UTILIZING THE INFRASTRUCTURE FOR GROUND TRUTH BASED LOCALIZATION

A Thesis

by

TYLER STEPHEN MARR

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

Co-Chairs of Committee,	Swaminathan Gopalswamy
	Swaroop Darbha
Committee Members,	Dezhen Song
Head of Department,	Andreas A. Polycarpou

May 2019

Major Subject: Mechanical Engineering

ABSTRACT

This paper seeks to implement new approaches for autonomous vehicle localization through utilizing the infrastructure. In both approaches a monocular vision system is used. An infrastructure enabled autonomy approach is taken wherein a camera mounted at 10 meters on roadside units is used along with vision processing to detect and localize vehicles on the road. After, a new technology developed by 3M, Smart Codes, is utilized by using an infrared camera mounted on a vehicle with Smart Code signs along the roadside to provide localization estimates. Filtering algorithms were developed for each specific localization method and experiments were performed to test the accuracy of all these localization approaches. It was found that the infrastructure enabled autonomy approach was able to estimate the vehicle's position to within a meter over a range of about 50 meters. The 3M Smart Code approach was also able to achieve accuracy to within a meter, but over a range of approximately 20 meters. The Smart Code approach performed more consistently and saw less variation in the accuracy of the position estimates. Both approaches saw improvements through the incorporation of filtering algorithms that were able to give better, more consistent position estimations.

DEDICATION

This work is dedicated to my wife who has helped encourage and support me throughout all of my studies.

ACKNOWLEDGMENTS

First I would like to thank Dr. Gopalswamy for all his aid, ideas, and consistent feedback for the last two years on all of my research. Additionally I would like to thank Dr. Rathinam and Dr. Darbha for their various inputs over the last two years.

Next I would like to thank all my fellow students who I have worked with and who have helped me with all my research. Specifically I would like to thank Kenny Chour and Abhishek Nayak who I have worked most closely with.

Finally I would like to thank the 3M Connected Roads team who gave me consistent constructive feedback every week on my work with them.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis dissertation committee consisting of Professor Swaminathan Gopalswamy as chair, Professor Swaroop Darbha as co-chair, and Head of Department Dr. Andreas A Polycarpou of the Department of Mechanical Engineering and Professor Dezhen Song of the Department of Computer Science.

Funding Sources

Graduate study was supported in part by a fellowship from the Dwight D. Eisenhower Transportation Fellowship Program and by the 3M Connected Roads Group.

NOMENCLATURE

IEA	Infrastructure Enabled Autonomy
MSSP	Multi-Sensor Smart Pack
RSU	Road Side Unit
GPS	Global Positioning System
IMU	Inertial Measurement Unit

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

1. INTRODUCTION

1.1 Need

Autonomous vehicle research is increasing in popularity across the globe and the eventual incorporation of some of these technologies into the real world is bound to happen. Along with these autonomous vehicles is a simultaneously developing smarter infrastructure to assist both vehicles on the road and other third parties. A smarter infrastructure provides unique opportunities to autonomous vehicle by increasing the amount of information these vehicles can know. It is key for both the development of both autonomous vehicles and the smart infrastructure that the interaction between these two entities is heavily investigated.

1.2 Related Research

In most standard vision based localization solutions two cameras, stereo vision, are used and object position determination is done through triangulation. However, a single camera, monocular vision, approach is also feasible in many instances. One such approach is known as visual odometry. In a visual odometry approach the 3D environment is mapped out and the location of various objects known so that from a single camera frame, from a vehicle mounted camera, a localization estimate can be made [1]. Another approach of using monocular vision can be from an external source localizing a vehicle such as using a fixed wing UAV to track objects on the ground [2]. The most similar approach to the one discussed in this paper uses monocular vision approach with a camera mounted near ground level [3]. In this approach a computationally intensive process of foreground extraction, edge and line detection, and fine-grained vehicle model classification was used to accurately localization the position and orientation of vehicles in a small area. However, this required a well defined environment and was limited in effective range.

At Texas AM the idea of infrastructure enabled autonomy is currently being developed by professors and students [4]. This idea seeks to redefine the current approach to autonomous vehicles. Additionally, at 3M a new idea has been introduced to allow computers to more easily read signs

and gather more information about their environment through the use of Smart Codes.

1.3 Infrastructure Enabled Autonomy Concept

In most current attempts at creating an autonomous vehicle today, the sensors are all mounted on a vehicle which determine situational awareness for the vehicle. This situational awareness is then used to make a decision based on the vehicle's goal. Once the decision is made, the vehicle responds and provides the appropriate steering and acceleration or braking. In a fully autonomous, level 4 or 5, vehicle, this process puts all of the responsibility on the manufacturer of the car. The concept of infrastructure enabled autonomy (IEA) aims to distribute this workload to multiple parties. This will be accomplished by putting the responsibility of situational awareness on the infrastructure, the responsibility of decision making on a third party, and the responsibility of enabling drive-by-wire on the automotive manufacturer [4].

The first step of the autonomous process is to provide accurate situational awareness. In IEA, the primary source of situational awareness is provided by a multi-sensor smart pack (MSSP) mounted on road side unit (RSU). The MSSP is a collection of sensors such as cameras, LIDAR, RADAR, and other sensors that can be found on the autonomous vehicles of today. Currently, the only sensor being used on the MSSPs is a single standard camera. The RSUs need to supply two primary functions. First, they need to perform as a mount for the MSSPs at a sufficient height. Ideally, the MSSPs are mounted as high as possible while still able to provide accurate localization of vehicles. Currently, the MSSPs are at a height of approximately 10 meters. Second, the RSUs need to supply sufficient power and provide a safe structure for the computing and communication components.

Once the situational awareness has been developed by the MSSPs on the RSUs, the information must be communicated to the vehicle. DSRC units are used to share localization information to the vehicle, which can then augment the data using on board sensors such as a GPS or IMU. A controller is then used on board the vehicle which uses the localization information in conjunction with a desired path to provide the desired vehicle inputs. A small scale version of this setup has shown successful localization, path planning, and control across two MSSPs in an indoor

environment using an RC car [5].

1.4 3M Smart Code

Currently, in order for vehicles to read road signs they must detect a sign in their environment, determine what the sign is conveying to the humans, and determine any other relevant information that may help in relating the meaning of the sign. 3M has developed a solution to improve the communication between road signs and computers through the implementation of Smart Codes. Smart Codes are formed in an 11x11 checkerboard pattern that convey an encoded hex message to a vehicle through proprietary software. This hex string message is then linked to a look-up table with any relevant information about the sign including its message and its location. These codes can be printed on road signs and be virtually invisible to the unaided eye, but fully illuminated to an infrared camera synced with an infrared strobe. This allows for implementation to the real world without impeding on humans and allowing for computers to become more incorporated to vehicles. Simultaneous to the encoded information is also inherent information about the sign and Smart Code itself. Since these codes are developed and printed to certain standards, by looking at the geometry of the sign from the camera readings, other information, such as distance to the sign, can be determined from the stream of data.

1.5 Research Question

How can the unique opportunities presented by the infrastructure be leveraged to improve the localization of autonomous vehicles?

1.6 Overview of Paper

First, the localization approach using monocular vision in infrastructure enabled autonomy will be discussed. From here, the localization approach using the 3M Smart Code technology will be investigated. Once both localization approaches have been developed and evaluated, the filter algorithms for these different approaches will be discussed. Following the discussion of filtering will be conclusions regarding these localization and filtering approaches. Finally, this paper will conclude with future work for all of these topics.

2. LOCALIZATION BY INFRASTRUCTURE ENABLED AUTONOMY

2.1 Infrastructure Enabled Autonomy Background

Infrastructure Enabled Autonomy (IEA) is being developed as a unique alternative approach for implementation of autonomous vehicles into the market where the responsibilities of autonomous driving are split between different groups. A visual depiction of the IEA concept can be seen in Figure 2.1. The role of the infrastructure will be providing the situational awareness to all relevant parties. This situational awareness will be developed through sensors contained in Multi-Sensor Smart Packs (MSSPs) mounted on Roadside Units (RSUs). The IEA concept is not restrictive to number of sensors used and sensor type, however in early conceptual stages each MSSP is only equipped with a single vision camera. The most important component of this situational awareness is the localization information of all vehicles and objects on the road. The goal of this section is to develop and examine a localization approach for IEA using monocular vision from atop RSUs approximately 10 meters above ground level.

2.2 Localization Approach

Traditional monocular vision localization approaches rely on intrinsic and extrinsic matrix calibration of a particular camera and lens setup. Through this approach a plane with a large checkerboard pattern can be transformed to be a two dimensional plane in world coordinates and give accurate localization results. This approach is best applied to localizing objects with a small height as compared to the height of the camera because the line of sight might not match with the true position of the vehicle. A visualization of this can be seen in Figure 2.2. This figure shows how the true position of the vehicle would always be closer to the MSSP than the estimated position. A different approach is required to compensate for this issued. The approach here will be to calibrate each camera with real position data of the vehicle and use this calibration data as a reference to estimate the true position of the vehicle.

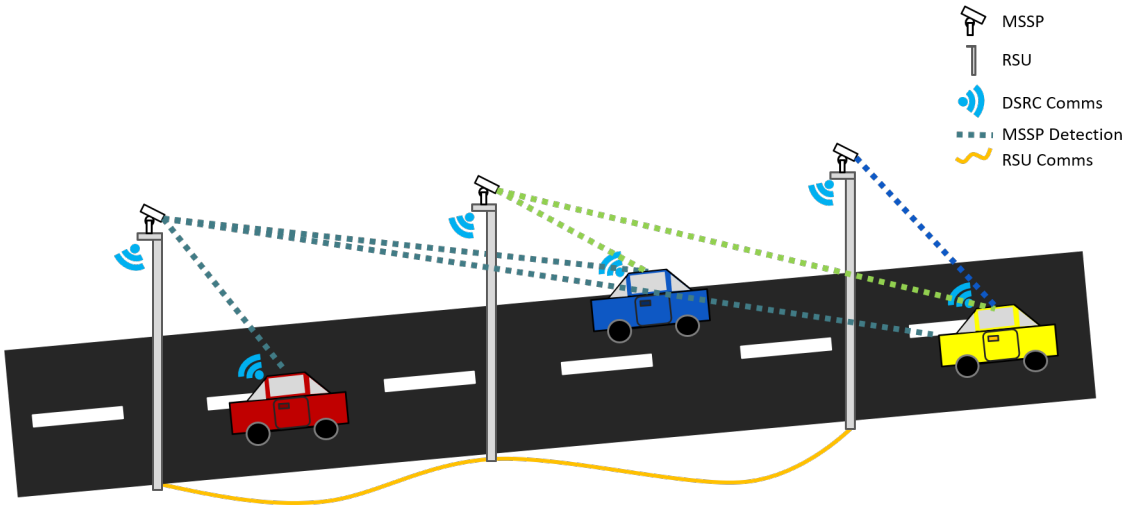

Figure 2.1: Visualization of the IEA concept

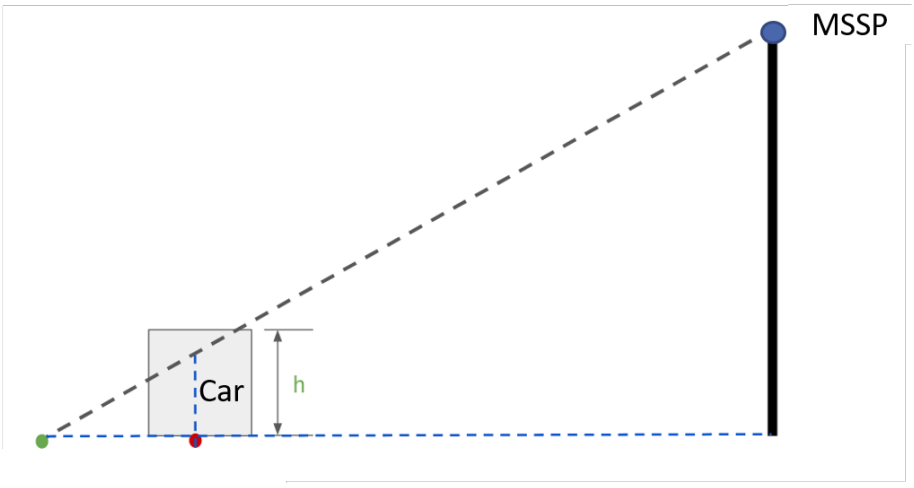

Figure 2.2: Representation of IEA setup and vehicle detection

Figure 2.3: Bounding box around vehicle as detected by an MSSP

2.3 Camera Calibration

Through computer vision processing a bounding box can be generated around a detected vehicle in the frame of the camera. An example of a detected vehicle can be seen in Figure 2.3. This bounding box has exactly known pixel coordinates and, assuming the camera is relatively stationary, it is safe to assume that the same vehicle in the same position on the road at a future time will cause a bounding box to be generated with very similar pixel coordinates. Using a differential GPS placed on the vehicle, the position of the vehicle can be known precisely to within a few centimeters of accuracy. This means we can record the position of the vehicle in both its pixel and world coordinates. Now, without a differential GPS, if the vehicle is driven to be in the exact same position, its world coordinates can be known simply from looking at its pixel coordinates. This idea can then be further expanded if the vehicle is driven throughout the entirety of the detectable region of the camera frame while simultaneously recording its world coordinates from a differential GPS. Using this method a one-to-one map can be generated between the pixel coordinates of the vehicle and its world coordinates. An example of these calibration points can be seen in the pixel frame in Figure 2.4 and in the world frame in Figure 2.5.

From these mapped coordinates a position estimate can be made given any pixel coordinates through a multivariate interpolation. This position estimate will vary in accuracy depending on the

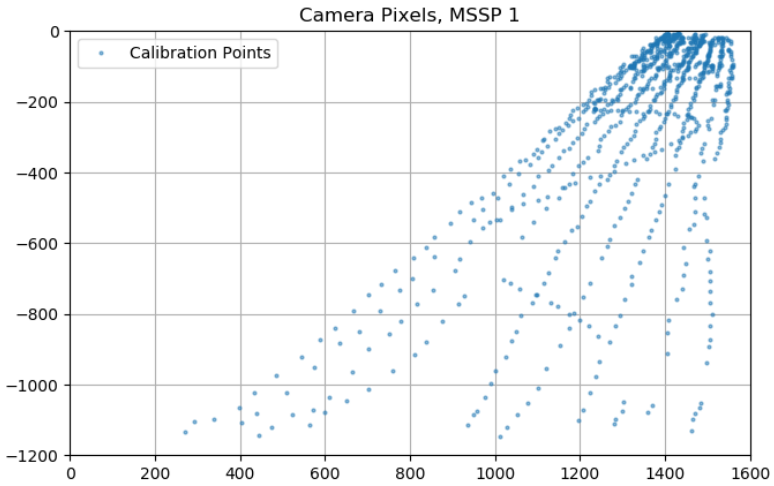

Figure 2.4: Calibration points in pixel coordinates

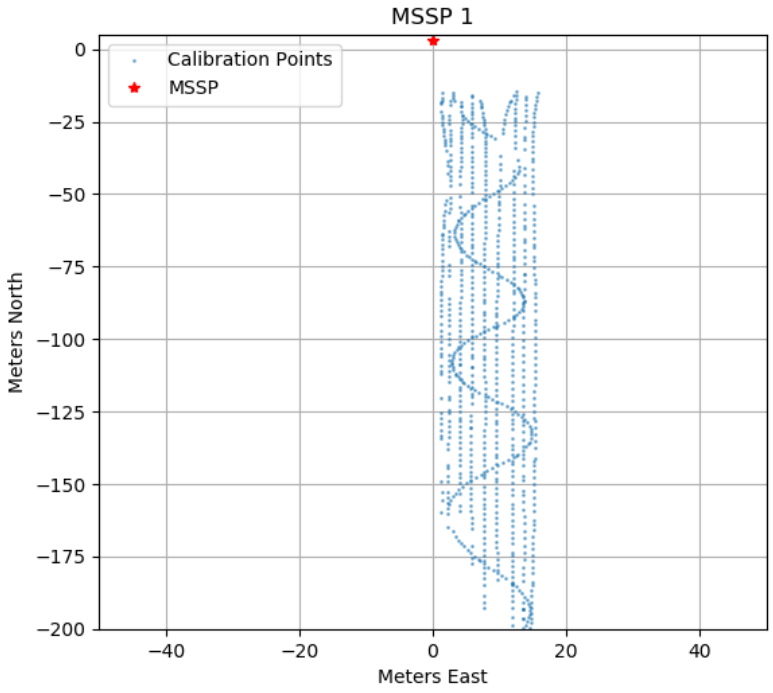

Figure 2.5: Calibration points in world coordinates

repeatability of the data collected, the overall accuracy of the measurements and the density of the calibration points.

2.4 Position Estimation

In order to estimate a point in between calibration points, an interpolation method is required. However this is not a simple 2 dimensional interpolation because the points are distributed highly irregularly and there is not a simple transformation between the two coordinate systems. A multivariate interpolation method known as inverse distance weighting is applied and used to estimate the position of the vehicle from any given input pixel coordinates. For inverse distance weighting, there are a given set of n input points (pixel coordinates), (x_i, y_i), and n output points (world coordinates), $v(\mathbf{x_i}) = f(x_i, y_i)$. Any given input, $\mathbf{x}$, is given a predicted output, $f(\mathbf{x})$ through Equation 2.1.

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) v(\mathbf{x_i}) \tag{2.1}$$

The weight function, $w_i(\mathbf{x})$, is a function of distance between the point being estimated and the i^{th} calibration point which is calculated by Equation 2.2.

$$d(\mathbf{x}, \mathbf{x_i}) = ||\mathbf{x} - \mathbf{x_i}|| \tag{2.2}$$

The weight of each calibration point is then directly related to distance calculated through the Equation 2.3.

$$w_i'(\mathbf{x}) = \frac{1}{d(\mathbf{x}, \mathbf{x_i})^p} \tag{2.3}$$

In this equation, $p > 1$ determines the rate of decay as distance from the calibration point increases. Therefore the closer calibration points are weighted more heavily when estimating the position of vehicle. A very high p value would lead to ignoring all but the nearest points and a very low p value would lead to using most calibration points for each estimate. The weights are then normalized so

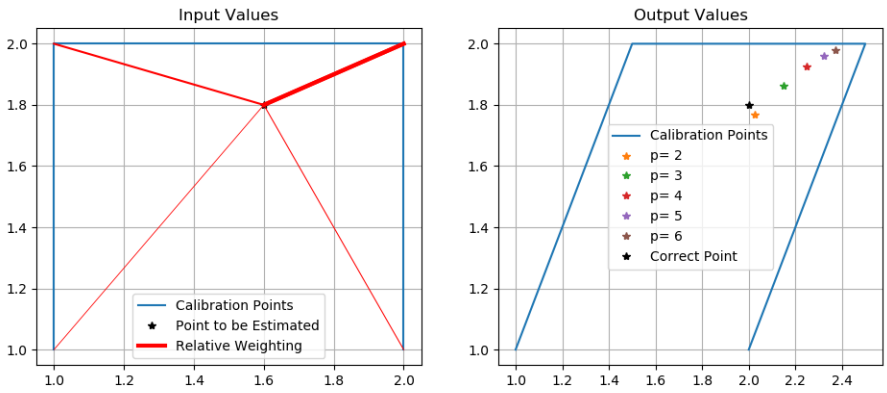

Figure 2.6: Simple estimation made by inverse distance weighting

that they sum to a value of 1 through Equation 2.4.

$$w_i(\mathbf{x}) = \frac{w_i'(\mathbf{x})}{\sum_{k=1}^{n} w_k'(\mathbf{x})} \tag{2.4}$$

After all the weights are calculated, Equation 2.1 is applied and a position estimate is generated. A simple example of the application of inverse distance weighting being applied with varying p values can be seen in 2.6. The four corners of the blue quadrilateral in each frame represent the calibration points and the black marker represents the point to be estimated and the correct output point. The thickness of the red lines represent the relative weight of each attached calibration point with a p value of 2. As can be seen, the nearest corner has a much higher weight than the furthest point. It can also be observed that as p increases, the estimated point approaches the nearest calibration point. This is because the weights become increasingly disproportionate as p increases. In this example a p value of 2 clearly leads to the most accurate prediction, however this is not necessarily true of more complex calibration data sets and more complex coordinate transformations.

2.5 Experimental Approach

In order to test this localization approach data was collected in an IEA setup with two RSU mounted MSSPs along a 200 meter stretch of road. Each MSSP contained a monocular vision

Figure 2.7: Experiment Setup

camera attached to a computer to perform vision processing. Communication between each MSSP computer and the vehicle was accomplished through DSRC units. A high accuracy differential GPS was used for position calibration as well as for a ground truth reference to compare with the localization estimates. The vehicle was driven through simple lane changes in a southward direction. An image of the setup of the experiment with the testing vehicle in the foreground and an RSU with mounted MSSP in the background can be seen in Figure 2.7.

2.6 Results and Discussion

Figure 2.8 shows the localization estimates from the first of the MSSPs. The red dots represent each position estimate, the black path is the true position of the vehicle from the differential GPS, and the blue marker is the approximate position of the MSSP.

From looking at the data in Figure 2.8, the vehicle appears to localize the vehicle accurately up to almost 100 meters away from the MSSP. After the vehicle is greater than 100 meters away it becomes much less accurate. This conclusion is further supported when looking at the position estimation error in Figure 2.9. In this plot the euclidean distance between the estimated position and true position is plotted as a function on meters south from the MSSP camera. Most of the data points predict the true position within one meter up until approximately 50 meters away where higher variance starts to be observed. Up until almost 100 meters away the many position estimates continue to be within 1 meter of error, with more less accurate predictions occurring. After 100 meters away the position estimates degrade significantly with the some being greater than 7 meters

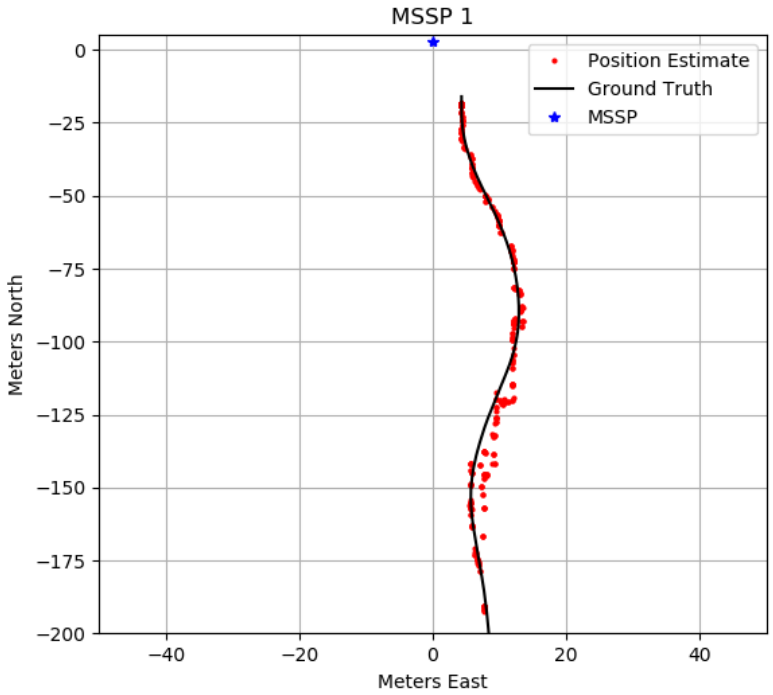

Figure 2.8: Position estimates from IEA

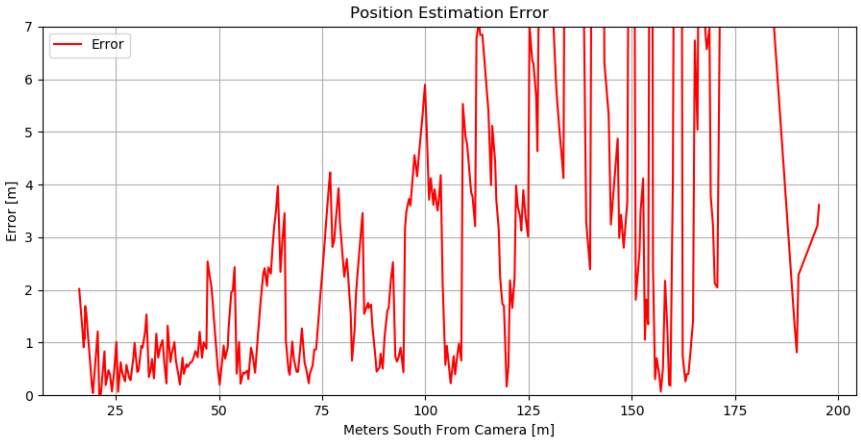

Figure 2.9: Euclidean error in position estimates from IEA

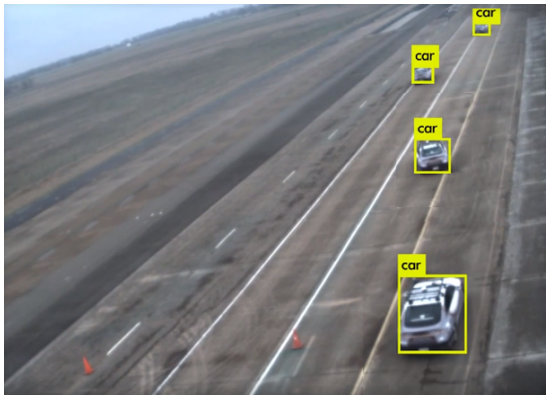

Figure 2.10: View from camera of the vehicle at different times

away in error. This plot reflects the best results achieved by varying the p factor from the inverse distance weighting interpolation method to be $p = 5$.

In general it is clear that as the distance away from the camera increases, the position estimate accuracy is reduced. Figure 2.10 displays the camera's view of the vehicle at multiple times throughout the experiment. As the vehicle begins to get very far away from the camera the bounding box size is significantly reduced and every difference in pixel is the equivalent of a large displacement in the world frame. Additionally, as the vehicle becomes smaller it is more difficult for the software to accurately create a bounding box around it, adding to the estimation error. Another factor to take into account is the density of the calibration points. It was found that by reducing the overall number of calibration points the position estimates became less accurate. In Figure 2.11 the plotted calibration points overlayed with a plot of the center of the bounding box of the vehicle can be seen. This plot shows that where the calibration points are most dense is where the position estimates are least accurate. In addition to the previous mentioned reasons for degraded performance at the edge of the frame the high density cluster leads to an increased chance of the bounding box center landing exactly on a calibration point. In this instance the estimated point will fall in the exact same point, but this might be less accurate for the previously mentioned reasons. Overall, at the edge of the frame there is too much variability for high accuracy.

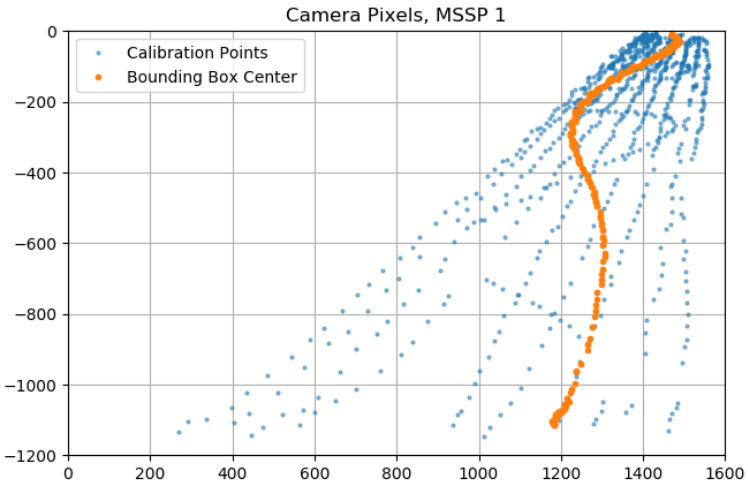

Figure 2.11: Bounding Box position in the camera frame

3. LOCALIZATION BY SMART CODE

3.1 Smart Code Functionality

Smart Codes were developed to convey information to computers using an 11x11 checkerboard pattern of white and black squares. An example of this layout can be seen in Figure 3.1. In order to display this information without disrupting the visual nature of the signs to humans the Smart Code is printed on retroreflective road signs underneath the paint so that infrared light will reflect back in a Smart Code pattern only visible to the camera. An example of the typical Smart Code on a sign see in the visible spectrum as compared to the infrared spectrum can be seen in Figure 3.2.

The primary function of Smart Codes is to provide encoded information in their checkerboard patterns. In order to provide this information, the signs need to follow strict guidelines in terms of patterns and shape. Adhering to these guidelines inherently provides for more information that can be leveraged beyond just the encoded information. Once such instance of this is providing a distance estimate from the sign. Figure 3.3 displays the general principal for how this distance can be estimated.

Essentially, there are known parameters of the camera and of the smart code dimensions, all lengths marked by a green line. By comparing the area of pixels in the quadrilateral that the smart code occupies with the known area of the smart code, the distance can be obtained. While this approach is very reliable, it has the strict condition that the object being detected is perfectly normal to the camera. In a realistic road environment a car will be driving on the road and reading

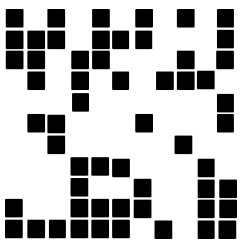

Figure 3.1: An example of a standard Smart Code layout

Figure 3.2: Sign as seen in visible spectrum, left, and infrared light, right

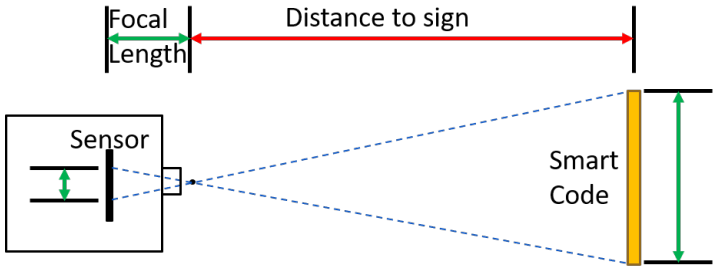

Figure 3.3: Standard distance estimation using Smart Code

a sign perpendicular to the road a few meters off to the side. If the distance is to be estimated from the sign in this instance, then the sign cannot be assumed to be normal to the camera in most scenarios. The goal, therefore, is to use as much information about the sign and camera setup that can be known to estimate the location of the camera relative to the sign.

3.2 Localization Approach

Each Smart Code has an associated hex code that links to a .json file which then contains any stored information about the sign. This means that the exact location and setup of the sign can be included in this file. In order to fully localize the camera relative to this sign, the relative displacement to the sign is the only remaining factor that needs to be calculated. While the Smart Codes are no longer perpendicular to the camera, there is a relationship between the height and width of the Smart Code as seen by the camera and the entrance angle between the Smart Code

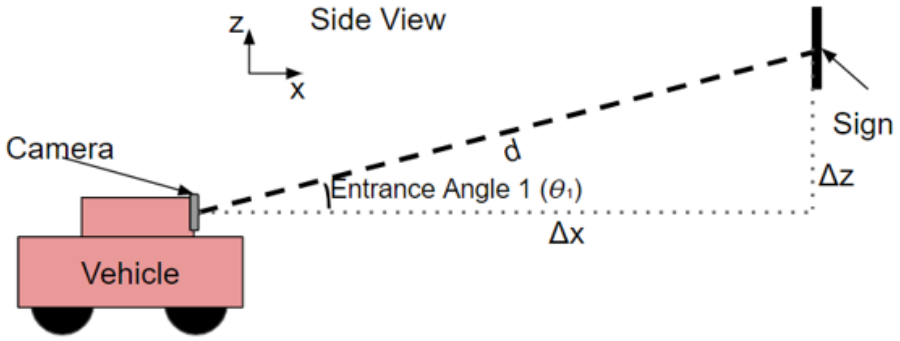

Figure 3.4: Side view of typical setup with car approaching Smart Code

and camera sensor. Using this relationship the relative displacement between the Smart Code sign and the camera can be calculated in three steps, using the pixel height of the sign, using the pixel width of the sign, and finally combining these results.

3.2.1 Smart Code Localization: Step 1

Figure 3.4 shows the side view of the standard setup of a car driving along the road towards a sign. Note, in this figure the car is not at the same depth, XZ plane, as the sign. Here, the true distance between the camera and the sign is represented by d, the longitudinal displacement is represented by Δx, the vertical displacement is represented by Δz, and the vertical entrance angle between the sign and camera is represented by θ_1.

Using the pixel height of the Smart Code, a distance estimate can be made in the same manner as the one represented in Figure 3.3. However this distance estimate will overestimate the true distance d because $\theta_1 \neq 0$, leading to the sign appearing smaller in the frame than it is in reality. Let this distance estimate be known as d_{m1}. Additionally, the vertical displacement, Δz, is known because the sign has encoded its exact position and the camera is mounted at a known point on the vehicle. Figure 3.5 shows the known values in green and the values to be calculated in red. From these givens, the entrance angle, θ_1, and distance, d, can be calculated using the following

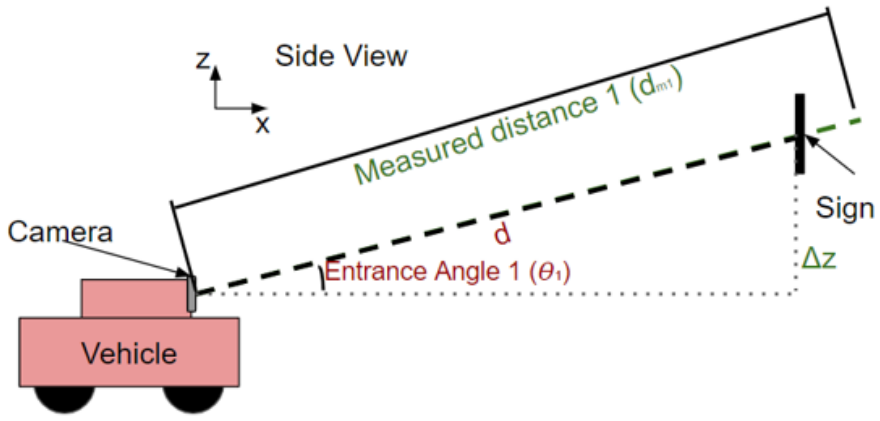

Figure 3.5: Side view car approaching Smart Code with variables

equations.

$$\theta_1 = \arcsin \frac{\Delta z}{d}$$
$$d_{m1} = \frac{d}{\cos \theta_1} \tag{3.1}$$

Here there are two equations accompanied with two unknowns, therefore these equations can be combined to solve for distance in the following manner.

$$d = \frac{\sqrt{2d_{m1}}\sqrt{\sqrt{d_{m1}^2 - 4\Delta z^2} + d_{m1}}}{2} \tag{3.2}$$

3.2.2 Smart Code Localization: Step 2

The next step is to use the pixel width reading of the Smart Code in conjunction with the distance value calculated in step 1. A top view of the standard setup with the vehicle on the road can be seen in Figure 3.6. Similar to step 1, from looking at the pixel width of the Smart Code a distance estimate can be made assuming the sign is square with the camera. Again this distance estimate will be an overestimate of the true distance, however it can still be used to assist in the localization of the vehicle. Figure 3.7 details the known and to be calculated variables in green and red, respectively. From these variables, the entrance angle, θ_2, can be calculated from the following

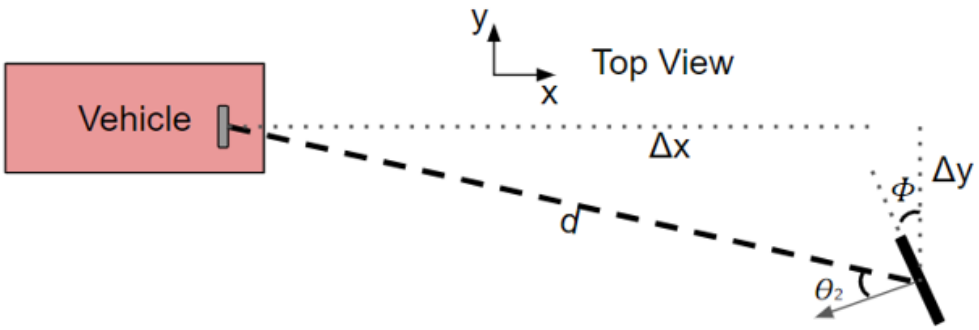

Figure 3.6: Top view of typical setup with car approaching Smart Code

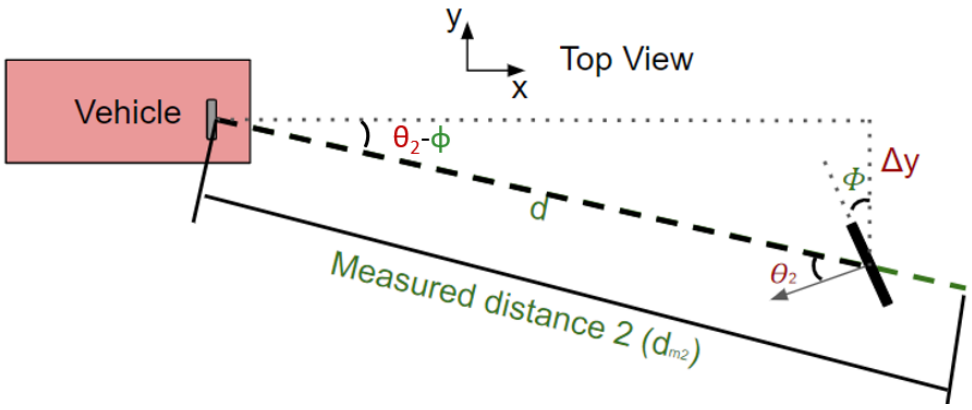

Figure 3.7: Top view car approaching Smart Code with variables

Figure 3.8: Experiment setup with labels

equation.

$$\theta_2 = \arccos \frac{d}{d_{m2}} + \phi \tag{3.3}$$

From this, the horizontal offset can then be determined in the following manner.

$$\Delta y = d \sin (\theta_2 - \phi) \tag{3.4}$$

3.2.3 Smart Code Localization: Step 3

Finally, once the distance, d, the z offset, Δz, and y offset, Δy, are known the x offset can be determined through the following equation.

$$\Delta x = \sqrt{d^2 - \Delta y^2 - \Delta z^2} \tag{3.5}$$

3.3 Experimental Setup

In order to test the accuracy of the localization algorithm, an experiment was setup to mimic a real world environment. Figure 3.8 shows the full setup of the experiment. Tests were performed by driving the car in 4 different paths past the three signs. The left lane, middle lane, and right lane as pictured, followed by tests where the vehicle was driving directly next to the signs. Additionally, the camera was mounted at three different angles, approximately 0°, 30°, and 45°. The three signs used can be seen in Figure 3.9. There was one large sign with a smart code measuring 22"x22"

Figure 3.9: Signs used for experiment

Figure 3.10: Camera apparatus mounted on vehicle used for experiment

and two smaller signs with smart codes of 5"x5".

A 2448x2048 infrared camera was set to capture at 30 frames per second and tests were performed at approximately 10 mph, 4.5m/s. The camera apparatus used for the experiment can be seen in Figure 3.10. Pictured and labeled here are the infrared camera, the infrared strobes, and a differential GPS which was used to give the ground truth location of the camera during the testing. This GPS antenna was also used to measure the precise location of the signs so that all data was in the same reference frame.

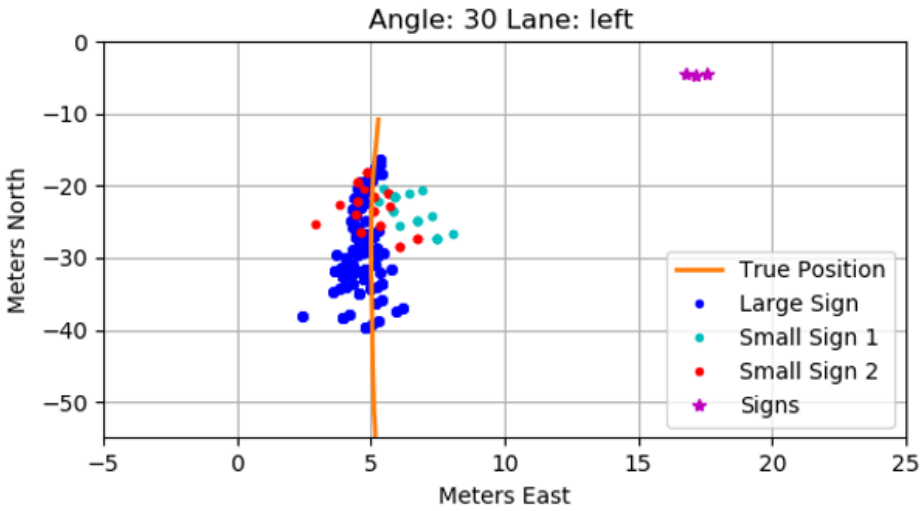

Figure 3.11: Localization results for the left lane with camera mounted at 30°

3.4 Results and Discussion

The most accurate results with the largest read range were found in the tests performed in the left lane at a camera mounting angle of 30°. The localization prediction data for this setup can be seen in Figure 3.11. The x and y axes represent Meters East and Meters North, respectively, from the location of the base station used for the differential GPS. The location of the three signs used for testing can be seen in the magenta points in the upper right corner of the figure. The orange vertical line represents the true path taken by the vehicle. Note that this true path taken begins before the sign was detected and ends after the sign stops being detected. All of the other many points on the figure each represent a position estimate of the algorithm. The blue dots are position estimates from reading the large sign, the cyan dots are from the right-most small sign, and the red dots are from the left-most small sign. From observation it is clear that there are far more estimates from the large sign over a greater distance, indicating that it had a greater read range than the small signs. Additionally, the large sign appears to have much more accurate position estimates in general, particularly where the readable range of the three signs all overlap. These conclusions are further supported when investigating the euclidean error in each position estimate, which can be found in Figure 3.12.

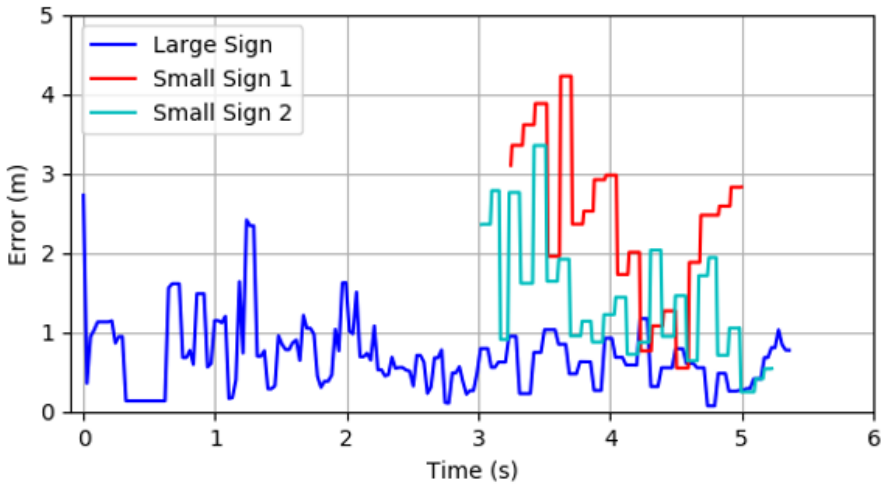

Figure 3.12: Localization errors for the left lane with camera mounted at 30°

From looking at Figure 3.12 it can be seen that there is much lower error for almost every read when comparing the large sign to the small signs. There is also a trend of reduction in error as the vehicle moves closer to the signs with the greatest error from the large sign occurring with the first reading. There was several seconds of readings that were within 1 meter accuracy for the large sign, a span of approximately 15 meters. This accuracy indicates that the large signs can reliably be used to localize a vehicle. However, when trying to increase the range on the readings the results become less accurate. This is further exemplified in Figure 3.13, which shows position estimates made from when the camera is mounted looking straight ahead, 0°, and the car is driven directly next to the signs.

From looking at this figure it is clear that the readable range has increased, however the accuracy at these great distances has significantly reduced. There are multiple distinct lines that appear in these position estimates that become very clear when the estimates are far away from the signs. The reasoning behind this is the resolution of the image being captured. The localization algorithm relies on a difference in pixel width and pixel height of the detected Smart Code to determine the displacement between the sign and the camera. In the case of the vertical line, the camera is reading that the pixel width is equivalent to the pixel height. The next closest diagonal line represents when the pixel width becomes one less than the pixel height, meaning that at a great distance the

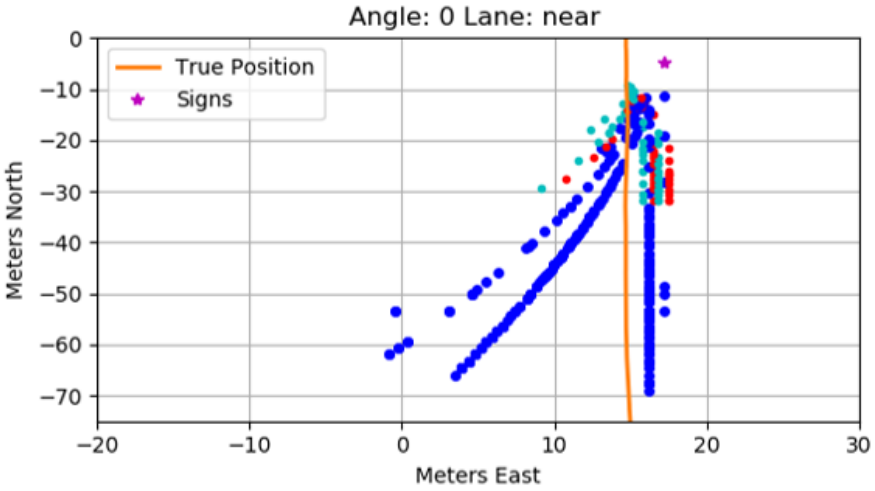

Figure 3.13: Localization results for the near lane with camera mounted at 0°

algorithm does not have high resolution. This is a limitation of the camera, and a higher resolution camera would give better results. Despite this poor localization at great distances, the overall euclidean distance estimate between the camera and the Smart Code was very accurate. Figure 3.14 shows the error in euclidean distance estimate when compared to the euclidean distance measured by the differential GPS.

Figure 3.14 shows that even at a great distance the distance estimate would often be within 1 meter of the true distance. As the vehicle got closer to the sign the distance estimate errors became more consistent with a bias of just under 1 meter of error. This means that distance estimates from two independent signs in different locations simultaneously could be used triangulate the position of the camera even at great distances from the signs.

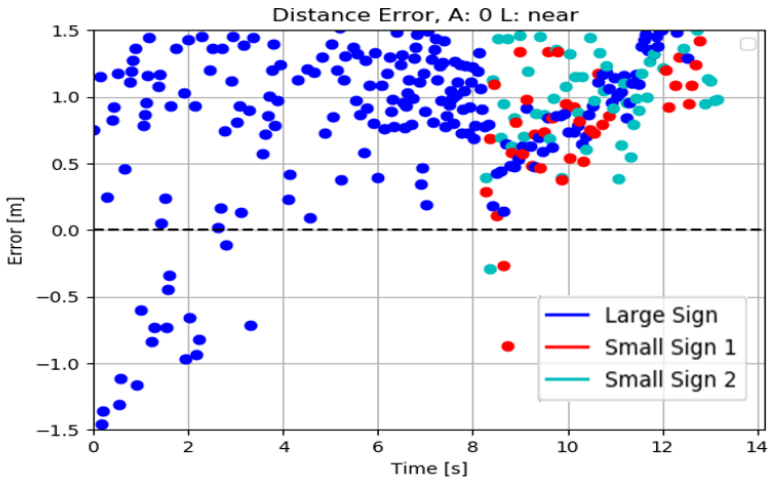

Figure 3.14: Error in distance estimates for the near lane with camera mounted at 0°

4. POSITION FILTERING

4.1 Goal

Both of these localization approaches rely primarily on visual data which is noisy, comes at infrequent rates, and has vary accuracies depending on the current resolution. This presents a filtering problem that must be able to handle a ever changing landscape. The filters are considered successful when it is able to track the position estimate within an acceptable margin of error and behaves smoothly.

4.2 IEA Position Filter

4.2.1 Background

Infrastructure enabled autonomy uses a variety of sensors and sources of information that can be used for vehicle localization. The intent is for the majority of the vehicle sensing to be performed by the MSSP units mounted on the side of the road. Therefore, the goal is to minimize the use of sensors on the vehicle to what is readily available in a typical vehicle on the road today. This includes a speed estimate, heading measurement, and a low frequency GPS.

4.2.2 Filter Design

The basic position estimate can be found using the previous position and adding the velocity in that direction multiplied by the time since the last measurement. This is presented in Equation 4.1.

$$\begin{aligned} \hat{x}(k+1) &= \hat{x}(k) + v_x * \Delta T \\ \hat{y}(k+1) &= \hat{y}(k) + v_y * \Delta T \end{aligned} \tag{4.1}$$

Only the overall speed of the vehicle is known and a prediction requires knowledge of the velocity in the x and y directions. This can be calculated using Equation 4.2, where θ represents the heading

of the car relative to the x-axis.

$$\begin{aligned} v_x &= v * cos(\hat{\theta}) \\ v_y &= v * sin(\hat{\theta}) \end{aligned} \tag{4.2}$$

A correction term can be applied to Equation 4.1 in order to increase it's accuracy since it is only an estimation. The resulting equation is as follows:

$$\begin{aligned} \hat{x}(k+1) &= \hat{x}(k) + v_x * \Delta T - \lambda_1(x_m(k+1) - \hat{x}(k)) \\ \hat{y}(k+1) &= \hat{y}(k) + v_y * \Delta T - \lambda_1(y_m(k+1) - \hat{y}(k)) \end{aligned} \tag{4.3}$$

In Equation 4.3 λ_1 acts as a correction gain and $x_m(k+1)$ and $y_m(k+1)$ represent the most recent x and y position measurements. The measured position in Equation 4.3 is a combination of MSSP and GPS measurements and is found using the following method.

$$\begin{aligned} x_{mssp}(k+1) &= x_{mssp}(k) + v * cos(\hat{\theta}) * \Delta T_1 \\ y_{mssp}(k+1) &= y_{mssp}(k) + v * sin(\hat{\theta}) * \Delta T_1 \\ x_{gps}(k+1) &= x_{gps}(k) + v * cos(\hat{\theta}) * \Delta T_2 \\ y_{gps}(k+1) &= y_{gps}(k) + v * sin(\hat{\theta}) * \Delta T_2 \end{aligned} \tag{4.4}$$

Where

$$\begin{aligned} \Delta T_1 &= \text{(current time) - (time of last GPS measurement)} \\ \Delta T_2 &= \text{(current time) - (time of last MSSP measurement)} \end{aligned} \tag{4.5}$$

$$x_m = \beta x_{gps} + (1 - \beta) x_{mssp}$$
$$y_m = \beta y_{gps} + (1 - \beta) y_{mssp} \tag{4.6}$$

$$\beta = \frac{e^{-\Delta T_1}}{e^{-\Delta T_1} + 20 * e^{-\Delta T_2}} \tag{4.7}$$

$$\lambda_1 = 1 - 0.005 * e^{-min(\Delta T_1, \Delta T_2)} \tag{4.8}$$

In this case, λ_1 has a minimum value of 0.995, and it will asymptotically approach 1 as time passes since the last measurement of either the GPS or MSSP. β was defined in such a way that it will favor whichever measurement was taking more recently between the GPS and MSSP with more weight favoring the MSSP.

4.2.3 Results

Figure 4.1 shows the position estimates made by the filter along with all of the raw data used for the estimates. This figure shows the true path of the vehicle in the black line, the position estimates from MSSP 1 in the blue dots, the position estimates from MSSP 2 in the orange dots, the filtered position estimate in the green line, and finally the GPS position estimates in the red line. The initial filtered position estimate starts off with a small offset from the true position by matching the GPS initial estimate, but quickly seems to move to the more accurate MSSP estimates. Overall, the path of the filtered estimates follows the path of the true position almost exactly with a small offset. The filter was also able to create a path of smooth estimates that would accurately reflect the behaviors of a vehicle. The filtered position estimates show a significant improvement over any of the sources of raw position estimates.

The error in the filtered position estimates can be closely observed by looking at Figure 4.2. This figure again shows improved performance of the filter over the raw position estimates. While the MSSP will often provide many estimates that are more accurate than the filtered estimate,

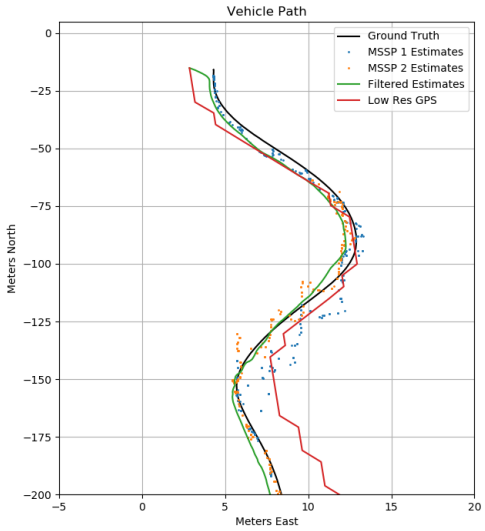

Figure 4.1: Filtered position estimates for vehicle driven across 2 MSSPs

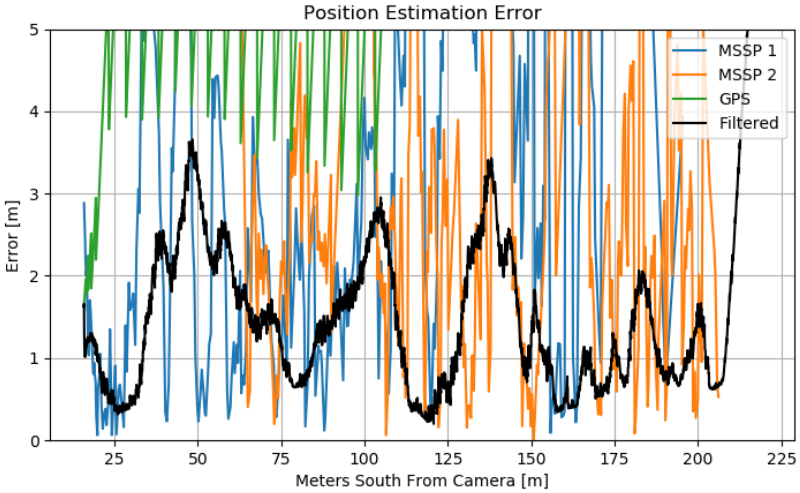

Figure 4.2: Filtered position estimate errors for vehicle driven across 2 MSSPs

in general the filter has a more consistently accurate position estimate. Additionally, the filtered position estimate trends towards convergence with the true position as can be observed by a overall decrease in variability of the error and a reduction in the peaks of the error.

4.3 Smart Code Position Filtering

4.3.1 Goal

The biggest challenges presented by the Smart Code localization setup are the low accuracy position estimates when the vehicle is far from the sign along with the difference in accuracy of the longitudinal offset versus horizontal offset of the vehicle. At great distances from the sign the longitudinal offset is fairly accurate, while the horizontal offset varies drastically. The filter developed for the Smart Code localization should take all of these factors into account and provide a solution that is able to mitigate these problems

4.3.2 Filter Design

At any given point when a position estimate is desired there is the possibility of receiving a localization estimate from the Smart Code. Whenever no new estimate is received, a simple step forward estimate is made in the same manner as Equation 4.1. However, when a new position estimate is received (x_m, y_m) the equations presented in Equation 4.9.

$$\begin{aligned} \hat{x}(k+1) &= \alpha(\hat{x}(k) + v_x * \Delta T) + (1-\alpha)x_m \\ \hat{y}(k+1) &= \beta(\hat{y}(k) + v_y * \Delta T) + (1-\beta)y_m \end{aligned} \tag{4.9}$$

In Equation 4.9 the terms α and β are both tuned independently based on their accuracy. In general, the x_m position estimate is more accurate, therefore $\alpha = 0.96$, while the y_m position estimate is less accurate, therefore $\beta = 0.99$. By tuning α and β the difference in accuracies of the two offsets is accounted for, but it does not solve the problem of low initial accuracies. To account for this issue the first position estimate is taken from the average of the first ten data points received. This will allow a prediction between the two lines of position estimates presented in Figure 3.13 and reduce the limitations of the filter estimates by avoiding a large bias offset at the

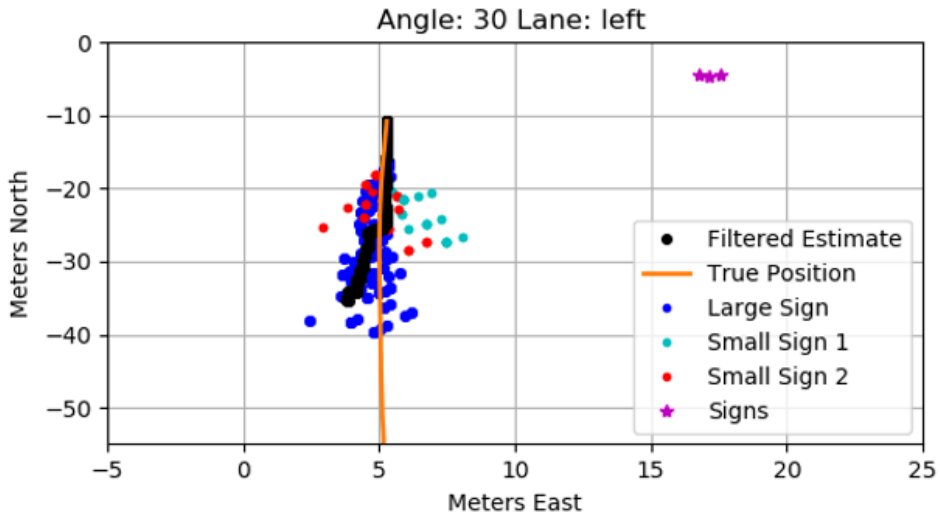

Figure 4.3: Filtered position estimates for camera mounted at 30°in the left lane

onset of the filter.

4.3.3 Results

Figure 4.3 shows the application of the previously described filter to the data obtained from previous results. From these results it can be seen that the filter tends to converge to the true position despite an initial offset to the west. Additionally, it can be seen that the filter continues to converge to the true position even after it stops receiving data due to the state estimation of the filter and the vehicle continuing on the same path. This convergence is further exemplified when looking at the error of the filtered position estimate in Figure 4.4. In this figure the filtered estimate starts very high, but quickly converges toward an error of zero. It is also apparent that the inclusion of the estimates from the small signs leads to a small increase in error. This implies that either the small signs should be excluded from the filter or there should be more careful implementation of this data.

The performance of the initial position estimate of the filter can be seen in Figure 4.5. All of the initial raw position estimates from the large sign are a few meters to either side of the true position, but when the average of the first ten points is taken the filtered position estimate appears very close to the true position. From there the filter is able to proceed and track the true position

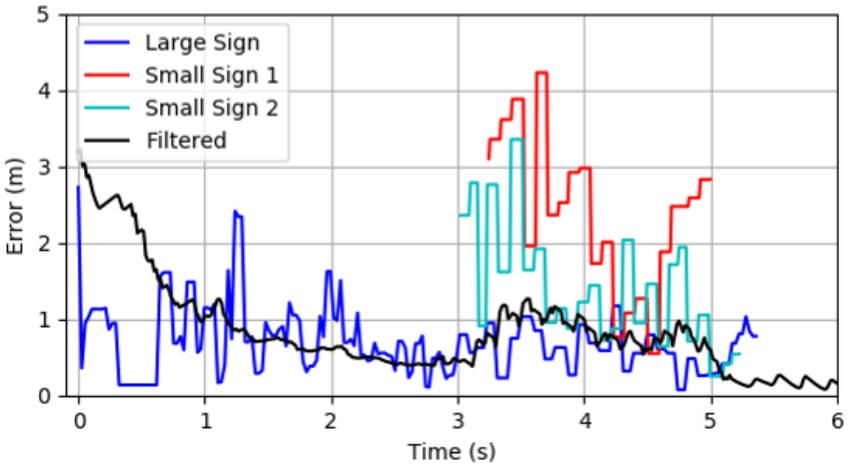

Figure 4.4: Filtered position error for camera mounted at 30°in the left lane

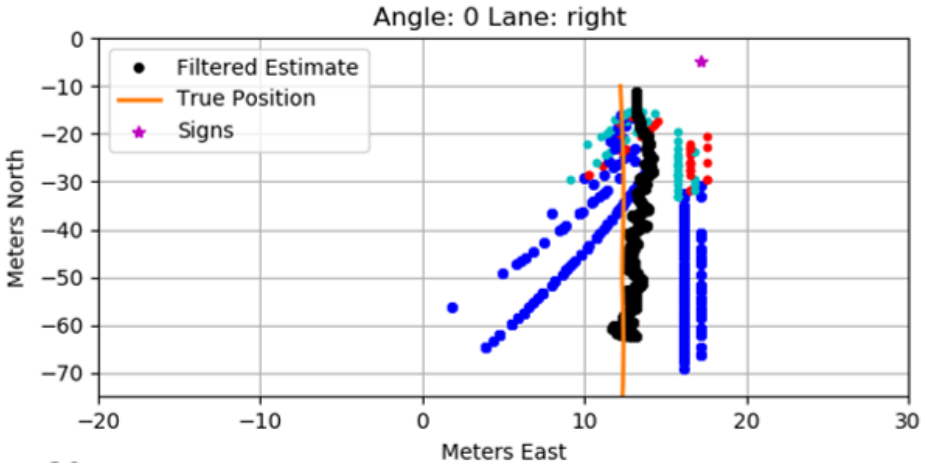

Figure 4.5: Filtered position estimates for camera mounted at 0°in the right lane

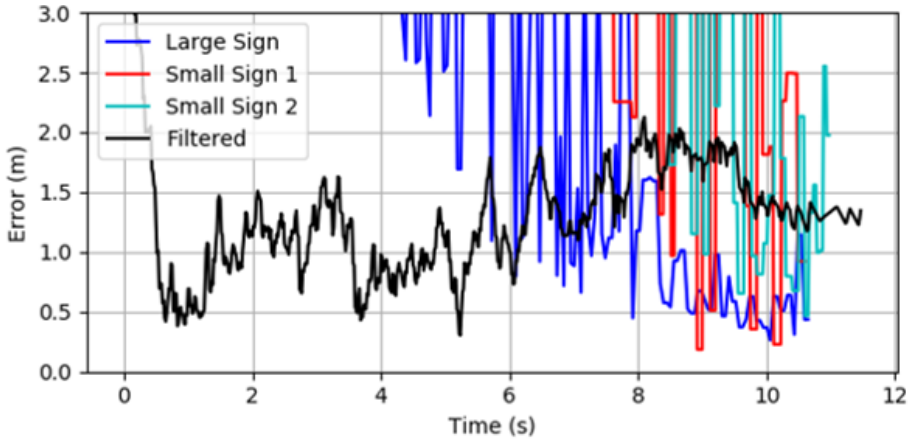

Figure 4.6: Filtered position error for camera mounted at 0°in the right lane

fairly well without having to overcome a large initial error. The significant reduction in error can be seen more clearly in Figure 4.6. Here the error in initial raw position estimates are greater than 3 meters, but very quickly the filter is able to be within 1 meter of the true position and stay relatively accurate until the addition of the smaller signs.

5. CONCLUSIONS

Autonomous vehicles require maximum accuracy and precision to achieve controllability and become a feasible alternative to human driven vehicles. Some sensors, like GPS units, do not inherently provide the high degree of accuracy required for these tasks. This paper has discussed some of the ways the infrastructure can be leveraged to potentially provide high accuracy localization. The infrastructure can be utilized from either perspective, with the sensors on the vehicle or with the sensors on the roadside. These methods have shown to be feasible for this required high accuracy. Both methods, IEA and Smart Code localization have shown the potential for localization within 1 meter accuracy. Additionally, these approaches provide a low computational cost as well as a wider range of uses than other monocular vision approaches previously discussed. Finally, these localization estimates can be further improved when incorporated with sensor fusion techniques and various filtering methods.

6. FUTURE WORK

Both of these localization approaches are in the very early stages of development and will require many iterations of refinement and adjusting to achieve maximum accuracy. Through more rigorous filtering methodologies such as Kalman filters or extended Kalman filters, the position estimates from filters can be improved to reduce error and achieve greater convergence. Additionally, the exact testing setups can be tuned for optimal results by changing the camera height and orientation in IEA or finding the optimal configuration of signs for the Smart Code localization. Beyond improvements of the filtering and tuning of the experimentation, an extension of the experimentation is required. By testing on a larger scale with more variety more of the patterns and behaviors of the localization approaches can be determined. This will help aided in the analysis of the feasibility of these methodologies in real world applications. Included with these experimental expansions will be the use of path planning and control algorithms developed specifically for these environments.

REFERENCES

[1] J.-P. Tardif and Y. Pavlidis, "Monocular visual odometry in urban environments using an omnidirectional camera," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

[2] X. Wang, "Vision-based detection and tracking of a mobile ground target using a fixed-wing uav," *Internarional Journal of Advanced Robotic Systems*, vol. 11, p. 156, 2014.

[3] S. Li and Y. Meng, "Monocular vision-based vehicle localization aided by fine-grained classification," 2018.

[4] S. Gopalswamy and S. Rathinam, "Infrastructure enabled autonomy: A distributed intelligence architecture for autonomous vehicles," *IEEE Intelligent Vehicles Symposium 2018*.

[5] A. Burch and S. Saripalli, "Infrastructure enabled autonomy for vehicles," *21st IEEE International Conference on Intelligent Transportation Systems*, 2018.