vehicle.

Two versions of the positioning systems are implemented. One version uses center of mass concept and background subtraction to estimate the position of the vehicle and the other version calculates gradients to estimate the position of the vehicle. Both algorithms are sensitive to point of view of the image i.e height of the camera. On comparing both algorithms, gradient based algorithm is less sensitive to the camera view.

Finally, the performance is greater dependent on the height of the camera position for center of mass positioning system, as compared to the gradient positioning system but the accuracy of the systems can be improved by increasing the height of the camera. In terms of the speed of processing, the gradient positioning system is faster than the center of mass positioning system. The first algorithm, based on center of mass has 89.75% accuracy with a standard deviation of 3 pixels and the second algorithm has an accuracy of 92.26%. Accuracy of the system is estimated from the number of false detected positions.

Keywords: Center of Mass, Gradient, Automatic Positioning System, Background Subtraction.
Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor Irina Gertsovich for the continuous support of my thesis study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my thesis work.

Besides my supervisor, I would like to thank Prof. Tobias C. Larsson for providing me an opportunity and a platform to carry out my thesis at Volvo CE demo site.

I thank my team leader in Product Development Research Laboratory: Ryan Ruvald for his continuous guidance through the research work. Last but not the least, I would like to thank my family: my parents, for having belief in me, support throughout my life and giving freedom to carry out the research in my area of interest.
## Contents

Abstract i

1 Introduction 1
   1.1 Definition ................................ 2
   1.2 Purpose .................................. 2
   1.3 Goals .................................. 2
   1.4 Research Questions ........................... 3
   1.5 Research Methodology ......................... 3
   1.6 Limitations ............................... 3
   1.7 Structure Of The Thesis ......................... 3

2 Related Work 4

3 Methodology 6
   3.1 Apparatus ................................ 6
   3.2 Research Process ............................ 6
      3.2.1 Object Detection ........................ 10
      3.2.2 Object Tracking ......................... 16

4 Results and Discussion 21
   4.1 Center of Mass Positioning System ............... 21
      4.1.1 Input Signal ........................... 21
      4.1.2 Background Subtraction .................... 23
      4.1.3 Smoothing ................................ 24
      4.1.4 Adaptive Thresholding ..................... 26
      4.1.5 Morphological Operations ................... 27
      4.1.6 Center of Mass ........................... 29
   4.2 Gradient based Positioning ..................... 30
      4.2.1 Input Signal ........................... 30
      4.2.2 Gradient Frame ......................... 30
      4.2.3 Smoothing ............................. 32
      4.2.4 Adaptive Thresholding ..................... 33
      4.2.5 Morphological Operations ................... 34
      4.2.6 Contours ............................... 35
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2.7</td>
<td>Contours Mean Estimation</td>
<td>37</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparison of Two Positioning Systems</td>
<td>38</td>
</tr>
<tr>
<td>4.4</td>
<td>Discussions</td>
<td>41</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Center Of Mass Positioning System</td>
<td>41</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Gradient Positioning System</td>
<td>42</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Challenges</td>
<td>43</td>
</tr>
<tr>
<td>5</td>
<td>Conclusions and Future Work</td>
<td>44</td>
</tr>
<tr>
<td>5.1</td>
<td>Conclusions</td>
<td>44</td>
</tr>
<tr>
<td>5.2</td>
<td>Future Works</td>
<td>44</td>
</tr>
<tr>
<td>References</td>
<td>46</td>
<td></td>
</tr>
</tbody>
</table>
List of Figures

3.1 Demo Site with a Vehicle Prototype .................. 7
3.2 Microsoft LifeCam Studio Camera ................... 7
3.3 Overview of Position Estimation ....................... 8
3.4 Overview of Center of Mass Positioning System ........... 9
3.5 Overview of Gradient Based Positioning System .......... 10
3.6 Flow Diagram Representing Background Subtraction ....... 11
3.7 Blue Marked Line Represent the Ground Truth Path Drawn Explicitly for Performance Evaluation ..................... 19
3.8 Fitted Polynomial Curve .................................. 20

4.1 First Frame Of The Limited Frames- Real-Time Video ........ 22
4.2 Middle Frame Of The Limited Frames- Real-Time Video ........ 22
4.3 Last Frame Of The Limited Frames- Real-Time Video .......... 23
4.4 Background Subtracted Middle Frame .................... 24
4.5 Background Subtracted last Frame ........................ 24
4.6 Gaussian Filtered Middle Video Frame .................. 25
4.7 Gaussian Filtered Last Frame ............................ 25
4.8 Adaptive Thresholded Middle Video Frame ................ 26
4.9 Adaptive Thresholded Last Video Frame ................... 27
4.10 Morphological Transformed Mid-Frame .................. 28
4.11 Morphological Transformed Last-Frame .................. 28
4.12 Center Of Mass Of The Vehicle At Mid-Frame ............ 29
4.13 Center Of Mass Of The Vehicle At Last Frame .......... 30
4.14 Gradient Image Of Approximate Middle Frame .......... 31
4.15 Gradient Image Of Last Frame .......................... 31
4.16 Median Filtered Middle Video Frame .................... 32
4.17 Median Filtered Last Video Frame ....................... 32
4.18 Adaptive Thresholded Random Video Frame ................ 33
4.19 Adaptive Thresholded Last Video Frame .................. 34
4.20 Morphological Transformed Random Frame ................ 35
4.21 Morphological Transformed Last-Frame .................. 35
4.22 Contours Of Random Frame ............................. 36
4.23 Contours Of Last-Frame .................................. 36
4.24 Estimated Position Of Random Frame .................... 37
4.25 Estimated Position Of Last-Frame .......................... 38
4.26 Comparison of Two Positioning Systems At Mid-Frame With Dark Blue Line And Light Blue Indicate The Vehicle Position Using Gradient Positioning System And Center Of Mass Positioning System Respectively 39
4.27 Comparison of Two Positioning Systems At Last-Frame With Dark Blue Line And Light Blue Indicate The Vehicle Position Using Gradient Positioning System And Center Of Mass Positioning System Respectively 40
4.28 Comparison of Two Positioning Systems For Curve Fitting Technique With Blue Dots And Red Indicate The Data Points And Fitted Curve Respectively For Two Positioning Systems .......................... 40
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Comparison Between Center Of Mass And Gradient Positioning System</td>
<td>41</td>
</tr>
<tr>
<td>4.2</td>
<td>Overview Of Comparisons Between Various Positioning Systems</td>
<td>41</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Autonomous vehicles can sense the environment and navigate without any human interference or input. The march towards automation of vehicles is well underway and a variety of techniques are in use including but not limited to radar, laser, Global Positioning System (GPS), odometry, and computer vision [1]. For autonomous functioning, the location of the vehicle is one of the key factors. In recent years the use of location information and its potentiality in the development of intelligent systems had led to the design of different positioning systems based on various technologies.

A Local Positioning System (LPS) is a navigation system for obtaining location or position information of vehicles in relation to the local area [2]. The difference from GPS is that, instead of using satellite, LPS works by using three or more short ranging beacons in an unobstructed line of sight. A special type of LPS is the Real-Time Locating System (RTLS) for real-time tracking of a vehicle in a confined region.

RTLS is used to track and identify the vehicles in a real-time scenario. The exact position is crucial for navigating the vehicle along a defined path. RTLS can collect the information either in a passive or active way. Sensors are powered by a battery for former case and whereas, in the latter case, no external supply is required. Sensors transmit the signal upon receiving the request from the reader, where reader receive and process the wireless signals to determine the location [3]. Proposed system tracks the vehicle using a camera and a locating system employing Computer Vision Locating system.

Computer vision is a science that aims to provide the capability to a machine or computer for gaining high-level understanding from digital images or videos [4]. It seeks to analyze and automate the tasks similar to a human visual system. Tasks include methods for acquiring, processing, analyzing and transforming digital images. Transforming in this context refers to converting the visual images to interface with other logical processes and draw out appropriate decisions [4]. The major advantage with camera based positioning system is that it doesn’t require any additional sensors for obtaining the position of the vehicles.
Implementing a positioning system within a confined region is not an easy task, as GPS uses triangulation of satellite signals and has positioning error in meters. Nowadays, people are more focused on developing the techniques to identify the position of the vehicle with millimeter accuracy, even after 20 years of research, this task still remains an open challenge [5].

1.1 Definition

Evaluating whether computer vision discipline assists to gain a high-level understanding of digital video to determine the position of the vehicle using a static camera. More specifically, generating an algorithm to estimate the position of the vehicle and investigate the accuracy of the estimated position.

1.2 Purpose

The main purpose of this thesis is to estimate the position of the vehicle in real-time without any human interference by employing two algorithms and analyzing the efficiency of each algorithm.

1.3 Goals

The goal of the thesis is to propose, implement and evaluate two algorithms to estimate the accurate position of the vehicle for autonomous operation. The goals are subdivided into sub-goals as:

1. Propose and implement two algorithms for estimating the position of the vehicle.
2. Carrying out experiments using the proposed algorithms.
3. Evaluate the performance of the proposed methods considering different factors.
Chapter 1. Introduction

1.4 Research Questions

1. How can the proposed algorithms be implemented for estimating the vehicle position?

2. How efficiently does a camera based positioning system work for a vehicle tracking by sensing and mapping its surrounding environment?

3. Is the height of the camera effects the performance of the system?

4. What are the challenges faced for accurate positioning of the vehicle?

1.5 Research Methodology

In this thesis, both qualitative and quantitative research methodologies are used. Qualitative research was carried out to understand the requirements and prerequisites for automation of vehicle, while quantitative research was carried to measure the performance of the positioning system using the proposed methodologies.

1.6 Limitations

The experimental work is carried out in a real-time scenario with an assumptions of a shadow-free environment, static background, and usage of the single vehicle.

1.7 Structure Of The Thesis

The remainder of the thesis is organized as follows: Chapter 2 deals with the related works about the techniques employed in this project. Chapter 3 presents the methodologies to solve the problems proposed in section 1.1. Chapter 4 presents the results, analysis, and a comparison of the generated algorithms. Chapter 5 concludes the thesis and suggests possible future directions for research in automation of vehicles using computer vision.
Chapter 2

Related Work

Positioning systems have been essential components in autonomous vehicle navigation because, it is still important for the vehicle to know its position relative to its immediate surroundings. There are many different algorithms exists for estimating the position of the vehicle using various sensing technologies such as infrared (IR), ultrasound and radio-frequency (RF) signals [6]. Positioning systems using infrared and ultrasound are line-of-sight needed systems, as IR and ultrasound can’t penetrate objects very well. RF signals are prone to reflection, diffraction, absorption and multi-path fading [6].

Active badges is the first indoor positioning system developed by AT&T Cambridge. It uses miniature infrared beacon, worn by a person, emits a unique code identifier every 15 seconds. Each location in a building covered with a network of IR sensors which detect these transmissions. A central server collects this data from fixed IR sensors around the building, gathers into a central data bank, and the location of the badge can thus be determined [7].

AT&T Cambridge had developed an ultrasonic tracking which provided a better and more accurate positioning system than the previous active badges [8]. Users and objects are tagged with ultrasonic tags identified as "bats", that emit periodic ultrasonic signals to the receivers mounted across the ceiling. The problem with this system was that it requires a large number of receivers across the ceiling and has quite sensitive alignments.

GPS is the most popular outdoor positioning system to find the location and the position of the objects. It employs triangulation process to determine the physical locations [9]. GPS signals are highly attenuated and reflected by the construction materials resulting in the maximum positional error of about 6-12 m [9].

Cricket positioning system is an ultrasonic positioning system. MIT laboratories had developed a small ultrasonic devices with transmitter and receiver application [5]. It had a positional accuracy of 1-2 cm but has a limited range of field.
Chapter 2. Related Work

Dolphin positioning system sends and receives RF and ultrasonic signals. It requires minimum manual configuration [10]. The disadvantage of this system is a limited range of field and the nodes are needed to be attached to various indoor objects.

Received Signal Strength Information (RSSI) system estimates the distance between sender and receivers. It uses RF signals and estimates the position by calculating the distance to the object from the transmitter using triangulation or trilateration techniques [11]. RF are prone to attenuation, reflection, diffraction, absorption, and multipath. Receivers perform poorly in indoor environments.

Ultrawide band impulse radio signals are employed for indoor location estimation. Time of Arrival (TOA) technique [12] is used to estimate the location. Factors such as reflections, multipath and attenuations effect the accuracy of the system.

RFID technology is non-contact and non-line-of-site locating system. They can work at high speeds and their RF tags can be read in any environment. They use received signal strength and triangulation to estimate the position of the system [12]. They are confined to room scale and require installing multiple base stations.

John Krumm et al., [13] developed a computer vision locating system which tracks people with multi-cameras and generates intelligent environment set up to look like a living room. It requires multiple cameras to cover all the corners and it is expensive.

Kim et al., [14] proposed a vision based localization system using two cameras. New feature initialization and feature matching techniques are used with two cameras to locate people. The work aimed for intelligent robot development to recognize robot’s environment.

Fingerprinting method is an another technique integrated with RFID technique. In first stage, off-line stage, Received Signal Strength(RSS) and physical coordinates are collected from RF transmitter at a reference point and stored in a database which is fingerprint. Second stage, On-line stage, user samples the RSS pattern [15] and searches for similar pattern in database to find the best possible position. The accuracy is around 5m and the system is not compatible with real-time scenario’s.

Pozyx positioning system uses ultrawide band technology to estimate the position. It uses an anchor and a tag with trilateration technique to estimate the location. For one tag, four anchors are needed for best possible position estimation. It uses Inertial measurement unit(IMU) to calculate the required values. More anchors need to be deployed for accurate position estimation [16].
Chapter 3

Methodology

This chapter serves as an outline of the research method used in the thesis. Two different methodologies have been employed for estimating the position of the vehicle as mentioned below:

1. Center of mass based positioning system.
2. Gradient based positioning system.

The algorithms for each of two proposed and developed systems were implemented in C++, and Python programming languages, making use of OpenCV library. Performance evaluation of the positioning systems are done in MATLAB [17]. OpenCV is a library of programming functions designed for computational efficiency and with a strong focus on real time applications [18]. It supports a wide variety of languages such as C++, Python, java, etc., and is available on a variety of platforms including Windows, Linux, IOS, OS X, and Android. The major advantage with OpenCV is it supports a multitude of algorithms related to computer vision [4].

3.1 Apparatus

The experimental work is carried out on a special demo site designed by Volvo CE with a vehicle prototype, and a Microsoft LifeCam Studio camera for collecting the data as shown in figures 3.1 and 3.2.

3.2 Research Process

The main aim of this research is to implement and examine the automatic positioning system. Estimating the position of the vehicle leads to automation of the vehicle. Two different methodologies have been proposed for automatic positioning of the vehicle.

(a) Center of mass based positioning system
(b) Gradient based positioning system
Figure 3.1: Demo Site with a Vehicle Prototype

Figure 3.2: Microsoft LifeCam Studio Camera
In each methodology, vehicle position estimation can be achieved by segregating the task into three subtasks as shown in figure 3.3:

(i) Real-time Video: Real time data collected from the static camera as a digital input signal for detecting and tracking the object.

(ii) Object Detection: It deals with detecting instances of objects in the digital input signal.

(iii) Object Tracking: It deals with the process of locating the detected object in the digital input signal.

The process involves collecting real time video data from a static camera fixed at a moderate height above the ground. The data is processed to detect moving objects present in the frame by object detection algorithm and the object tracking algorithm is employed to locate the detected object.

**Center of mass positioning system**

Overview of the center of mass positioning system is shown in figure 3.4. Background Subtraction algorithm is employed to extract foreground information by eliminating background from the digital input signal acquired from the static camera. Background subtracted frame is smoothened to reduce the noise and morphological operations are applied to the filtered frame. Transformed video frames are analyzed to obtain white pixel locations to estimate the center of mass of the vehicle.
Chapter 3. Methodology

Gradient based positioning system

Overview of the gradient positioning system is shown in figure 3.5. The gradient of an image represents the change in intensity level of the pixel of the video frame. Gradients of an image provide useful information about the object. The gradient value of a pixel is yielded by convolving the pixel in the video frame with a desired filter, where the filter can be eg. Sobel filter as described further in the text. Generated gradient values are enhanced to modify the content of the video frame for further processing. Modified video frames are transformed by applying morphological operations and these transformed video frames are further processed by drawing contours around the detected vehicle. The position of the vehicle is then estimated using the arithmetic mean of the pixels coordinates in the contours.
3.2.1 Object Detection

Object detection deals with detecting instances of a vehicle in a real-time video. Detection of a vehicle plays a crucial role in estimating the position accurately. It is important to implement a suitable algorithm that would accurately detect the vehicle.

3.2.1.1 Background Subtraction

Background subtraction is a technique of extracting image’s foreground from the video frames for further processing. It is also called as foreground detection. A robust algorithm is essential to handle long-term scene changes and lighting changes in the environment. Background Subtraction is performed using frame differencing technique where the first frame is background frame without any moving objects and all other frames are foreground frames [19]. The video frame with background information and the video frame with both the foreground and background information are shown in figures 3.6(a) and 3.6(b) respectively. The resultant frame after background subtraction method is shown in figure 3.6(c).

3.2.1.1.a Frame Differencing

Vehicle detection begins with separating foreground from background in each frame. The simplest arithmetic equation to segregate the vehicle from its background is:

\[ I_F(x, y) = I_c(x, y) - I_b(x, y), \]  

(3.1)
where
\[ I_F(x,y) = \text{foreground frame intensity value at pixel location}(x,y), \]
\[ I_c(x,y) = \text{current frame intensity value at pixel location}(x,y), \]
\[ I_b(x,y) = \text{background frame intensity value at pixel location}(x,y). \]

Equation (3.1) refers to subtracting each pixel in background frame with respective pixel in foreground frame.

(a) Background Image

(b) Foreground Image

(c) Image with no Background

Figure 3.6: Flow Diagram Representing Background Subtraction
3.2.1.2 Image Gradient

An image gradient is the directional change in the intensity of the pixels in the video frames. Image gradients can be used to extract information from the video frames. Gradient is a vector which points in the direction of steepest ascent. Gradient images are generally obtained by convolving the video frame with a filter and can be calculated along two directions i.e., horizontal (x-plane) and vertical (y-plane) by measuring the intensity change of pixel with respect to some reference pixel [20]. The gradient of image \( I \) at \((x,y)\) is defined as two dimensional column vector, according [21]

\[
\nabla I = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{pmatrix},
\]

(3.2)

where

\( \nabla I \) = gradient vector,

\( g_x \) = gradient in the x direction,

\( g_y \) = gradient in the y direction,

Gradient frames provide more information about edges. Using a suitable operator for estimating the gradient magnitude reduce the inaccuracies in the video frames. Sobel filter is one of the most widely used filter for obtaining image gradients. It is a discrete differentiation and joint Gaussian smoothing operator [22]. It is more resistant to noise. The horizontal gradient is a partial derivative of image in x direction and vertical gradient is a partial derivative of image in y direction, according to equation (3.2) or the horizontal and vertical gradients are computed by convolving video frame with a kernel of an odd size. A kernel is a small matrix used to apply predefined effects to the video frame.

- **Horizontal gradient** \( G_x \) is computed by convolving video frame \((I_t)\) with a kernel \( K \) with an odd size. \( G_x \) is computed according [20]

\[
G_x = K \ast I_t,
\]

(3.3)

where 3x3 kernel \( K \) is given by the matrix

\[
\begin{pmatrix}
-1 & 0 & +1 \\
-2 & 0 & +2 \\
-1 & 0 & +1
\end{pmatrix},
\]

\( G_x \) is discrete approximation of \( g_x \) in equation (3.2), and \( \ast \) is 2D convolution.
Chapter 3. Methodology

- **Vertical gradient** $G_y$ is computed by convolving video frame ($I_t$) with a kernel $K'$ with an odd size. $G_y$ is computed [20]

\[
G_y = K' \ast I_t, \tag{3.4}
\]

where 3x3 kernel $K'$ is given by the matrix

\[
\begin{pmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
+1 & +2 & +1
\end{pmatrix},
\]

$G_y$ is discrete approximation of $g_y$ in equation (3.2), and $\ast$ is 2D convolution.

Gradient magnitude $G$ at each pixel is approximated by combining both results above according [20]

\[
||\nabla G|| = \sqrt{G_x^2 + G_y^2} \tag{3.5}
\]

Gradient direction $\theta$ can be computed by

\[
\theta = \arctan\left(\frac{G_y}{G_x}\right) \tag{3.6}
\]

### 3.2.1.3 Smoothing

Image processing operations are performed to improve the quality of the frames for further processing. Smoothing is simple and frequently used image processing operation. It is also referred as "Blurring". The main intention is to reduce the noise level in the video frames.

Smoothing is performed by applying a filter to the sequences of images or frames. Filtering is a neighborhood operation, in which the pixel value in the resultant frame is determined by applying an algorithm to the neighborhood pixel value of the corresponding pixel value in the input frame.

Gaussian filter with size 5x5 has the highest noise suppression capability [23]. It works on the principle of point spread function. Point Spread Function (PSF) describes the response of the imaging system to a point object or function [24]. The filter outputs a weighted average of the neighborhood pixels for the particular pixel of the input video frame. The average weights more towards the central pixel, because of this property Gaussian filter smoothen the video frame while preserving the edge information. The degree of smoothness depends on the value of standard deviation of the gaussian distribution $\sigma$. Larger standard deviation implies larger convolution kernel required to accurately represent the pixel values and probability of losing edge
information. Generally, the noise resides at high-frequency levels so to reduce the noise, low-frequency signals are to be passed through the filter. A Gaussian filter is a low-pass filter which removes sudden discontinuities of grey levels in the kernel size, hence it reduces the noise levels significantly. The filter approximates the pixel value with its weighted average of each pixels neighborhood, with the average weighted more towards the central pixel. Gaussian kernel coefficients are sampled from the 2D Gaussian function according \[25\]

\[
G(x, y) = \frac{1}{(2\pi\sigma^2)} e^{-\frac{x^2 + y^2}{2\sigma^2}}, \tag{3.7}
\]

where

\((x, y)\) = pixel location in space, and

\(\sigma\) = standard deviation of the Gaussian distribution.

Median filter is most widely used when complete edge information is required. The filter replaces each pixel value by the median of its neighborhood pixels that fits in the kernel. This filter is highly effective in removing salt and pepper noise from the video frames [26]. The most interesting fact is the pixel value is replaced by some pixel value in the neighborhood \(w\) [27] [28]. The kernel size must be positive odd integer.

\[
I(x, y) = \text{median}[I(i, j) \quad \forall (i, j) \in w], \tag{3.8}
\]

where \(w = \) a neighborhood size of the median filter.

3.2.1.4 Adaptive Thresholding

Setting up a threshold level can change the video frame or image representation into a binary image and are easier to extract information. Thresholding can be of two types:

(a) Simple Thresholding

(b) Adaptive Thresholding

In simple thresholding, the threshold level is global i.e every pixel in the video frame is examined with a same threshold value, but it may not be a good approach when there is a change in illumination. Adaptive thresholding has a different threshold level for different regions of the video frames [4]. Different threshold level provides better results for video frames with varying illumination. The threshold level is determined either by the mean of the neighborhood area or by the weighted sum of neighborhood
values where the weights are of a Gaussian type. The major advantage with adaptive thresholding is clear identification of uncertainties that needs to be removed in further processing of video frames.

Threshold operation is given according [4]

$$\theta(I, t) = \begin{cases} 1 & \text{if } I \geq t \\ 0 & \text{otherwise} \end{cases}, \quad (3.9)$$

where

$I = $ Image pixel intensity and
$t = $ threshold level.

### 3.2.1.5 Morphological Operations

Morphology is a set of non-linear image processing operations that process video frames using a predefined structuring element also known as a kernel [4]. Morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore they are more suitable for binary images. The operations are based on set theory [29]. To perform such operation, the binary image is convolved with the structuring element using set operators such as intersection, union, inclusion, and complement and a threshold level defines the output pixel value. The value of each pixel in the output video frame is based on comparison with the neighbors of the corresponding pixel in input video frame.

Morphological transformations reduce the uncertainties in the video frame on a significant level. To reduce the uncertainties, correct morphological transformation needs to be applied. These operations require a binary image with a structuring element or kernel [30] which defines the nature of the operation. A rectangular structuring element has been defined because of the shape of the vehicle.

Number of 1’s inside each structuring element on scanning a binary image $I$ is given by according [4]

$$c = I \otimes s \quad (3.10)$$

Dilation and erosion are two basic morphological operations and opening, closing and majority are combinations of the two operations. Dilation and erosion are the two opposite operations where erosion removes pixels and dilation adds pixels to the boundaries of the object in the video frames [4]. The number of pixels added or removed depends on the shape and size of the structuring element.
• **Dilation:** The intensity of the output pixel is the maximum value of all the pixels that fits within the structuring element according [4].

\[
dilate(I, s) = \theta(c, 1) \tag{3.11}
\]

• **Erosion:** The intensity of the output pixel is the minimum value of all the pixels that fits in the shape and size \( S \) of the structuring element according [4].

\[
erode(I, s) = \theta(c, S) \tag{3.12}
\]

• **Closing:** The closing of the video frame \( I \) by a structuring element \( s \) is a dilation followed by erosion according [4].

\[
closing(I, s) = erode(dilate(I, s), s) \tag{3.13}
\]

### 3.2.2 Object Tracking

#### 3.2.2.1 Center of Mass

The center of mass is a unique point in the center of distribution of mass in space where the sum of the points relative to the weighted position vectors is zero [31]. In image applications, centroid and center of mass are two equivalent terms.

Moments are scalar quantities used to characterize the function and to capture significant features. Image moments is a weighted average of the image pixel intensities, chosen to characterize the shape of the vehicle in the video frames [32]. These moments provide the center of mass of the detected vehicle, alongside with area and orientation of the object. A moment \( \mu \) of orders \( m^{th} \) and \( n^{th} \) at \( I(x,y) \) is given by according [33]

\[
\mu_{mn} = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} x^m y^n I(x,y), \tag{3.14}
\]

where \( m \) and \( n \) represent order moments along \( x \) and \( y \) directions, respectively.

Zeroth image moment \( (\mu_{0,0}) \) represents the area of the binary image as according [33]

\[
\mu_{00} = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} I(x,y), \tag{3.15}
\]

where

\[
I(x,y) = \text{Intensity of the pixel},
\]

\[
w = \text{width of the video frame}, \quad \text{and}
\]

\[
h = \text{height of the video frame}.
\]
Center of mass \((c_x, c_y)\) of the detected object is given by calculating two coordinates \(\mu_{01}\) and \(\mu_{10}\) normalized by \(\mu_{00}\) according \([33]\)

\[
(c_x, c_y) = \left( \frac{\mu_{10}}{\mu_{00}}, \frac{\mu_{01}}{\mu_{00}} \right),
\]

(3.16)

where

\[
\mu_{01} = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} y I(x,y) \quad \text{and} \quad \mu_{10} = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} x I(x,y).
\]

### Contours

Contours are a curve joining all continuous points, having the same intensity \([34]\). Contours are the shape analyzers representing the outline of the detected object. It can be described as a curve joining all the continuous points with same color and intensity. Contour tracing is a technique applied to binary images to extract their boundary by joining all the continuous points. Contour tracing \([34]\) is also known as border following. Contours produce best results when the input image is a binary image \([35]\). The shape information is extracted from an ordered sequence of the boundary pixels. Contour tracing is often a major contributor to the efficiency of estimating the position of the vehicle. Contours approximation method decide the storage of a number of boundary points. In the case of simple contour approximation, only boundary points along the edges are stored else all the boundary points are stored \([34]\).

### Contour Mean

The terms arithmetic mean and sometimes average are used synchronously to refer to a central value of a discrete points \([20]\). For the contour data set, mean value of x and y coordinates are computed as according \([20]\)

\[
[x', y'] = \left[ \frac{1}{p} \sum_{i=0}^{p-1} i, \frac{1}{p} \sum_{j=0}^{p-1} j \right] \quad \forall (i,j) \in c
\]

(3.17)

where

\[
x' = x\text{-coordinate of the contour mean},
\]

\[
y' = y\text{-coordinate of the contour mean},
\]

\[
c = \text{pixels belonging to contours},
\]

\[
[i,j] = x \text{ and } y \text{ location of the pixels in the detected contours, and}
\]

\[
p = \text{number of } x \text{ or } y \text{-coordinates of the pixels in the contours}.
\]
3.2.2.3 Performance Evaluation

A tedious task had been done to evaluate the performance of the proposed algorithms. A marker line was drawn on the site along the desired path of the vehicle. An expert in driving the vehicle was instructed to drive on the marked path and the video was recorded covering a total of 517 frames at a frame rate of 25 by covering 90% of the driving path. The background frame consisting of a marked line was used to evaluate the performance. MATLAB simulator was used to detect the number of false detection points. The estimated points were considered as false detection if:

1. The point lie outside the marked line, and

2. The point is in the backward direction of the past location on the marked line.

Accuracy is defined in terms of range and percentage of true estimation. Range accuracy is measured by calculating the deviation of the estimated point (not classified as false detected point) from the ground true position. The accuracy in percentage of true estimation is measured by calculating the ratio of difference between the number of detected points and number of false detected points to the number of total estimated points. In addition, every 50th frame is manually analyzed by plotting the frame in MATLAB and verifying the estimated position.

\[
\text{Accuracy} = \left( \frac{T - F}{T} \right) \times 100
\]

where

\( T = \) number of estimated points, and
\( F = \) number of false detected points.
Curve Fitting

Curve fitting is another technique to estimate the performance of the system in terms of accuracy. Approximately, a third degree polynomial curve is fitted on the desired path

\[ f_1(x) = p_1 x^3 + p_2 x^2 + p_3 x + p_4 \]  

(3.19)

where coefficients with 95% confidence bounds

- \[ p_1 = 2.472 \times 10^{-6} \ (2.337 \times 10^{-6}, \ 2.607 \times 10^{-6}) \]
- \[ p_2 = -0.000939 \ (-0.001051, \ -0.0008265) \]
- \[ p_3 = 0.2012 \ (0.1726, \ 0.2297) \]
- \[ p_4 = 72.55 \ (70.36, \ 74.73) \]

Figure 3.8 represents the polynomial curve fitting using the ground truth data points. The estimated points are plotted on the curve and the performance is analyzed. Points that lie outside the curve are false detected points.
Figure 3.8: Fitted Polynomial Curve
Chapter 4

Results and Discussion

In this chapter, detailed analysis of the research process carried out is presented. Two methodologies proposed in the thesis were implemented and simulated in C++ and Python programming environment but for convenience, only simulated results in a C++ environment were shown. Input signals were obtained from Microsoft Lifecam Studio static camera, as shown in figure 3.2 with a resolution of 1920 x 1080 pixels placed at a certain height above the demo site. The real-time video was limited to a certain number of frames, assuming it to be 517. The experiment was carried out with an assumption of capturing real-time video in a shadow free region.

4.1 Center of Mass Positioning System

4.1.1 Input Signal

The input signal is a real time video stream capturing the movement of the vehicle. It consists of a stream of the digital images. Figures 4.1, 4.2, and 4.3 represent the first, middle, and last frame of the limited-frame real-time video respectively. First figure is considered as a background image, it shows the view of the demo site without any vehicle. The other two figures show the vehicle prototype in two different positions.
Chapter 4. Results and Discussion

Figure 4.1: First Frame Of The Limited Frames- Real-Time Video

Figure 4.2: Middle Frame Of The Limited Frames- Real-Time Video
4.1.2 Background Subtraction

Background subtraction was utilized to obtain foreground information in each video frame. By subtracting each frame from the first frame (Background frame), foreground information is obtained. Figures 4.4 and 4.5 show the foreground information of the middle and last frame of limited-frame real-time video. The resulting video frames are gray scale images clearly showing the information regarding the vehicle position in different frames. The trail of pixels in figures 4.4 and 4.5 outside the cluster of grey pixels were considered to be a noise and should be removed before the estimation of the vehicle position.
Chapter 4. Results and Discussion

4.1.3 Smoothing

Obtained gray scale video frames are filtered to reduce the noise level by convolving with a Gaussian filter of predefined kernel size 5x5. Figures 4.6 and 4.7 show the blurred gray scale images filtered using Gaussian kernel of size 5x5 window and with a standard deviation of $\sigma = 3.0$ pixels, according to equation (3.7). The trail of pixels
were not clearly visible in figures 4.6 and 4.7 as compared to figures 4.4 and 4.5. To extract more information gray scale images were converted to binary images.

![Figure 4.6: Gaussian Filtered Middle Video Frame](image)

![Figure 4.7: Gaussian Filtered Last Frame](image)
4.1.4 Adaptive Thresholding

Smoothed frames are further utilized to extract information from the video frames. Mean value adaptive threshold is applied to the smoothened frames. The threshold is based on the mean value of the neighborhood area, where the area represents block size. If the pixel satisfies the threshold condition, according to equation (3.9) then the pixel is replaced by the value 1.

The threshold value $T(x,y)$ was the mean of block size $7 \times 7$ neighborhood of $(x,y)$. The block size was of an odd size, as odd size kernels have only one center pixel. If the threshold type is of the inverse binary image then the frame pixel value will be zero if the condition is satisfied.

Figures 4.8 and 4.9 show the smoothened video frames after setting up an adaptive threshold values rather than a simple threshold level and also the noise pixels that needs to be removed.

![Adaptive Thresholded Middle Video Frame](image)

Figure 4.8: Adaptive Thresholded Middle Video Frame
4.1.5 Morphological Operations

For morphological transformations, the size of the kernel depends on the frame size, for convenience, the frame size of 1920x1080 pixels had been resized to 480x240. The binary video frame was eroded and dilated. Dilation and closing operations are applied to the resultant transformed video frame.

Figures 4.10 and 4.11 show the binary image after performing the morphological operation on mid and last frames respectively. The rectangular structuring element transformed the cluster of white pixels into a rectangular shape.
Figure 4.10: Morphological Transformed Mid-Frame

Figure 4.11: Morphological Transformed Last-Frame
4.1.6 Center of Mass

After successful morphological operations, the major task was to find the center of mass which replicates the vehicle position. The transformed video frame is analyzed according to equation (3.16) to find the unique point where the weighted average of the pixel is zero in the cluster of white pixels. This unique point represents the vehicle center of mass and its location in the site. Figures 4.12 and 4.13 show the real-time video frame with a series of black spots representing the center of mass of the vehicle. The unique point was overlaid on the video frame in parallel in such a way that center of mass of every frame was located on the screen.

Figure 4.12: Center Of Mass Of The Vehicle At Mid-Frame


4.2 Gradient based Positioning

4.2.1 Input Signal

The Input signal is the same input signal used for center of mass positioning system. Figures 4.1, 4.2 and 4.3 represent the first, middle, and last frame of the limited-frame real-time video respectively.

4.2.2 Gradient Frame

Gradients along the horizontal and vertical axis with gradient magnitude at each pixel were calculated. The gradient was calculated by convolving the frame with a 3x3 kernel, according to equation (3.3) and (3.4). Sobel operator was used for calculating the gradient values along both the axes and the gradient values were used to calculate the magnitude of gradient according to equation (3.5) for each pixel. Gradient frames detect the vehicle and frame differencing eliminates background from foreground.

Figures 4.14 and 4.15 show the gray scale gradient frames after calculating the magnitude of gradient according to equation (3.5) for each pixel. Image gradient method shown in figures 4.14 and 4.15 has less noise level compared to background subtraction method shown in figures 4.4 and 4.5. The trail of pixels outside the cluster of pixels were considered as noise and should be removed before the estimation of vehicle position.
Figure 4.14: Gradient Image Of Approximate Middle Frame

Figure 4.15: Gradient Image Of Last Frame
4.2.3 Smoothing

Figures 4.14 and 4.15 show the edge information of the vehicle. To preserve the edge information a median filter of size 3x3 sampling window was used, according to equation (3.8). Median filter preserves edges while reducing noises. Figures 4.16 and 4.17 show the smoothened image with a median filter of size 3x3 sampling window and also shows the discontinuous edges.

Figure 4.16: Median Filtered Middle Video Frame

Figure 4.17: Median Filtered Last Video Frame
4.2.4 Adaptive Thresholding

Median filtered frames were further utilized to extract the information. Mean value adaptive threshold was applied to the smoothened frames. The threshold was based on the mean value of the neighborhood pixels with a block size 7x7. If the pixel satisfies the threshold condition, according to equation (3.9) then the pixel is replaced by the value 1. By filtering the video frames, typical noise such as salt and pepper has been significantly reduced.

The threshold value \( T(x, y) \) is the mean of 7x7 block size neighborhood of \((x, y)\). The block size is of odd size, as odd size kernels have only one center pixel. The gray scale image was converted to binary image where the image pixel value will be zero if the condition is not satisfied.

Figures 4.18 and 4.19 show the smoothened binary video frames after filtering with an adaptive threshold value. The major advantage with this method is clear identification of uncertainties and discontinuous edges that needs to be removed in further processing of video frames.

Figure 4.18: Adaptive Thresholded Random Video Frame
4.2.5 Morphological Operations

For morphological operations, the size of the kernel depends on the frame size, for convenience, the frame size of 1920x1080 pixels had been resized to 480x240. The binary image was dilated followed by closing operation.

Figures 4.20 and 4.21 show the binary image after performing morphological operation on random and last frames respectively. The rectangular structuring element transformed the cluster of white pixels into a rectangular shape.
4.2.6 Contours

Based on several observation, contours produced best results when the input image was a binary image. External contours were extracted for every frame in the real-time video and a simple contour approximation method was selected to store only the edge point contours.
Figures 4.22 and 4.23 show the closed contours of two different frames. These closed contours are essential to estimate the position of the vehicle in each frame.

Figure 4.22: Contours Of Random Frame

Figure 4.23: Contours Of Last-Frame
4.2.7 Contours Mean Estimation

External contours provide the shape estimation with pixel coordinates. The mean value of the pixel coordinates belonging to the contours estimate the positional coordinates of the vehicle by taking the weighted average of x and y coordinates of the external contours, according to equation (3.17). In each frame, the mean value of x coordinate was estimated by summing up all the pixel coordinates along x-axis belonging to the contours divided by the total number of pixel coordinates along x axis in the contours, similarly the mean value of y coordinates is estimated. Figures 4.24 and 4.25 show the vehicle movement in real-time scenario. The movement of the vehicle is represented by series of black spots which are the mean value along x and y direction of contours in each frame.

Figure 4.24: Estimated Position Of Random Frame
4.3 Comparison of Two Positioning Systems

Two positioning systems were tested with the same video track of frame rate 25 fps. The camera was placed at a height to capture the top view before the turn of the vehicle and side view after the turn of the vehicle along the desired path. On analyzing both the positioning systems, the following observations are made:

1. Center of mass positioning system is slower in estimating the vehicle position by 3 frames and gradient positioning system is slower by 1 frame with respect to the real time video.

2. Both the systems follow the line-of-path.

3. There is difference between two positioning system estimation but the difference reduces when the view of the camera changes to side view.

4. The estimated position of the two systems approximately coincides when the point of view changes from top view to side view.

5. Gradient positioning system has greater accuracy than center of mass positioning system.
Figures 4.26 and 4.27 show the comparison of two positioning systems at mid and last frame respectively. The dark blue line indicate the vehicle position using gradient based positioning system and the light blue line indicates the vehicle position using center of mass positioning system.

Figure 4.28 shows the approximation method for estimating accuracy of the positioning system. Red line represents the fitted curve and blue line dots represents the estimated points. The left line is the traced path of the vehicle using gradient positioning system and the right line is using center of mass positioning system.
Figure 4.27: Comparison of Two Positioning Systems At Last-Frame With Dark Blue Line And Light Blue Indicate The Vehicle Position Using Gradient Positioning System And Center Of Mass Positioning System Respectively

Figure 4.28: Comparison of Two Positioning Systems For Curve Fitting Technique With Blue Dots And Red Indicate The Data Points And Fitted Curve Respectively For Two Positioning Systems
### Table 4.1: Comparison Between Center Of Mass And Gradient Positioning System

<table>
<thead>
<tr>
<th>Factor</th>
<th>Center of Mass Positioning System</th>
<th>Gradient Positioning System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame rate</td>
<td>25 fps</td>
<td>25 fps</td>
</tr>
<tr>
<td>Number of frames(assumed)</td>
<td>517</td>
<td>517</td>
</tr>
<tr>
<td>Delay in frames</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Number of false detections</td>
<td>53</td>
<td>40</td>
</tr>
<tr>
<td>Accuracy(True Estimation)</td>
<td>89.75%</td>
<td>92.26%</td>
</tr>
</tbody>
</table>

### Table 4.2: Overview Of Comparisons Between Various Positioning Systems

<table>
<thead>
<tr>
<th>System</th>
<th>Range</th>
<th>Signal type</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Badge</td>
<td>5m</td>
<td>Infra Red</td>
<td>7cm</td>
</tr>
<tr>
<td>Active Bat</td>
<td>50m</td>
<td>Ultrasound</td>
<td>9 cm</td>
</tr>
<tr>
<td>Cricket</td>
<td>10m</td>
<td>Ultrasound</td>
<td>2 cm</td>
</tr>
<tr>
<td>Dolphin</td>
<td>Room Scale</td>
<td>Ultrasound</td>
<td>2 cm</td>
</tr>
<tr>
<td>Ultra-Wide Band</td>
<td>15m</td>
<td>RF</td>
<td>10 cm</td>
</tr>
<tr>
<td>RFID</td>
<td>Indoors</td>
<td>RF</td>
<td>2 m</td>
</tr>
<tr>
<td>Fingerprinting</td>
<td>Indoors</td>
<td>RF</td>
<td>1.7 m</td>
</tr>
<tr>
<td>Computer Vision</td>
<td>Room Scale</td>
<td>Images</td>
<td>10 cm</td>
</tr>
<tr>
<td>Center of Mass</td>
<td>Room Scale</td>
<td>Images</td>
<td>&lt; 2 cm</td>
</tr>
<tr>
<td>Gradient</td>
<td>Room Scale</td>
<td>Images</td>
<td>&lt; 1 cm</td>
</tr>
</tbody>
</table>

### 4.4 Discussions

#### 4.4.1 Center Of Mass Positioning System

The input signal of 1920x1080 resolution was resized to 480x240 resolution for visual convenience. Theoretically, background subtraction could be a great tool for pre-processing a video stream to get an input frame to track but in practice, it was a non-trivial task even for a static camera. Smoothing operation was performed to reduce the noise levels but there might be a chance of losing an edge information for larger convolution kernels. A Gaussian filter with kernel size 5x5 sampling window was used to smoothen the frames without much loss of the information. Adaptive threshold with a mean neighborhood area type is performed on each frame to convert the frames into binary frames with noise reduction. Morphological operations were performed depending upon the binary frame information. With respect to demo site and the camera used, the binary image \( w_s \) eroded and dilated followed by another dilation and closing operation with a structuring element of rectangular shape. The transformed binary image was analyzed to find the boundaries of the vehicle, from which the center of mass of the vehicle is calculated.
Chapter 4. Results and Discussion

The positioning system has an accuracy of 89.75% with a deviation of 3 pixels. The performance of the system depends on the location of the camera. In this experimental work, the camera was placed at a mid-high level (approx 1.8 m) to examine the performance of the system.

Morphological operations were key factors for estimating the accurate location. Binary video frames was analyzed and the above mentioned morphological operations are performed. In a real-time scenario, the operations should produce similar output for each frame, upon which proper positional coordinates can be obtained i.e, the coordinates will not deviate away from the line of the path.

4.4.2 Gradient Positioning System

The input signal of 1920x1080 resolution is resized to 480x240 resolution for visual convenience. OpenCV library provides only three types of operators to calculate the gradient vectors. The gradient along the horizontal and vertical axis are calculated and then the magnitude vectors for every pixel is determined. Median filter with a 3x3 kernel size is used to smoothen each video frame and the obtained filtered frame is converted to a binary image with an adaptive threshold value. The binary image mostly contains the edge information, so dilation operation followed by a closing operation is performed to transform each frame such that there is some curve to join all the continuous points with the same intensity. Mean of all the obtained x-coordinate and mean of all the obtained y-coordinates values of pixels in the curve resulted in the position of the vehicle.

This positioning system has an accuracy of 92.26%. The accuracy of the system is greater than the center of mass positioning system while the factors such as camera location, camera view are same. The estimated coordinates are in the line of the vehicle path.

The performance of both systems approximately coincides when the camera view of the vehicle changes from the top view to the side view. As the height of the camera was increased, the vehicle is projected as the top view. Top view of the vehicle has rectangular shape and better accuracy than side view of the vehicle which is a combination of rectangular and circular shape because the systems consider rectangular shape for tracing the vehicle.
4.4.3 Challenges

Many challenges appeared during the period of experimental work. The major challenges were:

1. What is the methodology to be employed?
2. Where should the camera be placed whether on the vehicle or on the site?
3. What is the view of the camera for processing the images?
4. What is the height of the camera above the ground?
5. What type of morphological operations need to be performed for object tracking?
6. How to measure the accuracy of the system?
Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, position of a single vehicle is estimated by proposing and implementing two different algorithms i.e, gradient based positioning system and center of mass based positioning system. The first algorithm gradient positioning system has achieved the positional accuracy of 92.26%. The second algorithm center of mass positioning system achieved the positional accuracy of 89.75%.

The experimental work is carried out in a real time scenario with a camera of 1920x1080 resolution placed at moderate height (approx 1.8 m) with respect to the ground to test the performance of the positioning system on a specially designed Volvo CE demo site. The percentage change in the accuracy of two positioning systems is low and the gradient positioning system had a delay of one frame and center of mass positioning system had a delay of 3 frames where the frame rate is 25 frames per second.

The center of mass and gradient positioning systems process the frame in parallel with the real time video. Morphological operations are key to the accurate position estimation of the vehicle in center of mass positioning system. Both the positioning system were found to be reliable for navigating the vehicle along the desired path without human interference.

5.2 Future Works

Many different adaptations, tests, and experiments have been left for the future due to lack of time. The following ideas could be tested:

1. To check the reliability of the center of mass and gradient positioning systems for multiple vehicles position estimation.

2. Generating a more robust background subtraction method.

3. Tracking the position of the vehicle in 3 dimensional coordinates.
4. It could be interesting to consider the shadow effects for estimating the position of the single vehicle or multiple vehicles.

5. Defining a standard morphological operation procedure suitable for estimating the location in any environment.

6. Sensitivity towards the camera views and height of the camera needs to be improved.

7. Finding a methodology to integrate both proposed algorithms into a single algorithm which has more accuracy than the two proposed algorithms.

8. Implementing gradient positioning system for non-static background conditions.

9. Filtering techniques such as Particle and Kalman filters are applied in real-time scenario without much delay in processing.

10. It could be a really fascinating work when the center of mass and gradient based positioning systems generate a good accuracy for navigating the vehicle in unknown environment by simultaneous localization and mapping.
References


REFERENCES


