Self-Calibration of the Offset Between GPS and Semantic Map Frames for Robust Localization

by

Wei-Kang Tseng

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science

Graduate Department of Institute for Aerospace Studies
University of Toronto

Self-Calibration of the Offset Between GPS and Semantic Map Frames for Robust Localization

Wei-Kang Tseng
Master of Applied Science
Graduate Department of Institute for Aerospace Studies
University of Toronto
2021

Abstract

In self-driving, GPS is generally considered to have insufficient positioning accuracy to stay in lane. Instead, many turn to LIDAR localization, but building and maintaining LIDAR maps can be costly. Another possibility is to use semantic cues such as lane markings and traffic lights to achieve localization, but these are usually not continuously visible. This can be remedied by combining semantic cues with GPS to fill in the gaps. However, due to biases accumulated between mapping and localization, the live GPS frame can be offset from the semantic map frame, requiring calibration. In this thesis, we propose a robust semantic localization algorithm that self-calibrates for the GPS-to-map offset by exploiting common semantic cues. We formulate the problem using a modified Iterated Extended Kalman Filter, which incorporates GPS and camera images for semantic cue detection. Experimental results show that the proposed algorithm achieves decimetre-level accuracy and is robust against sparse semantic features and frequent GPS dropouts.

Acknowledgements

I would first like to express my deepest gratitude to my supervisors, Professor Tim Barfoot and Professor Angela Schoellig. They have contributed many key ideas to my thesis and always kept me on the right track. Without their support and guidance, this thesis would have never been accomplished.

Thank you to all the members of Autonomous Space Robotics Laboratory for creating an inclusive and welcoming environment to conduct my research in. A special shout out to David and Jeremy for patiently explaining to me their research work, which became the mathematical foundation of my thesis.

During this degree, I had the pleasure to be part of the aUToronto team, which gave me invaluable hands-on experience with our self-driving car, *Zeus*. I also want to thank aUToronto for providing access to its internal programs and dataset to facilitate my thesis experiments. In particular, I really appreciate the technical support from Joe and Ziyad.

I also want to thank the Dean's Strategic Fund and NSERC for funding my project. Beyond that, I really appreciate the part-time opportunity offered by Tim, which helped address my financial concerns during my studies.

Most importantly, I would not have made it this far without the unfailing support of my mother. Your words of encouragement and unconditional love is the reason I am able to overcome all the challenges and complete the degree.

Contents

List of Tables

List of Figures

Notations

$(\cdot)_k$	The value of a quantity at timestep k.
$\underrightarrow{\boldsymbol{\mathcal{F}}}_a$	A vectrix representing a reference frame in three dimensions
$\mathbf{1}$	The identity matrix.
$\mathbf{0}$	The zero matrix.
$\mathbb{R}^{M\times N}$	The vectorspace of real $M \times N$ matrices
$\hat{(\cdot)}$	A posterior (estimated) quantity
$\check{(\cdot)}$	A prior quantity
$SO(3)$	The special orthogonal group, a matrix Lie group used to represent rotations
$\mathfrak{so}(3)$	The Lie algebra associated with $SO(3)$
$SE(3)$	The special Euclidean group, a matrix Lie group used to represent poses
$\mathfrak{se}(3)$	The Lie algebra associated with $SE(3)$
$(\cdot)^\wedge$	An operator associated with the Lie algebra for rotations and poses
$(\cdot)^\curlywedge$	An operator associated with the adjoint of an element from the Lie algebra for poses
$(\cdot)^\odot$	An operator associated with the Jacobian term for the Lie algebra

$\mathrm{Ad}(\cdot)$	An operator producing the adjoint of an element from the Lie group for rotations and poses
$\boldsymbol{\mathcal{J}}(\cdot)$	The left Jacobian of $SE(3)$ in Lie group theory
$\mathbf{C}_{ba}$	A 3×3 rotation matrix (member of $SO(3)$) that takes points expressed in $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$ and re-expresses them in $\underrightarrow{\boldsymbol{\mathcal{F}}}_b$, which is rotated with respect to $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$
$\mathbf{T}_{ba}$	A 4×4 transformation matrix (member of $SE(3)$) that takes points expressed in $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$ and re-expresses them in $\underrightarrow{\boldsymbol{\mathcal{F}}}_b$, which is rotated/translated with respect to $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$
$\boldsymbol{\mathcal{T}}_{ba}$	A 6×6 adjoint of a transformation matrix (member of $\mathrm{Ad}(SE(3))$)

Chapter 1

Introduction

1.1 Background and Motivation

In autonomous driving applications, semantic maps have proven to be an invaluable component for most self-driving cars. They provide important prior knowledge of the surrounding environment, including the locations of drivable lanes, traffic lights, and traffic signs, as well as the traffic rules. This information is crucial for real-time behavioural planning of the vehicle under various traffic scenarios.

In order to effectively utilize semantic maps, the vehicle must be localized in the map frame down to decimetre accuracy. This proves to be challenging for the Global Positioning System (GPS), where even the best corrected version of GPS is generally considered inadequate in achieving the required accuracy consistently. Furthermore, GPS suffers from signal dropouts in situations such as inside tunnels or in dense urban environments. In light of these issues, many self-driving systems have adopted LIDAR (Light Detection and Ranging) localization methods, which require the construction of LIDAR maps prior to driving in a certain area. LIDAR localization has demonstrated great success in satisfying the stringent requirements of autonomous driving [10, 44, 1], but this comes at the cost of building detailed geometric models of the world and keeping them up to date solely for the purpose of localization. Moreover, because the autonomous driving system, and in particular the planning component, ultimately requires the vehicle's location with respect to the

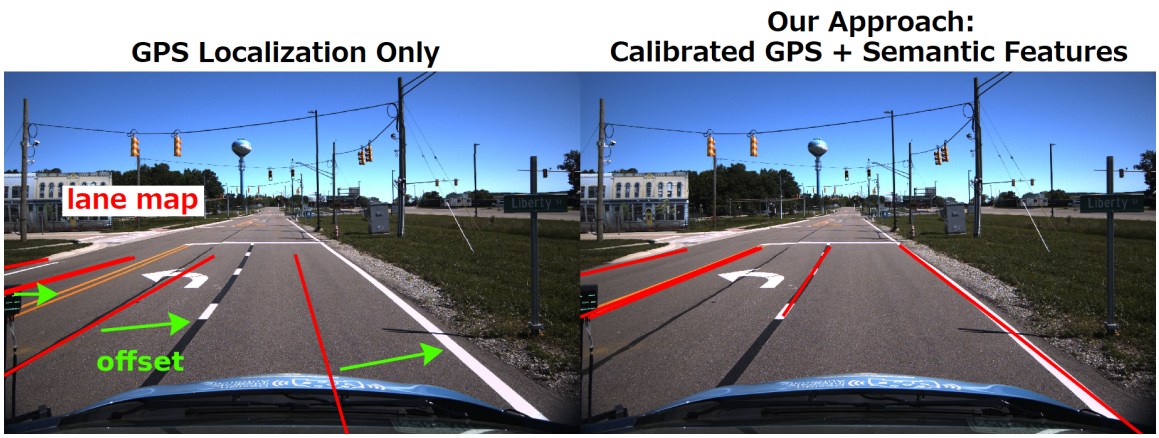


Figure 1.1: Vehicle localization using uncalibrated GPS (left) compared to our approach (right). The red lines are the projected lane boundaries from the semantic map. Our approach is able to self-calibrate for the GPS-to-map offset and achieve alignment between the observed lane markings and the projected lane boundaries [4].

semantic map, it requires the additional step of aligning the LIDAR maps with the semantic maps.

An alternative to LIDAR localization is to directly take advantage of the semantic maps for localization, which the self-driving vehicle already utilizes for path planning and behavioural decision making. Through the detection of some common objects on the road (e.g., in the vehicle's camera images) such as traffic lights and lane markings that are also present in the semantic maps, the vehicle location can be inferred. A major downside of such an approach is that these objects, collectively termed "semantic cues", are fairly sparse and not always present in enough numbers to ensure reliable localization. A potential solution is to adopt a hybrid approach that combines GPS and semantic cues. However, a new problem arises: the offset between the semantic map frame and the GPS frame, in which the vehicle position is reported, must be known accurately before fusing the two sources of information. This offset is a common issue and emerges because the semantic maps are aligned to the global frame using GPS data gathered at a different time/day than when the live drive occurs. Therefore, due to different positioning of the satellites in the sky and varying atmospheric conditions [22], among other factors, there will be a GPS-to-map offset requiring calibration such that the GPS frame aligns with the semantic map

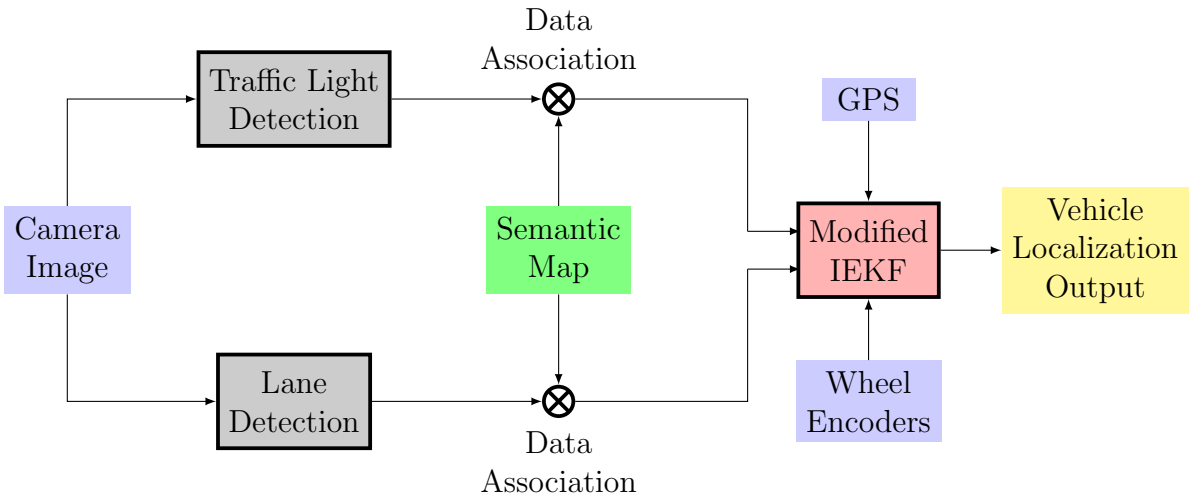


Figure 1.2: System architecture of our proposed localization pipeline. The camera image is passed through the lane and traffic light detectors. The data association step finds the correspondences between detection results and the semantic map projected into the image space. The results of the data association are then fused with GPS and wheel encoders in a modified IEKF to produce the final localization output.

frame. Usually, such offset is simply corrected by hand on an occasional basis, but such manual calibration is generally not reliable.

As an illustrative example, aUToronto, the team that won the self-driving competition hosted by SAE International in 2019 [4], experienced an uncalibrated GPS-to-map offset in the magnitude of a few metres, which was corrected manually just in time for the competition run, see Figure 1.1.

To address these challenges, we propose a robust localization algorithm that integrates GPS and semantic cues while performing self-calibration of the offset between the GPS and semantic map frames. By folding the offset into our state estimation, we can properly fuse the two sources of information while benefitting from both. For this work, we assume detection of semantic cues using a front-facing monocular camera, and formulate the localization problem as a modified Iterated Extended Kalman Filter (IEKF), which improves upon the linearization of EKF. The system architecture is summarized in Figure 1.2.

The proposed approach has minimal computational impact because GPS is low-cost to process, and common semantic cues such as lane markings and traffic lights

are already tracked for the purpose of vehicle behavioural planning, so the added cost of using them is also low. The result is an accurate and robust self-driving localization pipeline that uses GPS to fill in the gaps between sparse semantic observations, avoids the need for expensive maps specifically for localization, and relies on features in the environment that are actively maintained and designed to be highly visible. Experimental results in an urban environment using the Carla simulator [11] as well as on a real-world dataset collected by aUToronto during the SAE AutoDrive competition show that we are able to achieve 3 cm lateral and 5 cm longitudinal accuracy on average, and also maintain similar performance with frequent GPS dropouts.

1.2 Contribution

The main contributions of this thesis are:

1. An online semantic localization algorithm for autonomous vehicles that simultaneously self-calibrates for the GPS-to-map offset by incorporating multiple classes of semantic cues.

2. A novel mathematical formulation of the vehicle localization problem in 3D space that processes the semantic cue detection results directly in the image space rather than in bird's-eye view.

3. A localizer that has minimal computational impact by taking advantage of the existing infrastructures on the autonomous vehicle, including semantic maps and detectors of various semantic cues.

4. The addition of wheel encoders to improve the robustness of the localizer against frequent GPS dropouts.

5. The proposal of a semantic map refinement pipeline that can potentially produce accurate semantic maps at low cost using satellite images.

The core of this work has previously been published in our paper for the Conference on Robots and Vision in a condensed manner [39]. It corresponds to the first four

contributions above. The fifth contribution, on the other hand, is newly conducted work that involves a semantic map refinement pipeline designed to improve an internal dataset used in the experiments.

1.3 Overview

The thesis is organized as follows. Chapter 2 summarizes the related work on vehicle localization. Chapter 3 describes the preprocessing necessary for the semantic cues before semantic localization can take place. Chapter 4 presents the mathematical formulation of the semantic localization algorithm. Chapter 5 provides the simulation and experimental results. Finally, Chapter 6 concludes the paper and discusses possible future work.

Chapter 2

Related Work

2.1 LIDAR Localization

One of the most popular localization approaches in self driving is LIDAR localization [13, 17, 12, 23]. By constructing a database of the detailed geometry of the environment in advance, localization can be achieved using a point cloud registration algorithm, which matches the LIDAR scans against the database at test time. Because the localization performance greatly depends on the accuracy of the database in capturing the ever-changing appearance of the world, the database needs to be frequently updated. In response, many have developed algorithms that extract features that are more invariant to environmental changes in the LIDAR data [46, 18, 26]. More recently, [27] proposed a novel learning-based approach that directly takes LIDAR point clouds as inputs and learns descriptors for matching in various driving scenarios. While these methods help mitigate the impact of outdated LIDAR database, they do not fundamentally address the issue of needing to maintain a separate database solely for localization.

2.2 Semantic Localization

Semantic localization exploits various common roadside semantic cues present in the semantic maps to achieve vehicle localization. In contrast to LIDAR localization, this

Table 2.1: List of public self-driving datasets with semantic maps and the semantic cues that are included.

Dataset	Available Semantic Cues inside Semantic Map				
	Lane Markings	Stop Lines	Other Road Markings	Traffic Lights	Traffic Signs
nuScenes (2019) [5]	✓	✓	×	✓	×
Lyft Level 5 (2019) [21]	✓	×	×	✓	✓
Argoverse (2019) [6]	×*	×	×	×	×

*Only has lane centerlines and lane polygons

method conveniently makes use of the same semantic maps already required by the autonomous vehicle for planning purposes. Therefore, no maintenance of a separate database of the environment is required. Among the various types of semantic cues, lane markings are most commonly utilized because they are abundant and provide important clues that keep the vehicle in the correct lane [14, 15, 7, 40, 34]. However, since lane markings tend to run parallel to the vehicle heading, the longitudinal localization accuracy is usually worse than lateral accuracy. Besides lane markings, other types of semantic cues have been exploited as well, including stop lines [32, 29], other road markings [20, 45, 35], traffic lights [41, 42], and traffic signs [43, 30, 33, 8]. A common issue that all types of semantic cues suffer from is sparsity. In response, approaches that combine multiple types of semantic cues have been proposed, most of which include lane markings in combination with traffic lights or traffic signs [28, 9, 25].

Because until recent years, there is a lack of public self-driving datasets that provide semantic maps, these prior works had to conduct experiments using internal datasets and produce their own semantic maps. The map generation process usually involves a combination of camera and LIDAR data followed by manual annotations. In the past few years, however, a few self-driving datasets with semantic maps have emerged. The semantic cues supported by these datasets are summarized in Table 2.1. The applicability of these datasets for our work is discussed in Section 5.2.

Many of the semantic localization papers referenced in this section have incorpo-

rated GPS into their localization pipelines, but none of them addressed a possible offset between GPS and semantic map frames due to reasons discussed above, presumably because the GPS offset has been manually corrected prior to experiments. However, as experienced by aUToronto, manual GPS calibration is often unreliable, and can lead to localization failures [4].

2.3 GPS Calibration with Semantic Cues

In this work, the GPS measurements are regarded as reporting the vehicle position with respect to a GPS frame, which is at an offset from the semantic map frame. Alternatively, we can treat the GPS as if it directly reports the vehicle position in the semantic map frame, but with a systematic bias. Some prior works took this fact into account when developing their semantic localization pipelines. For instance, [24] simply modelled the GPS errors as a random constant since the change in the GPS bias is small. A more sophisticated model utilizing an autoregressive process such as a random walk was shown by [38] to achieve superior performance compared to the random constant model, and was similarly adopted by [19] and [37]. All of these approaches only adopted road markings as the semantic cues. While our approach is similar in spirit to these papers, there are also notable differences, including the addition of traffic lights as part of the semantic cues, and their detections using Convolutional Neural Networks (CNNs).

Chapter 3

Semantic Cue Preprocessing

The positional information of the semantic cues provided by the semantic maps plays a crucial role in the development of semantic localization. To make use of this information, the vehicle must be capable of detecting nearby semantic cues in real time using onboard sensors. Fortunately, because the semantic cues provide important traffic information, we can safely assume that such detectors are already in place as part of the autonomous driving system, thus minimizing the impact on the computational cost. In this work, we assume the sensor to be a common front-facing camera with plenty of off-the-shelf CNN image detectors available from which to choose. Lastly, for the detections to be useful, some calculations have to take place that associate the detections with the semantic map.

3.1 Semantic Map

The lightweight HD semantic maps are commonly equipped by autonomous vehicles for planning and navigation. They are often supplied by commercial mapping companies such as HERE and CARMERA, and have become increasingly accessible as the companies continue to map more regions around the world. The quality of the semantic map can potentially have a great impact on the performance of the vehicle.

Our semantic localization algorithm utilizes a HD semantic map that consists of a lane graph and traffic light locations. A lane graph is a set of polylines that defines

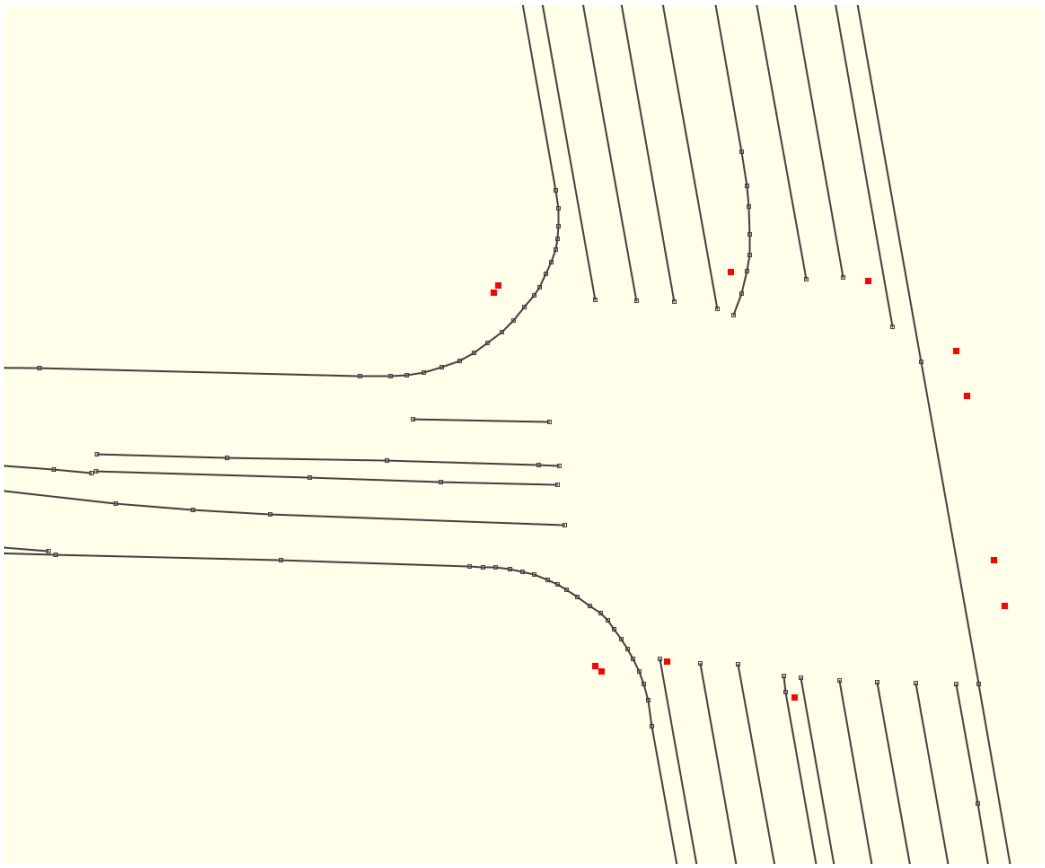


Figure 3.1: Example of a semantic map at a road intersection. The black polylines form the lane graph, and the red points are traffic lights.

all the lane boundaries of the road network. It corresponds to visually distinctive lane markings as well as road curbs, which can be easily identified in a camera image by the CNN detector. The traffic lights are treated as point landmarks where the coordinates of their centres are recorded in the semantic map. Their orientations are not included. Figure 3.1 illustrates a semantic map with lane graph and traffic lights. In this work, we assume the semantic map has been provided.

3.2 Traffic Light Detection

Traffic lights always appear sparsely yet regularly at road intersections. They are crucial to the vehicle's understanding of the traffic situation surrounding it, and also provide useful information for longitudinal localization of the vehicle. Given a camera

Figure 3.2: Example of YOLOv3 traffic light detection. A red bounding box in which the center point can be extracted is shown around each detection.

image, the traffic light CNN detector outputs bounding boxes that locate all the traffic lights identified. The centre of each bounding box is then obtained as the observed point landmark of a traffic light. Since the detector assigns a confidence level to each bounding box, we can reliably filter out false detections by only keeping bounding boxes with high confidence scores for localization. In this work, we adopted YOLOv3 for the detection of traffic lights [31]. An example output of YOLOv3 traffic light detector is shown in Figure 3.2.

Before the traffic light detections can be made useful for localization, a data association scheme must first be devised to correctly associate the detections in the camera image with corresponding traffic lights in the semantic map. This is achieved by first projecting the locations of all nearby traffic lights in the semantic map that are in front of the vehicle to the image space using the estimated vehicle position. We then

apply Iterative Closest Point followed by nearest neighbour to obtain the desired data associations. Detections that have no nearby associations within a certain distance threshold are identified as outliers and discarded. Figure 3.4 illustrates the results of the traffic light data association process.

3.3 Lane Marking Detection

Lane markings are one of the most common type of semantic cues that primarily help with lateral localization of the vehicle to keep it in the correct lane. Given a camera image, a lane marking CNN detector produces a grayscale mask where the value of each pixel corresponds to the probability of the pixel being part of a lane marking in the camera image. A probability threshold is then applied to the grayscale mask to obtain a binary mask, which classifies each pixel as being part of a lane marking or not. Because lane makings closer to the vehicle, which corresponds to the bottom portion of the camera image, are easier to identify than those further away, only the bottom portion of the mask is retained. The resulting image coordinates of the pixels classified as lane markings are then evenly subsampled to reduce the computational burden.

In this work, we adopted the gated shape CNN (GSCNN) lane marking detector [36] and trained it on the BDD100K dataset [47]. The GSCNN detector captures lane markings as well as road boundaries from the camera observations, and has a low rate of false positives. This is preferable over a detector with a low rate of false negatives because the detection of non-existent lane markings has a greater impact on the process of data association compared to false negatives. Furthermore, GSCNN detector is capable of inferring continuous lane boundaries from not just the solid lane markings, but also the dashed ones. An example of GSCNN lane detector is shown in Figure 3.3.

The data association process for lane markings also begins by finding all the lane lines that are close to and in front of the vehicle and projecting them from the semantic map to the image space using the estimated vehicle position. Next, each

(a) Camera image (b) Lane detection mask (c) Lane binary mask

Figure 3.3: Example of GSCNN lane detector. The camera image (a) is input into the detector, which produces the mask (b). A threshold is then applied, resulting in the binary mask (c). Notice the dashed lane marking on the right still produces a continuous lane boundary detection.

subsampled lane pixel from the binary mask is matched to the nearest projected lane line. Lane pixels without nearby matches are treated as outliers and discarded. Figure 3.4 demonstrates the result of such lane marking matching process. Now, given all the lane pixels that are matched to a projected lane line, we can fit a straight line using least squares in the image space. Finally, data association is obtained between the pairs of fitted lines from lane marking detection and projected lane lines from the semantic map. Aside from the lane markings that are too distant, those that are too far to the sides of the ego-lane are also difficult to identify properly by the lane detector due to the camera view angle. Therefore, to avoid such lane misassociations, we perform a check on all the fitted lines by computing the intersection between it and the bottom edge of the image. If the point of intersection is too far outside the image, the fitted line is discarded.

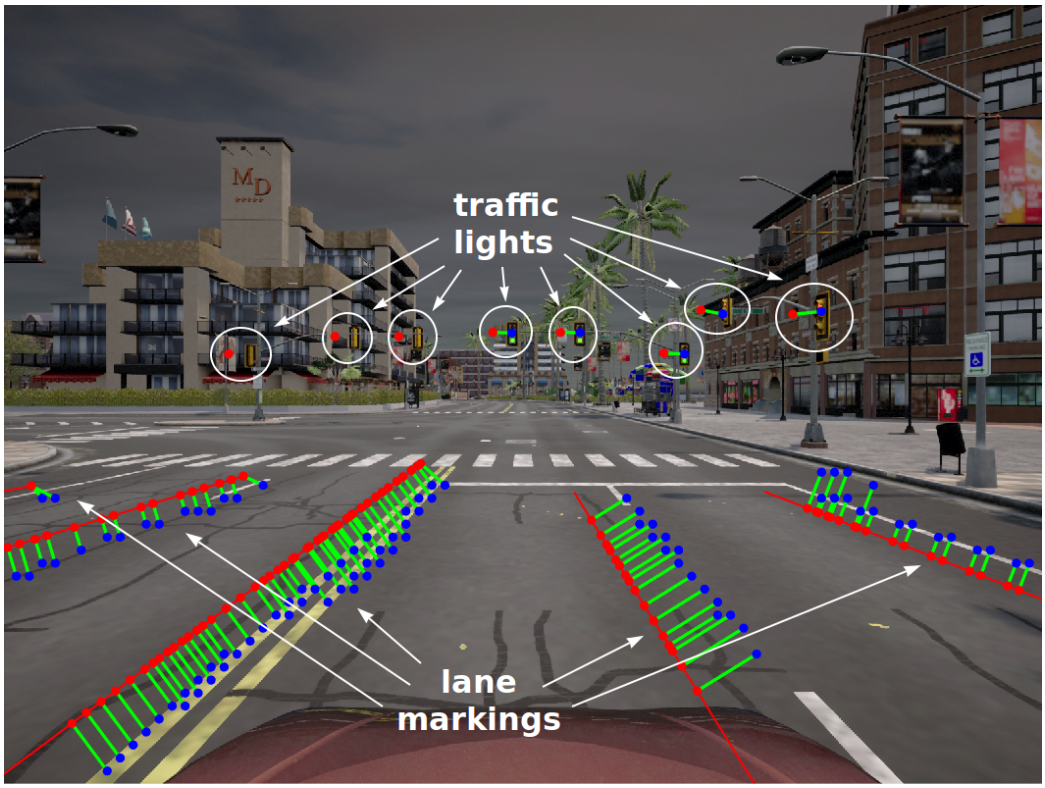


Figure 3.4: Data association process of traffic lights and lane markings. The red lines and points are known positions of semantic cues projected into the image using the estimated vehicle position; the blue points are semantic cue detections; and the green lines indicate the matching results. A few incorrect matches of outlier lane pixels can be observed on the right due to their proximity to a projected lane line as a result of the camera angle.

Chapter 4

Vehicle Localization

4.1 Problem Setup

We adopt the mathematical notations from [3] and formulate the semantic localization problem with GPS offset by first discretizing time denoted by subscript k. There are three reference frames. $\underrightarrow{\boldsymbol{\mathcal{F}}}_M$ is the non-moving frame associated with the semantic map, $\underrightarrow{\boldsymbol{\mathcal{F}}}_{V,k}$ is attached to a moving vehicle, and $\underrightarrow{\boldsymbol{\mathcal{F}}}_{G,k}$ is the GPS frame, which is at an offset from $\underrightarrow{\boldsymbol{\mathcal{F}}}_M$. We then have three corresponding transformation matrices between the frames. $\mathbf{T}_{VG,k} \in SE(3)$ is the GPS measurement of the pose of vehicle that is corrupted by noise, $\mathbf{T}_{GM,k} \in SE(3)$ is the GPS-to-map offset, which needs to be estimated for self-calibration, and $\mathbf{T}_{VM,k} \in SE(3)$ is the pose of vehicle with respect to the semantic map, which we ultimately desire. Figure 4.1 illustrates the described problem setup.

At time step k, the j-th semantic cue, P^j, detected by the onboard camera has the pixel coordinates, $\mathbf{p}^j_{I,k} \in \mathbb{R}^2$, as well as its known location in the map frame, $\mathbf{p}^j_M \in \mathbb{R}^3$, obtained from the semantic map. Using $\mathbf{T}_{VM,k}$, we can transform and project the known location, $\mathbf{p}^j_M$, to the image space and obtain the reprojection error for simultaneous localization and GPS-to-map offset calibration.

Given the camera measurements of semantic cues, $\mathbf{p}^j_{I,k}$, and their corresponding locations known in the map, $\mathbf{p}^j_M$, as well as the GPS measurements, $\mathbf{T}_{VG,k}$, we will

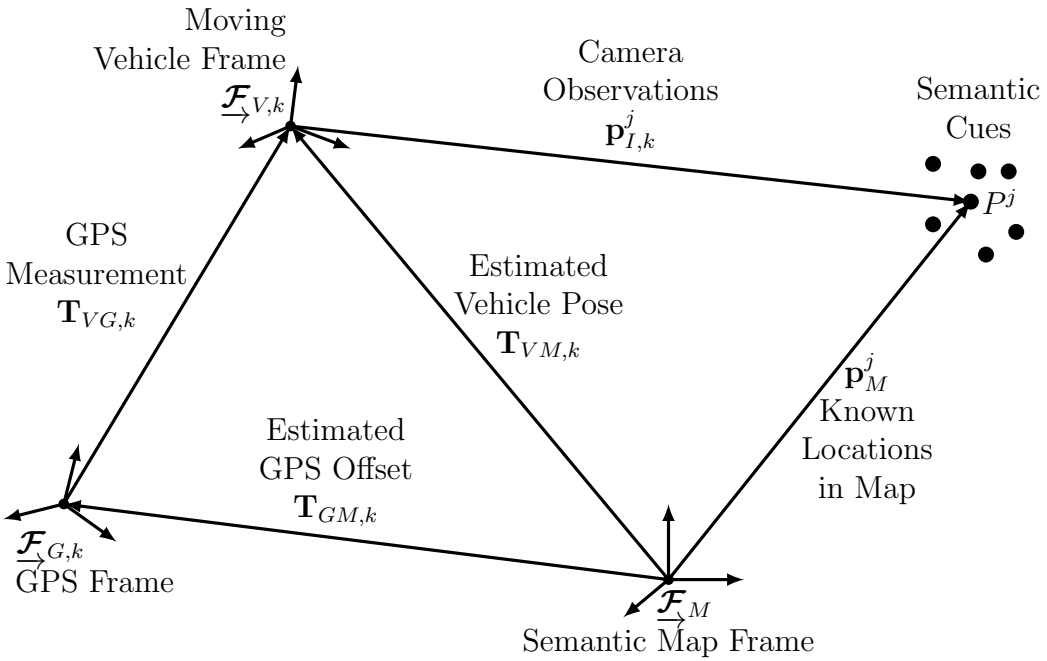


Figure 4.1: Definition of reference frames for the localization problem with semantic cues and offset between GPS and semantic map frames.

estimate the vehicle pose,

$$\mathbf{T}_{VM,k} \sim \mathcal{N}\left(\bar{\mathbf{T}}_{VM,k}, \mathbf{\Sigma}_{VM,k}\right), \tag{4.1}$$

and GPS-to-map offset,

$$\mathbf{T}_{GM,k} \sim \mathcal{N}\left(\bar{\mathbf{T}}_{GM,k}, \mathbf{\Sigma}_{GM,k}\right), \tag{4.2}$$

where we assume them to be Gaussians. Their respective means are $\bar{\mathbf{T}}_{VM,k}$ and $\bar{\mathbf{T}}_{GM,k} \in SE(3)$, and their associated covariance matrices are $\mathbf{\Sigma}_{VM,k}$ and $\mathbf{\Sigma}_{GM,k} \in \mathbb{R}^{6\times 6}$, respectively. Using the technique of perturbation, we can express them as

$$\mathbf{T}_{VM,k} = \exp(\delta\boldsymbol{\xi}^{\wedge}_{VM,k})\bar{\mathbf{T}}_{VM,k}, \tag{4.3}$$

$$\mathbf{T}_{GM,k} = \exp(\delta\boldsymbol{\xi}^{\wedge}_{GM,k})\bar{\mathbf{T}}_{GM,k}, \tag{4.4}$$

where the perturbation terms, $\delta\boldsymbol{\xi}_{VM,k}, \delta\boldsymbol{\xi}_{GM,k} \in \mathbb{R}^6$, are zero-mean Gaussians,

$$\delta\boldsymbol{\xi}_{VM,k} \sim \mathcal{N}\left(\mathbf{0}_{6\times 1}, \boldsymbol{\Sigma}_{VM,k}\right), \tag{4.5}$$

$$\delta\boldsymbol{\xi}_{GM,k} \sim \mathcal{N}\left(\mathbf{0}_{6\times 1}, \boldsymbol{\Sigma}_{GM,k}\right). \tag{4.6}$$

In addition, the three-dimensional translational and rotational velocities of the vehicle, $\boldsymbol{\varpi}_k \in \mathbb{R}^6$, expressed in the vehicle frame $\underrightarrow{\mathcal{F}}_{V,k}$, are also estimated. Similar to $\mathbf{T}_{VM,k}$ and $\mathbf{T}_{GM,k}$, we can also break it down into the mean, $\bar{\boldsymbol{\varpi}}_k \in \mathbb{R}^6$, and the zero-mean Gaussian perturbation term, $\delta\boldsymbol{\varpi}_k \in \mathbb{R}^6$,

$$\boldsymbol{\varpi}_k = \bar{\boldsymbol{\varpi}}_k + \delta\boldsymbol{\varpi}_k. \tag{4.7}$$

It is important to note that for the sake of the following mathematical formulations, we have independently expressed the states mentioned above. However, these states are in fact correlated and have to be jointly estimated, which is addressed when we formulate the IEKF algorithm in Section 4.4.

4.2 Process Models

4.2.1 Vehicle Pose Process Model

We adopt the white-noise-on-acceleration model [2]. By assuming constant vehicle velocity between consecutive discrete time steps, the process model of the vehicle pose is

$$\begin{aligned}\mathbf{T}_{VM,k} &= \exp(\mathbf{w}_{VM}^{\wedge})\overbrace{\exp(\Delta t_k \boldsymbol{\varpi}_{k-1}^{\wedge})}^{\mathbf{T}_{\boldsymbol{\varpi},k-1}}\mathbf{T}_{VM,k-1} \\ &= \exp(\mathbf{w}_{VM}^{\wedge})\mathbf{T}_{\boldsymbol{\varpi},k-1}\mathbf{T}_{VM,k-1},\end{aligned} \tag{4.8}$$

where $\Delta t_k = t_k - t_{k-1}$ is the time interval, and $\mathbf{w}_{VM}$ is the zero-mean Gaussian process noise.

We apply the same perturbation technique as in (4.3) and (4.4) to model $\mathbf{T}_{\boldsymbol{\varpi},k-1} \sim$

$\mathcal{N}\left(\bar{\mathbf{T}}_{\boldsymbol{\varpi},k-1}, \boldsymbol{\Sigma}_{\boldsymbol{\varpi},k-1}\right)$:

$$\mathbf{T}_{\boldsymbol{\varpi},k-1} = \exp(\delta\boldsymbol{\xi}^{\wedge}_{\boldsymbol{\varpi},k-1})\bar{\mathbf{T}}_{\boldsymbol{\varpi},k-1}, \tag{4.9}$$

where

$$\delta\boldsymbol{\xi}_{\boldsymbol{\varpi},k-1} \sim \mathcal{N}\left(\mathbf{0}_{6\times 1}, \boldsymbol{\Sigma}_{\boldsymbol{\varpi},k-1}\right), \tag{4.10}$$

is the zero-mean Gaussian perturbation term.

Next, by substituting the expressions (4.3) and (4.9) into the right-hand side of the process model (4.8), we have

$$\begin{aligned}
\mathbf{T}_{VM,k} &= \exp(\mathbf{w}^{\wedge}_{VM})\mathbf{T}_{\boldsymbol{\varpi},k-1}\mathbf{T}_{VM,k-1} \\
&= \exp(\mathbf{w}^{\wedge}_{VM})\exp(\delta\boldsymbol{\xi}^{\wedge}_{\boldsymbol{\varpi},k-1})\underbrace{\bar{\mathbf{T}}_{\boldsymbol{\varpi},k-1}\exp(\delta\boldsymbol{\xi}^{\wedge}_{VM,k-1})}_{\exp((\bar{\boldsymbol{\mathcal{T}}}_{\boldsymbol{\varpi},k-1}\delta\boldsymbol{\xi}_{VM,k-1})^{\wedge})\bar{\mathbf{T}}_{\boldsymbol{\varpi},k-1}}\bar{\mathbf{T}}_{VM,k-1} \\
&= \exp(\mathbf{w}^{\wedge}_{VM})\exp(\delta\boldsymbol{\xi}^{\wedge}_{\boldsymbol{\varpi},k-1})\exp((\bar{\boldsymbol{\mathcal{T}}}_{\boldsymbol{\varpi},k-1}\delta\boldsymbol{\xi}_{VM,k-1})^{\wedge})\bar{\mathbf{T}}_{\boldsymbol{\varpi},k-1}\bar{\mathbf{T}}_{VM,k-1}.
\end{aligned} \tag{4.11}$$

If we assume the nominal kinematics to be

$$\bar{\mathbf{T}}_{VM,k} = \bar{\mathbf{T}}_{\boldsymbol{\varpi},k-1}\bar{\mathbf{T}}_{VM,k-1}, \tag{4.12}$$

and compare the right-hand side of the expressions (4.3) and (4.11), then

$$\exp(\delta\boldsymbol{\xi}^{\wedge}_{VM,k}) = \exp(\mathbf{w}^{\wedge}_{VM})\exp(\delta\boldsymbol{\xi}^{\wedge}_{\boldsymbol{\varpi},k-1})\exp((\bar{\boldsymbol{\mathcal{T}}}_{\boldsymbol{\varpi},k-1}\delta\boldsymbol{\xi}_{VM,k-1})^{\wedge}). \tag{4.13}$$

Since all $\mathbf{w}_{VM}$, $\delta\boldsymbol{\xi}_{VG,k}$, and $\delta\boldsymbol{\xi}_{GM,k-1}$ are small noise or perturbation terms, we can approximate the equation (4.13) using Baker–Campbell–Hausdorff (BCH) formula by keeping the first order terms:

$$\begin{aligned}
\delta\boldsymbol{\xi}_{VM,k} &\approx \mathbf{w}_{VM} + \delta\boldsymbol{\xi}_{\boldsymbol{\varpi},k-1} + \bar{\boldsymbol{\mathcal{T}}}_{\boldsymbol{\varpi},k-1}\delta\boldsymbol{\xi}_{VM,k-1} \\
&= \bar{\boldsymbol{\mathcal{T}}}_{\boldsymbol{\varpi},k-1}\delta\boldsymbol{\xi}_{VM,k-1} + \delta\boldsymbol{\xi}_{\boldsymbol{\varpi},k-1} + \mathbf{w}_{VM},
\end{aligned} \tag{4.14}$$

which is the perturbation kinematics.

Notice that we need to relate $\bar{\mathbf{T}}_{\boldsymbol{\varpi},k-1}$ and $\delta\boldsymbol{\xi}_{\boldsymbol{\varpi},k-1}$ back to $\bar{\boldsymbol{\varpi}}_{k-1}$ and $\delta\boldsymbol{\varpi}_{k-1}$, which

are the mean and perturbation of the velocity state vector $\boldsymbol{\varpi}_{k-1}$. We start with the substitution made in (4.8),

$$\mathbf{T}_{\boldsymbol{\varpi},k-1} = \exp(\Delta t_k \boldsymbol{\varpi}_{k-1}^{\wedge}), \tag{4.15}$$

and substitute in the perturbation expressions (4.7) and (4.9):

$$\begin{aligned}\exp(\delta\boldsymbol{\xi}_{\boldsymbol{\varpi},k-1}^{\wedge})\bar{\mathbf{T}}_{\boldsymbol{\varpi},k-1} &= \exp(\Delta t_k(\bar{\boldsymbol{\varpi}}_{k-1} + \delta\boldsymbol{\varpi}_{k-1})^{\wedge}) \\ &\approx \exp(\Delta t_k(\boldsymbol{\mathcal{J}}(\Delta t_k\bar{\boldsymbol{\varpi}}_{k-1})\delta\boldsymbol{\varpi}_{k-1})^{\wedge})\exp(\Delta t_k\bar{\boldsymbol{\varpi}}_{k-1}^{\wedge}).\end{aligned} \tag{4.16}$$

If we assume the nominal term to be

$$\bar{\mathbf{T}}_{\boldsymbol{\varpi},k-1} = \exp(\Delta t_k\bar{\boldsymbol{\varpi}}_{k-1}^{\wedge}), \tag{4.17}$$

then we are left with

$$\exp(\delta\boldsymbol{\xi}_{\boldsymbol{\varpi},k-1}^{\wedge}) \approx \exp(\Delta t_k(\boldsymbol{\mathcal{J}}(\Delta t_k\bar{\boldsymbol{\varpi}}_{k-1})\delta\boldsymbol{\varpi}_{k-1})^{\wedge}). \tag{4.18}$$

Comparing the exponents on both sides, we obtain the relationship between the perturbation terms $\delta\boldsymbol{\xi}_{\boldsymbol{\varpi},k-1}$ and $\delta\boldsymbol{\varpi}_{k-1}$:

$$\delta\boldsymbol{\xi}_{\boldsymbol{\varpi},k-1} \approx \Delta t_k\boldsymbol{\mathcal{J}}(\Delta t_k\bar{\boldsymbol{\varpi}}_{k-1})\delta\boldsymbol{\varpi}_{k-1}. \tag{4.19}$$

We can now substitute the relationships (4.17) and (4.19) back into (4.12) and (4.14) to arrive at the desired nominal kinematics:

$$\begin{aligned}\bar{\mathbf{T}}_{VM,k} &= \bar{\mathbf{T}}_{\boldsymbol{\varpi},k-1}\bar{\mathbf{T}}_{VM,k-1} \\ &= \exp(\Delta t_k\bar{\boldsymbol{\varpi}}_{k-1}^{\wedge})\bar{\mathbf{T}}_{VM,k-1},\end{aligned} \tag{4.20}$$

and perturbation kinematics:

$$\begin{aligned}\delta\boldsymbol{\xi}_{VM,k} &\approx \bar{\boldsymbol{\mathcal{T}}}_{\boldsymbol{\varpi},k-1}\delta\boldsymbol{\xi}_{VM,k-1} + \delta\boldsymbol{\xi}_{\boldsymbol{\varpi},k-1} + \mathbf{w}_{VM} \\ &\approx \exp(\Delta t_k \bar{\boldsymbol{\varpi}}_{k-1}^{\curlywedge})\delta\boldsymbol{\xi}_{VM,k-1} + \Delta t_k \boldsymbol{\mathcal{J}}(\Delta t_k \bar{\boldsymbol{\varpi}}_{k-1})\delta\boldsymbol{\varpi}_{k-1} + \mathbf{w}_{VM}. \end{aligned} \tag{4.21}$$

4.2.2 Vehicle Velocity Process Model

The vehicle velocity process model is simply

$$\boldsymbol{\varpi}_k = \boldsymbol{\varpi}_{k-1} + \mathbf{w}_{\varpi}. \tag{4.22}$$

We again follow the white-noise-on-acceleration model, which breaks it down into the following:

$$\text{nominal: } \bar{\boldsymbol{\varpi}}_k = \bar{\boldsymbol{\varpi}}_{k-1}, \tag{4.23}$$

$$\text{perturbation: } \delta\boldsymbol{\varpi}_k = \delta\boldsymbol{\varpi}_{k-1} + \mathbf{w}_{\boldsymbol{\varpi}}, \tag{4.24}$$

where $\mathbf{w}_{\boldsymbol{\varpi}} \in \mathbb{R}^6$ is the process noise.

It is clear that the process noise of vehicle pose and velocity, $\mathbf{w}_{VM}$ and $\mathbf{w}_{\varpi}$, are correlated. Their joint distribution, as formulated in [2], is

$$\begin{bmatrix}\mathbf{w}_{VM} \\ \mathbf{w}_{\varpi}\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}_{12\times 1}, \underbrace{\begin{bmatrix}\frac{1}{3}\Delta t_k^3\mathbf{Q}_C & \frac{1}{2}\Delta t_k^2\mathbf{Q}_C \\ \frac{1}{2}\Delta t_k^2\mathbf{Q}_C & \Delta t_k\mathbf{Q}_C\end{bmatrix}}_{\mathbf{Q}_{VM}}\right), \tag{4.25}$$

where the tunable parameter $\mathbf{Q}_C \in \mathbb{R}^{6\times 6}$ is a diagonal matrix with non-zero values in its first and last diagonal entries corresponding to the vehicle's translational and rotational accelerations in the vehicle frame, which are along the x-axis (tangential to its motion) and about the z-axis (normal to the ground plane), respectively.

4.2.3 GPS Offset Process Model

The GPS-to-map offset, which very gradually varies over time, is modelled as a random walk. This is a convenient way to handle the estimation of such a time-dependent unknown parameter:

$$\mathbf{T}_{GM,k} = \exp(\mathbf{w}_{GM}^{\wedge})\mathbf{T}_{GM,k-1}, \tag{4.26}$$

where $\mathbf{w}_{GM} \sim \mathcal{N}(\mathbf{0}_{6\times1}, \mathbf{Q}_{GM})$ is the process noise. We can break it down into the following nominal and perturbation kinematics:

$$\text{nominal kinematics: } \bar{\mathbf{T}}_{GM,k} = \bar{\mathbf{T}}_{GM,k-1}, \tag{4.27}$$

$$\text{perturbation kinematics: } \delta\boldsymbol{\xi}_{GM,k} = \delta\boldsymbol{\xi}_{GM,k-1} + \mathbf{w}_{GM}. \tag{4.28}$$

4.3 Observation Models

4.3.1 GPS Observation Model

In this work, a GPS measurement refers to a preprocessed quantity that is a three-dimensional transformation matrix $\mathbf{T}_{VG,k}$ with three degrees of freedom each in position and orientation. This is the output of commercial GPS-based localization solutions such as Applanix POS LV, which integrates GPS and IMU information. The observation model of GPS measurement, $\mathbf{T}_{VG,k}$, is

$$\mathbf{T}_{VG,k} = \exp(\mathbf{n}_{VG}^{\wedge})\mathbf{T}_{VM,k}\mathbf{T}_{GM,k}^{-1}, \tag{4.29}$$

where the measurement noise is $\mathbf{n}_{VG} \sim \mathcal{N}(\mathbf{0}_{6\times1}, \mathbf{R}_{VG})$.

Next, we linearize the observation model. Using the perturbation technique, we

can express both side of (4.29) as

$$\begin{aligned}\underbrace{\exp(\delta\boldsymbol{\xi}_{VG,k}^{\wedge})\bar{\mathbf{T}}_{VG,k}}_{\mathbf{T}_{VG,k}} &= \exp(\mathbf{n}_{VG}^{\wedge})\underbrace{\exp(\delta\boldsymbol{\xi}_{VM,k}^{\wedge})\bar{\mathbf{T}}_{VM,k}}_{\mathbf{T}_{VM,k}}\underbrace{[\exp(\delta\boldsymbol{\xi}_{GM,k}^{\wedge})\bar{\mathbf{T}}_{GM,k}]^{-1}}_{\mathbf{T}_{GM,k}} \\ &= \exp(\mathbf{n}_{VG}^{\wedge})\exp(\delta\boldsymbol{\xi}_{VM,k}^{\wedge})\bar{\mathbf{T}}_{VM,k}\bar{\mathbf{T}}_{GM,k}^{-1}\exp(-\delta\boldsymbol{\xi}_{GM,k}^{\wedge}) \\ &= \exp(\mathbf{n}_{VG}^{\wedge})\exp(\delta\boldsymbol{\xi}_{VM,k}^{\wedge}) \\ &\quad \exp((\mathrm{Ad}(\bar{\mathbf{T}}_{VM,k}\bar{\mathbf{T}}_{GM,k}^{-1})(-\delta\boldsymbol{\xi}_{GM,k}))^{\wedge})\bar{\mathbf{T}}_{VM,k}\bar{\mathbf{T}}_{GM,k}^{-1}. \end{aligned} \tag{4.30}$$

If we assume the nominal kinematics to be

$$\bar{\mathbf{T}}_{VG,k} = \bar{\mathbf{T}}_{VM,k}\bar{\mathbf{T}}_{GM,k}^{-1}, \tag{4.31}$$

then we have

$$\exp(\delta\boldsymbol{\xi}_{VG,k}^{\wedge}) = \exp(\mathbf{n}_{VG}^{\wedge})\exp(\delta\boldsymbol{\xi}_{VM,k}^{\wedge})\exp((\mathrm{Ad}(\bar{\mathbf{T}}_{VM,k}\bar{\mathbf{T}}_{GM,k}^{-1})(-\delta\boldsymbol{\xi}_{GM,k}))^{\wedge}). \tag{4.32}$$

Since all $\mathbf{n}_{VG}$, $\delta\boldsymbol{\xi}_{VM,k}$, and $\delta\boldsymbol{\xi}_{GM,k}$ are small, we can obtain the following approximation by applying the BCH formula to (4.32):

$$\begin{aligned}\delta\boldsymbol{\xi}_{VG,k} &\approx \mathbf{n}_{VG} + \delta\boldsymbol{\xi}_{VM,k} + \mathrm{Ad}(\bar{\mathbf{T}}_{VM,k}\bar{\mathbf{T}}_{GM,k}^{-1})(-\delta\boldsymbol{\xi}_{GM,k}) \\ &= \delta\boldsymbol{\xi}_{VM,k} - \mathrm{Ad}(\bar{\mathbf{T}}_{VM,k}\bar{\mathbf{T}}_{GM,k}^{-1})\delta\boldsymbol{\xi}_{GM,k} + \mathbf{n}_{VG} \\ &= \underbrace{\begin{bmatrix}\mathbf{1}_{6\times6} & \mathbf{0}_{6\times6} & -\mathrm{Ad}(\bar{\mathbf{T}}_{VM,k}\bar{\mathbf{T}}_{GM,k}^{-1})\end{bmatrix}}_{\mathbf{G}_{VG,k}}\delta\boldsymbol{\xi}_k + \mathbf{n}_{VG}, \end{aligned} \tag{4.33}$$

where the perturbation terms of the vehicle states and GPS offset are stacked into

$$\delta\boldsymbol{\xi}_k = \begin{bmatrix}\delta\boldsymbol{\xi}_{VM,k} \\ \delta\boldsymbol{\varpi}_k \\ \delta\boldsymbol{\xi}_{GM,k}\end{bmatrix}, \tag{4.34}$$

and $\mathbf{G}_{VG,k}$ is the resulting coefficient matrix of the linearized GPS observation model.

4.3.2 Traffic Light Observation Model

The traffic lights are modelled as point landmarks as described in Section 3.2. The observation model of the j-th traffic light pixel measurement, $\mathbf{p}_{I,k}^{j}$, can be broken down into two steps. First, the corresponding known location in the semantic map, $\mathbf{p}_{M}^{j}$, obtained through data association, is transformed to the camera reference frame:

$$\mathbf{p}_{C,k}^{j} = \mathbf{h}(\mathbf{p}_{M}^{j}, \mathbf{T}_{VM,k}) = \mathbf{D}_{C}\mathbf{T}_{CV}\mathbf{T}_{VM,k}\mathbf{q}_{M}^{j}, \tag{4.35}$$

where $\mathbf{p}_{M}^{j}$ is expressed in 4×1 homogeneous coordinates,

$$\mathbf{q}_{M}^{j} = \begin{bmatrix} \mathbf{p}_{M}^{j} \\ 1 \end{bmatrix}, \tag{4.36}$$

and $\mathbf{T}_{CV} \in SE(3)$ is the known constant transformation between vehicle frame and camera frame. To convert the resulting homogeneous coordinates back to 3×1, the projection matrix

$$\mathbf{D}_{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{4.37}$$

is applied. The function $\mathbf{h}(\cdot)$ summarizes this coordinate transformation.

Next, we project $\mathbf{p}_{C,k}^{j} = \begin{bmatrix} x_{C,k}^{j} & y_{C,k}^{j} & z_{C,k}^{j} \end{bmatrix}^{T}$ to the image space:

$$\mathbf{p}_{I,k}^{j} = \underbrace{\mathbf{D}_{I}\mathbf{K}_{C}\frac{\mathbf{p}_{C,k}^{j}}{z_{C,k}^{j}}}_{\mathbf{g}(\mathbf{p}_{C,k}^{j})} + \mathbf{n}_{\text{light}}, \tag{4.38}$$

where

$$\mathbf{D}_{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{4.39}$$

makes the resulting pixel measurement 2×1, and $\mathbf{K}_{C} \in \mathbb{R}^{3\times3}$ is the known camera intrinsic matrix. The projection is represented by the function $\mathbf{g}(\cdot)$, with the additional pixel measurement noise $\mathbf{n}_{\text{light}} \sim \mathcal{N}(\mathbf{0}_{2\times1}, \mathbf{R}_{\text{light}})$ that is assumed to be Gaussian.

The overall observation model combining (4.35) and (4.38) can be summarized as

$$\begin{aligned}\mathbf{p}_{I,k}^{j} &= \mathbf{g}(\mathbf{h}(\mathbf{p}_{M}^{j}, \mathbf{T}_{VM,k})) + \mathbf{n}_{\text{light}} \\ &= \mathbf{g}(\mathbf{p}_{C,k}^{j}) + \mathbf{n}_{\text{light}}. \end{aligned} \tag{4.40}$$

To linearize the the observation model, we first compute the Jacobian matrix of $\mathbf{p}_{C,k}^{j}$ by substituting

$$\mathbf{T}_{VM,k} = \exp(\delta\boldsymbol{\xi}_{VM,k}^{\wedge})\bar{\mathbf{T}}_{VM,k}, \tag{4.41}$$

into (4.35):

$$\begin{aligned}\mathbf{p}_{C,k}^{j} &= \mathbf{D}_C\mathbf{T}_{CV}(\exp(\delta\boldsymbol{\xi}_{VM,k}^{\wedge})\bar{\mathbf{T}}_{VM,k})\mathbf{q}_{M}^{j} \\ &\approx \mathbf{D}_C\mathbf{T}_{CV}(\mathbf{1} + \delta\boldsymbol{\xi}_{VM,k}^{\wedge})\bar{\mathbf{T}}_{VM,k}\mathbf{q}_{M}^{j} \\ &= \underbrace{\mathbf{D}_C\mathbf{T}_{CV}\bar{\mathbf{T}}_{VM,k}\mathbf{q}_{M}^{j}}_{\bar{\mathbf{p}}_{C,k}^{j}} + \mathbf{D}_C\mathbf{T}_{CV}\delta\boldsymbol{\xi}_{VM,k}^{\wedge}\bar{\mathbf{T}}_{VM,k}\mathbf{q}_{M}^{j} \\ &= \bar{\mathbf{p}}_{C,k}^{j} + \underbrace{\mathbf{D}_C\mathbf{T}_{CV}(\bar{\mathbf{T}}_{VM,k}\mathbf{q}_{M}^{j})^{\odot}}_{\mathbf{Z}_{\text{light},k}^{j}}\delta\boldsymbol{\xi}_{VM,k} \\ &= \bar{\mathbf{p}}_{C,k}^{j} + \mathbf{Z}_{\text{light},k}^{j}\delta\boldsymbol{\xi}_{VM,k}, \end{aligned} \tag{4.42}$$

where the Jacobian matrix is $\mathbf{Z}_{\text{light},k}^{j} = \left.\dfrac{\partial\mathbf{p}_{C,k}^{j}}{\partial\mathbf{T}_{VM,k}}\right|_{\bar{\mathbf{T}}_{VM,k}}$.

Then, using the chain rule, we work out the linearized observation model. Starting with (4.40), we have

$$\begin{aligned}\mathbf{p}_{I,k}^{j} &= \mathbf{g}(\mathbf{p}_{C,k}^{j}) + \mathbf{n}_{\text{light}} \\ &\approx \mathbf{g}(\bar{\mathbf{p}}_{C,k}^{j} + \mathbf{Z}_{\text{light},k}^{j}\delta\boldsymbol{\xi}_{VM,k}) + \mathbf{n}_{\text{light}} \\ &\approx \mathbf{g}(\bar{\mathbf{p}}_{C,k}^{j}) + \mathbf{S}_{\text{light},k}^{j}\mathbf{Z}_{\text{light},k}^{j}\delta\boldsymbol{\xi}_{VM,k} + \mathbf{n}_{\text{light}}, \end{aligned} \tag{4.43}$$

where

$$\mathbf{S}^j_{\text{light},k} = \left.\frac{\partial \mathbf{g}}{\partial \mathbf{p}^j_{C,k}}\right|_{\bar{\mathbf{p}}^j_{C,k}} = \mathbf{D}_I \mathbf{K}_C \left[\begin{array}{ccc} \frac{1}{z^j_{C,k}} & 0 & \frac{-x^j_{C,k}}{(z^j_{C,k})^2} \\ 0 & \frac{1}{z^j_{C,k}} & \frac{-y^j_{C,k}}{(z^j_{C,k})^2} \\ 0 & 0 & 0 \end{array}\right]_{\bar{\mathbf{p}}^j_{C,k}} . \tag{4.44}$$

If we express the left-hand side of (4.43) as a mean and a perturbation,

$$\mathbf{p}^j_{I,k} = \bar{\mathbf{p}}^j_{I,k} + \delta\mathbf{p}^j_{I,k}, \tag{4.45}$$

and subtract off the nominal part,

$$\bar{\mathbf{p}}^j_{I,k} = \mathbf{g}(\bar{\mathbf{p}}^j_{C,k}), \tag{4.46}$$

from both sides of (4.43), then we are left with

$$\begin{aligned} \delta\mathbf{p}^j_{I,k} &\approx \mathbf{S}^j_{\text{light},k}\mathbf{Z}^j_{\text{light},k}\delta\boldsymbol{\xi}_{VM,k} + \mathbf{n}_{\text{light}} \\ &= \left[\mathbf{S}^j_{\text{light},k}\mathbf{Z}^j_{\text{light},k} \quad \mathbf{0}_{2\times 6} \quad \mathbf{0}_{2\times 6}\right]\delta\boldsymbol{\xi}_k + \mathbf{n}_{\text{light}}. \end{aligned} \tag{4.47}$$

Finally, the coefficient matrix of the linearized observation model is

$$\mathbf{G}^j_{\text{light},k} = \left[\mathbf{S}^j_{\text{light},k}\mathbf{Z}^j_{\text{light},k} \quad \mathbf{0}_{2\times 6} \quad \mathbf{0}_{2\times 6}\right]. \tag{4.48}$$

4.3.3 Lane Marking Observation Model

Using the data association process described in Section 3.3, we obtain the lane marking observations as straight lines in the image space. For the j-th observed lane marking, the corresponding lane line in the semantic map is defined by two end points, $\mathbf{p}^j_{M,1}$ and $\mathbf{p}^j_{M,2}$. We start out similarly to the traffic light observation model

and transform the two points to the camera frame:

$$\mathbf{p}_{C,km}^{j} = \begin{bmatrix} x_{C,km}^{j} \\ y_{C,km}^{j} \\ z_{C,km}^{j} \end{bmatrix} = \mathbf{D}_C \mathbf{T}_{CV} \mathbf{T}_{VM,k} \mathbf{q}_{M,m}^{j}, \tag{4.49}$$

where $m = 1, 2$, and $\mathbf{p}_{M,m}^{j}$ is expressed in 4×1 homogeneous coordinates,

$$\mathbf{q}_{M,m}^{j} = \begin{bmatrix} \mathbf{p}_{M,m}^{j} \\ 1 \end{bmatrix}. \tag{4.50}$$

Then, they are projected to the camera image space:

$$\mathbf{p}_{I,km}^{j} = \begin{bmatrix} x_{I,km}^{j} \\ y_{I,km}^{j} \end{bmatrix} = \mathbf{D}_I \mathbf{K}_C \frac{\mathbf{p}_{C,km}^{j}}{z_{C,km}^{j}}. \tag{4.51}$$

Because the lane markings run mostly parallel to the vehicle trajectory, they offer information largely in the lateral direction of the vehicle as opposed to longitudinal direction. Therefore, we define the observation model to be the horizontal component of two pixel coordinates that are on the lane line in the image space, $\mathbf{x}_k^j = \begin{bmatrix} x_{1,k}^j & x_{2,k}^j \end{bmatrix}^T$, which contain no longitudinal information. The pixels are selected to correspond with two different vertical pixel coordinates, $\mathbf{y}^j = \begin{bmatrix} y_1^j & y_2^j \end{bmatrix}^T$, which we can use to solve for $\mathbf{x}_k^j$:

$$\mathbf{x}_k^j = \begin{bmatrix} x_{I,k1}^j \\ x_{I,k1}^j \end{bmatrix} + \frac{x_{I,k2}^j - x_{I,k1}^j}{y_{I,k2}^j - y_{I,k1}^j} \begin{bmatrix} y_1^j - y_{I,k1}^j \\ y_2^j - y_{I,k1}^j \end{bmatrix} + \mathbf{n}_{\text{lane}}, \tag{4.52}$$

where the Gaussian measurement noise is $\mathbf{n}_{\text{lane}} \sim \mathcal{N}\left(\mathbf{0}_{2\times 1}, \mathbf{R}_{\text{lane}}\right)$.

The overall observation model combining (4.49), (4.51), and (4.52) can be summarized as

$$\mathbf{x}_k^j = \begin{bmatrix} x_{1,k}^j \\ x_{2,k}^j \end{bmatrix} = \underbrace{\begin{bmatrix} f(\boldsymbol{\ell}_M^j, \mathbf{T}_{VM,k}, y_1^j) \\ f(\boldsymbol{\ell}_M^j, \mathbf{T}_{VM,k}, y_2^j) \end{bmatrix}}_{\mathbf{f}(\boldsymbol{\ell}_M^j, \mathbf{T}_{VM,k}, \mathbf{y}^j)} + \mathbf{n}_{\text{lane}}, \tag{4.53}$$

where $\boldsymbol{\ell}_M^j = \{\mathbf{p}_{M,1}^j, \mathbf{p}_{M,2}^j\}$ is the known lane line from the semantic map, and $f(\cdot)$

projects the lane line, $\boldsymbol{\ell}_M^j$, from semantic map to camera image space given vehicle pose estimation, then computes the horizontal pixel coordinates given y_m^j.

Because the linearization process is identical for each $x_{m,k}^j$, $m = 1, 2$, in $\mathbf{x}_k^j$ of the observation model, we only need to do the derivation once. From (4.53), we know that $x_{m,k}^j$ is a function of the vehicle pose, $\mathbf{T}_{VM,k}$, so it can be expressed as

$$\begin{aligned} x_{m,k}^j &= f(\underbrace{\mathbf{T}_{VM,k}}_{\exp(\delta\boldsymbol{\xi}_{VM,k}^{\wedge})\bar{\mathbf{T}}_{VM,k}}) + n_{\text{lane},m} \\ &\approx f(\bar{\mathbf{T}}_{VM,k}) + \mathbf{G}_{\text{lane},km}^j \delta\boldsymbol{\xi}_{VM,k} + n_{\text{lane},m}, \end{aligned} \tag{4.54}$$

where $n_{\text{lane},m}$ is the component of the measurement noise term $\mathbf{n}_{\text{lane}}$.

The coefficient matrix of the linearized observation model, $\mathbf{G}_{\text{lane},km}^j$, can be expanded using the chain rule as follows:

$$\begin{aligned} \mathbf{G}_{\text{lane},km}^j &= \frac{\partial x_{m,k}^j}{\partial \mathbf{T}_{VM,k}}\bigg|_{\bar{\mathbf{T}}_{VM,k}} \\ &= \left(\frac{\partial x_{m,k}^j}{\partial \mathbf{p}_{I,k1}^j} \frac{\partial \mathbf{p}_{I,k1}^j}{\partial \mathbf{p}_{C,k1}^j} \frac{\partial \mathbf{p}_{C,k1}^j}{\partial \mathbf{T}_{VM,k}} + \frac{\partial x_{m,k}^j}{\partial \mathbf{p}_{I,k2}^j} \frac{\partial \mathbf{p}_{I,k2}^j}{\partial \mathbf{p}_{C,k2}^j} \frac{\partial \mathbf{p}_{C,k2}^j}{\partial \mathbf{T}_{VM,k}} \right) \bigg|_{\bar{\mathbf{T}}_{VM,k}}. \end{aligned} \tag{4.55}$$

We now need to derive each term in this expression. The Jacobian matrices, $\dfrac{\partial x_{m,k}^j}{\partial \mathbf{p}_{I,k1}^j}$ and $\dfrac{\partial x_{m,k}^j}{\partial \mathbf{p}_{I,k2}^j}$, can be computed from equation (4.52):

$$\frac{\partial x_{m,k}^j}{\partial \mathbf{p}_{I,k1}^j} = \begin{bmatrix} \left(1 - \dfrac{y_m^j - y_{I,k1}^j}{y_{I,k2}^j - y_{I,k1}^j}\right) & \dfrac{(x_{I,k2}^j - x_{I,k1}^j)[(y_m^j - y_{I,k1}^j) - (y_{I,k2}^j - y_{I,k1}^j)]}{(y_{I,k2}^j - y_{I,k1}^j)^2} \end{bmatrix}, \tag{4.56}$$

$$\frac{\partial x_{m,k}^j}{\partial \mathbf{p}_{I,k2}^j} = \begin{bmatrix} \dfrac{y_m^j - y_{I,k1}^j}{y_{I,k2}^j - y_{I,k1}^j} & \dfrac{-(x_{I,k2}^j - x_{I,k1}^j)(y_m^j - y_{I,k1}^j)}{(y_{I,k2}^j - y_{I,k1}^j)^2} \end{bmatrix}, \tag{4.57}$$

and $\dfrac{\partial \mathbf{p}^j_{I,km}}{\partial \mathbf{p}^j_{C,km}}$ is the same as in (4.44):

$$\frac{\partial \mathbf{p}^j_{I,km}}{\partial \mathbf{p}^j_{C,km}} = \mathbf{D}_I \mathbf{K}_C \begin{bmatrix} \dfrac{1}{z^j_{C,km}} & 0 & \dfrac{-x^j_{C,km}}{(z^j_{C,km})^2} \\ 0 & \dfrac{1}{z^j_{C,km}} & \dfrac{-y^j_{C,km}}{(z^j_{C,km})^2} \\ 0 & 0 & 0 \end{bmatrix}. \tag{4.58}$$

For $\dfrac{\partial \mathbf{p}^j_{C,km}}{\partial \mathbf{T}_{VM,k}}$, we start with (4.49) and use the same trick as in (4.42):

$$\begin{aligned} \mathbf{p}^j_{C,km} &= \mathbf{D}_C \mathbf{T}_{CV} (\exp(\delta \boldsymbol{\xi}^\wedge_{VM,k}) \bar{\mathbf{T}}_{VM,k}) \mathbf{q}^j_{M,m} \\ &\approx \mathbf{D}_C \mathbf{T}_{CV} (1 + \delta \boldsymbol{\xi}^\wedge_{VM,k}) \bar{\mathbf{T}}_{VM,k} \mathbf{q}^j_{M,m} \\ &= \underbrace{\mathbf{D}_C \mathbf{T}_{CV} \bar{\mathbf{T}}_{VM,k} \mathbf{q}^j_{M,m}}_{\bar{\mathbf{p}}^j_{C,km}} + \underbrace{\mathbf{D}_C \mathbf{T}_{CV} (\bar{\mathbf{T}}_{VM,k} \mathbf{q}^j_{M,m})^\odot}_{\mathbf{Z}^j_{\text{lane},km}} \delta \boldsymbol{\xi}_{VM,k} \\ &= \bar{\mathbf{p}}^j_{C,km} + \mathbf{Z}^j_{\text{lane},km} \delta \boldsymbol{\xi}_{VM,k}, \end{aligned} \tag{4.59}$$

where $\mathbf{Z}^j_{\text{lane},km} = \dfrac{\partial \mathbf{p}^j_{C,km}}{\partial \mathbf{T}_{VM,k}}$ is the desired Jacobian matrix.

Lastly, if we express the left-hand side of (4.54) as a mean and a perturbation,

$$x^n_{km} = \bar{x}^j_{m,k} + \delta x^j_{m,k}, \tag{4.60}$$

and subtract off the nominal solution,

$$\bar{x}^j_{m,k} = f(\bar{\mathbf{T}}_{VM,k}), \tag{4.61}$$

from both sides of (4.54), then we are left with

$$\begin{aligned} \delta x^n_{km} &\approx \mathbf{G}^j_{\text{lane},km} \delta \boldsymbol{\xi}_{VM,k} + n_{\text{lane},m} \\ &= \begin{bmatrix} \mathbf{G}^j_{\text{lane},km} & \mathbf{0}_{1\times 6} & \mathbf{0}_{1\times 6} \end{bmatrix} \delta \boldsymbol{\xi}_k + n_{\text{lane},m}. \end{aligned} \tag{4.62}$$

To obtain the coefficient matrix of the linearized observation model for $\mathbf{x}_k^j$, which consists of two horizontal pixel measurements, we simply stack them:

$$\mathbf{G}_{\text{lane},k}^j = \begin{bmatrix} \mathbf{G}_{\text{lane},k1}^j & \mathbf{0}_{1\times 6} & \mathbf{0}_{1\times 6} \\ \mathbf{G}_{\text{lane},k2}^j & \mathbf{0}_{1\times 6} & \mathbf{0}_{1\times 6} \end{bmatrix}. \tag{4.63}$$

4.3.4 Wheel Encoders Observation Model

The onboard wheel encoders provide measurements on vehicle velocities. Specifically, it measures the vehicle's longitudinal velocity, v_k, and angular velocity, ω_k, in yaw. Wheel encoders are included to improve robustness against GPS dropouts. The observation model is

$$\boldsymbol{\varpi}_{\text{wheel},k} = \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} = \mathbf{D}_{\varpi}\boldsymbol{\varpi}_k + \mathbf{n}_{\varpi}, \tag{4.64}$$

where

$$\mathbf{D}_{\varpi} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.65}$$

extracts the corresponding vehicle velocities from $\boldsymbol{\varpi}_k$, and the Gaussian noise term is $\mathbf{n}_{\varpi} \sim \mathcal{N}(\mathbf{0}_{2\times 1}, \mathbf{R}_{\varpi})$.

We do not need to linearize the observation model because it is already linear. Therefore, the derivation of the coefficient matrix is straightforward by substituting the following two expressions,

$$\boldsymbol{\varpi}_{\text{wheel},k} = \bar{\boldsymbol{\varpi}}_{\text{wheel},k} + \delta\boldsymbol{\varpi}_{\text{wheel},k}, \tag{4.66}$$

$$\boldsymbol{\varpi}_k = \bar{\boldsymbol{\varpi}}_k + \delta\boldsymbol{\varpi}_k, \tag{4.67}$$

into the observation model (4.64):

$$\bar{\boldsymbol{\varpi}}_{\text{wheel},k} + \delta\boldsymbol{\varpi}_{\text{wheel},k} = \mathbf{D}_{\varpi}(\bar{\boldsymbol{\varpi}}_k + \delta\boldsymbol{\varpi}_k) + \mathbf{n}_{\varpi}. \tag{4.68}$$

After subtracting the nominal kinematics,

$$\bar{\boldsymbol{\varpi}}_{\text{wheel},k} = \mathbf{D}_{\varpi}\bar{\boldsymbol{\varpi}}_k, \tag{4.69}$$

from both sides of the equation, the following remains:

$$\begin{aligned}\delta\boldsymbol{\varpi}_{\text{wheel},k} &= \mathbf{D}_{\varpi}\delta\boldsymbol{\varpi}_k + \mathbf{n}_{\varpi} \\ &= \underbrace{\begin{bmatrix}\mathbf{0}_{2\times 6} & \mathbf{D}_{\varpi} & \mathbf{0}_{2\times 6}\end{bmatrix}}_{\mathbf{G}_{\varpi}} \delta\boldsymbol{\xi}_k + \mathbf{n}_{\varpi},\end{aligned} \tag{4.70}$$

where $\mathbf{G}_{\varpi}$ is the constant coefficient matrix of the wheel encoder observation model.

4.3.5 Pseudo-Measurement Observation Model

In addition to the aforementioned sensor observations, some pseudo-measurements are also introduced by leveraging the physical constraints of the vehicle, including the vehicle pose and velocity.

Vehicle Elevation

We take advantage of the fact that the vehicle always stays on the ground, which is assumed to be the xy-plane of the map frame. Therefore, its elevation is soft-constrained to zero with respect to the semantic map frame. This effectively reduces the localization problem down to a 2D space while maintaining the problem formulation in 3D.

For the vehicle elevation, we simply have

$$z_{\text{pseudo},k} = \mathbf{D}_{z1}\mathbf{T}_{VM,k}^{-1}\mathbf{D}_{z2} + n_z, \tag{4.71}$$

where

$$\mathbf{D}_{z1} = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix}, \tag{4.72}$$

and

$$\mathbf{D}_{z2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{4.73}$$

collectively extract the vehicle elevation from $\mathbf{T}_{VM,k}$. The pseudo-measurement, $z_{\text{pseudo},k}$, is always zero, and the Gaussian noise term is $n_z \sim \mathcal{N}(0, r_z)$, with a small r_z to effectively constraint the vehicle elevation.

As usual, we linearize the observation model utilizing the technique of perturbation and substitute

$$z_{\text{pseudo},k} = \bar{z}_{\text{pseudo},k} + \delta z_{\text{pseudo},k}, \tag{4.74}$$

$$\mathbf{T}_{VM,k} = \exp(\delta \boldsymbol{\xi}^{\wedge}_{VM,k}) \bar{\mathbf{T}}_{VM,k}, \tag{4.75}$$

into (4.71):

$$\begin{aligned} \bar{z}_{\text{pseudo},k} + \delta z_{\text{pseudo},k} &= \mathbf{D}_{z1}(\exp(\delta \boldsymbol{\xi}^{\wedge}_{VM,k}) \bar{\mathbf{T}}_{VM,k})^{-1} \mathbf{D}_{z2} + n_z \\ &= \mathbf{D}_{z1} \bar{\mathbf{T}}^{-1}_{VM,k} \exp(-\delta \boldsymbol{\xi}^{\wedge}_{VM,k}) \mathbf{D}_{z2} + n_z \\ &\approx \mathbf{D}_{z1} \bar{\mathbf{T}}^{-1}_{VM,k} (\mathbf{1} - \delta \boldsymbol{\xi}^{\wedge}_{VM,k}) \mathbf{D}_{z2} + n_z \\ &= \underbrace{\mathbf{D}_{z1} \bar{\mathbf{T}}^{-1}_{VM,k} \mathbf{D}_{z2}}_{\bar{z}_{\text{pseudo},k}} - \mathbf{D}_{z1} \bar{\mathbf{T}}^{-1}_{VM,k} \delta \boldsymbol{\xi}^{\wedge}_{VM,k} \mathbf{D}_{z2} + n_z. \end{aligned} \tag{4.76}$$

The perturbation part is

$$\begin{aligned} \delta z_{\text{pseudo},k} &\approx -\mathbf{D}_{z1} \bar{\mathbf{T}}^{-1}_{VM,k} \delta \boldsymbol{\xi}^{\wedge}_{VM,k} \mathbf{D}_{z2} + n_z \\ &= -\mathbf{D}_{z1} \bar{\mathbf{T}}^{-1}_{VM,k} \mathbf{D}^{\odot}_{z2} \delta \boldsymbol{\xi}_{VM,k} + n_z \\ &= \underbrace{\begin{bmatrix} -\mathbf{D}_{z1} \bar{\mathbf{T}}^{-1}_{VM,k} \mathbf{D}^{\odot}_{z2} & \mathbf{0}_{1\times 12} \end{bmatrix}}_{\mathbf{G}_{z,k}} \delta \boldsymbol{\xi}_k + n_z, \end{aligned} \tag{4.77}$$

and we obtain the coefficient matrix $\mathbf{G}_{z,k}$ for vehicle elevation observation model.

Vehicle Orientation

In addition to zero vehicle elevation, the assumption also implies that the vehicle's roll and pitch are near zero with respect to the semantic map frame.

The observation model of the roll and pitch is a little more complicated. We know that the vehicle orientation, $\mathbf{C}_{VM,k} \in SO(3)$, which is part of the current vehicle pose estimation, $\mathbf{T}_{VM,k}$, contains the information of roll, pitch, and yaw. Rather than complicating the mathematical formulation in an attempt to isolate roll and pitch for the pseudo-measurement observation model, we instead keep $\mathbf{C}_{VM,k}$ intact and have the following observation model:

$$\mathbf{C}_{\text{pseudo},k} = \exp(\mathbf{n}_C^{\wedge})\mathbf{C}_{VM,k}^{-1}, \tag{4.78}$$

where the Gaussian noise term is

$$\mathbf{n}_C \sim \mathcal{N}\left(\mathbf{0}_{3\times 1}, \mathbf{R}_C\right), \mathbf{R}_C = \begin{bmatrix} r_{c1} & & \\ & r_{c2} & \\ & & r_{c3} \end{bmatrix}. \tag{4.79}$$

In order to have its roll and pitch be zero, the pseudo-measurement, $\mathbf{C}_{\text{pseudo},k}$, takes the form of a rotation matrix that rotates about the z-axis in the semantic map frame:

$$\mathbf{C}_{\text{pseudo},k} = \begin{bmatrix} \cos\theta_k & -\sin\theta_k & 0 \\ \sin\theta_k & \cos\theta_k & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.80}$$

where θ_k is the vehicle yaw extracted from the current vehicle orientation estimation expressed in the map frame, $\mathbf{C}_{VM,k}^{-1} = \mathbf{C}_{VM,k}^{T}$.

Once again, we express the pseudo-measurement and the current vehicle orientation estimation in terms of means and perturbations:

$$\mathbf{C}_{\text{pseudo},k} = \exp(\delta\boldsymbol{\zeta}_{\text{pseudo},k}^{\wedge})\bar{\mathbf{C}}_{\text{pseudo},k}, \tag{4.81}$$

$$\mathbf{C}_{VM,k} = \exp(\delta\boldsymbol{\zeta}_{VM,k}^{\wedge})\bar{\mathbf{C}}_{VM,k}, \tag{4.82}$$

where the perturbation terms are $\delta\boldsymbol{\zeta}_{\text{pseudo},k}, \delta\boldsymbol{\zeta}_{VM,k} \in \mathbb{R}^3$, and $\delta\boldsymbol{\zeta}_{VM,k}$ is the last three elements of $\delta\boldsymbol{\xi}_{VM,k} \in \mathbb{R}^6$. We substitute these two expressions into the observation model (4.78):

$$\begin{aligned}\exp(\delta\boldsymbol{\zeta}^{\wedge}_{\text{pseudo},k})\bar{\mathbf{C}}_{\text{pseudo},k} &= \exp(\mathbf{n}_C^{\wedge})[\exp(\delta\boldsymbol{\zeta}^{\wedge}_{VM,k})\bar{\mathbf{C}}_{VM,k}]^{-1} \\ &= \exp(\mathbf{n}_C^{\wedge})\bar{\mathbf{C}}^{-1}_{VM,k}\exp(-\delta\boldsymbol{\zeta}^{\wedge}_{VM,k}) \\ &= \exp(\mathbf{n}_C^{\wedge})\exp((-\bar{\mathbf{C}}^{-1}_{VM,k}\delta\boldsymbol{\zeta}_{VM,k})^{\wedge})\bar{\mathbf{C}}^{-1}_{VM,k}, \end{aligned} \tag{4.83}$$

and assume the nominal kinematics to be

$$\bar{\mathbf{C}}_{\text{pseudo},k} = \bar{\mathbf{C}}^{-1}_{VM,k}, \tag{4.84}$$

then the remaining perturbation terms are

$$\exp(\delta\boldsymbol{\zeta}^{\wedge}_{\text{pseudo},k}) = \exp(\mathbf{n}_C^{\wedge})\exp((-\bar{\mathbf{C}}^{-1}_{VM,k}\delta\boldsymbol{\zeta}_{VM,k})^{\wedge}). \tag{4.85}$$

Since both $\mathbf{n}_C$ and $\delta\boldsymbol{\zeta}_{VM,k}$ are small, we can obtain the following approximation using the BCH formula:

$$\begin{aligned}\delta\boldsymbol{\zeta}_{\text{pseudo},k} &\approx \mathbf{n}_C - \bar{\mathbf{C}}^{-1}_{VM,k}\delta\boldsymbol{\zeta}_{VM,k} \\ &= -\bar{\mathbf{C}}^{-1}_{VM,k}\delta\boldsymbol{\zeta}_{VM,k} + \mathbf{n}_C \\ &= \underbrace{\begin{bmatrix}\mathbf{0}_{3\times 3} & -\bar{\mathbf{C}}^{-1}_{VM,k} & \mathbf{0}_{3\times 12}\end{bmatrix}}_{\mathbf{G}_{C,k}}\delta\boldsymbol{\xi}_k + \mathbf{n}_C, \end{aligned} \tag{4.86}$$

where $\mathbf{G}_{C,k}$ is the coefficient matrix of the linearized observation model.

Recall that in order to keep the observation model (4.78) simple, the vehicle yaw, which should not be part of the pseudo-measurements, has been included in the formulation. Fortunately, the yaw component can be easily discarded by removing the last row of $\mathbf{G}_{C,k}$. We will denote the remaining part of the coefficient matrix as $\mathbf{G}'_{C,k} = \mathbf{D}_I\mathbf{G}_{C,k}$. For the same reason, the covariance component, r_{c3}, which corresponds to the vehicle yaw, is merely a dummy variable, and the remaining covariance matrix is

$$\mathbf{R}'_C = \begin{bmatrix} r_{c1} & \\ & r_{c2} \end{bmatrix}. \tag{4.87}$$

Vehicle Velocity

Besides the vehicle pose, we also assume that the rear wheels do not slip sideways, thus the lateral vehicle velocity is also near zero. The observation model is straightforward:

$$v_{\text{pseudo},k} = \mathbf{D}_v \boldsymbol{\varpi}_k + n_v, \tag{4.88}$$

where

$$\mathbf{D}_v = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.89}$$

extracts the lateral vehicle velocity from $\boldsymbol{\varpi}_k$. Again, the pseudo-measurement, $v_{\text{pseudo},k}$, is always zero, and the Gaussian noise term is $n_v \sim \mathcal{N}(0, r_v)$, with a small r_v to constraint the lateral velocity.

Similar to the wheel encoders, the observation model is linear. So we can go through the same derivation and obtain the coefficient matrix:

$$\mathbf{G}_v = \begin{bmatrix} \mathbf{0}_{1\times 6} & \mathbf{D}_v & \mathbf{0}_{1\times 6} \end{bmatrix}. \tag{4.90}$$

4.4 Modified IEKF Formulation

4.4.1 Prediction Step

The prediction step follows the standard IEKF formulation by jointly estimating the vehicle pose, velocity, and GPS offset. Using the derived nominal kinematics models (4.20), (4.23) and (4.27), we have the linearized predicted means:

$$\check{\mathbf{T}}_{VM,k} = \exp(\Delta t_k \hat{\boldsymbol{\varpi}}_{k-1}^{\wedge}) \hat{\mathbf{T}}_{VM,k-1}, \tag{4.91}$$

$$\check{\boldsymbol{\varpi}}_k = \hat{\boldsymbol{\varpi}}_{k-1}, \tag{4.92}$$

$$\check{\mathbf{T}}_{GM,k} = \hat{\mathbf{T}}_{GM,k-1}. \tag{4.93}$$

Because the states are correlated, we combine the associated perturbation terms, $\delta\check{\boldsymbol{\xi}}_{VM,k}$, $\delta\check{\boldsymbol{\varpi}}_k$, and $\delta\check{\boldsymbol{\xi}}_{GM,k}$, into one perturbation term, $\delta\check{\boldsymbol{\xi}}_k \in \mathbb{R}^{18}$, then apply the perturbation kinematics models (4.21), (4.24) and (4.28):

$$
\begin{aligned}
\delta\check{\boldsymbol{\xi}}_k &= \begin{bmatrix} \delta\check{\boldsymbol{\xi}}_{VM,k} \\ \delta\check{\boldsymbol{\varpi}}_k \\ \delta\check{\boldsymbol{\xi}}_{GM,k} \end{bmatrix} \\
&= \begin{bmatrix} \exp(\Delta t_k \hat{\boldsymbol{\varpi}}_{k-1}^{\curlywedge})\delta\hat{\boldsymbol{\xi}}_{VM,k-1} + \Delta t_k \boldsymbol{\mathcal{J}}(\Delta t_k \hat{\boldsymbol{\varpi}}_{k-1})\delta\hat{\boldsymbol{\varpi}}_{k-1} + \mathbf{w}_{VM,k} \\ \delta\hat{\boldsymbol{\varpi}}_{k-1} + \mathbf{w}_{\boldsymbol{\varpi},k} \\ \delta\hat{\boldsymbol{\xi}}_{GM,k-1} + \mathbf{w}_{GM,k} \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} \exp(\Delta t_k \hat{\boldsymbol{\varpi}}_{k-1}^{\curlywedge}) & \Delta t_k \boldsymbol{\mathcal{J}}(\Delta t_k \hat{\boldsymbol{\varpi}}_{k-1}) & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}}_{\mathbf{F}_{k-1}} \underbrace{\begin{bmatrix} \delta\hat{\boldsymbol{\xi}}_{VM,k-1} \\ \delta\hat{\boldsymbol{\varpi}}_{k-1} \\ \delta\hat{\boldsymbol{\xi}}_{GM,k-1} \end{bmatrix}}_{\delta\hat{\boldsymbol{\xi}}_{k-1}} + \underbrace{\begin{bmatrix} \mathbf{w}_{VM,k} \\ \mathbf{w}_{\boldsymbol{\varpi},k} \\ \mathbf{w}_{GM,k} \end{bmatrix}}_{\mathbf{w}_k} \\
&= \mathbf{F}_{k-1}\delta\hat{\boldsymbol{\xi}}_{k-1} + \mathbf{w}_k,
\end{aligned} \tag{4.94}
$$

where $\mathbf{F}_{k-1}$ is the combined Jacobian matrix of the linearized process models (4.8), (4.22), and (4.26) at time step $k-1$, and the combined process noise term is

$$
\mathbf{w}_k \sim \mathcal{N}\left(\mathbf{0}_{18\times 1}, \underbrace{\begin{bmatrix} \mathbf{Q}_{VM} & \mathbf{0}_{12\times 6} \\ \mathbf{0}_{6\times 12} & \mathbf{Q}_{GM} \end{bmatrix}}_{\mathbf{Q}_k} \right). \tag{4.95}
$$

Lastly, the predicted joint covariance matrix is

$$
\check{\mathbf{P}}_k = E\left[\delta\check{\boldsymbol{\xi}}_k \delta\check{\boldsymbol{\xi}}_k^T\right] = \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}{}^T + \mathbf{Q}_k. \tag{4.96}
$$

4.4.2 Correction Step

The iterative correction step of the IEKF is modified by replacing it with a batch optimization formulation with time window size of one (the current time step) [2].

The cost function to optimize is $J = J_v + J_y$, where

$$J_v = \frac{1}{2}\mathbf{e}_{v,k}{}^T\check{\mathbf{P}}_k^{-1}\mathbf{e}_{v,k}, \tag{4.97}$$

$$J_y = \sum_i \frac{1}{2}\mathbf{e}_{y,k}^i{}^T\mathbf{R}^{i-1}\mathbf{e}_{y,k}^i, \tag{4.98}$$

are the prior and measurement cost terms, respectively.

Prior Cost Term

The prior errors, $\mathbf{e}_{v,k}$, are computed using the predicted means, $\check{\mathbf{T}}_{VM,k}$, $\check{\boldsymbol{\varpi}}_k$, and $\check{\mathbf{T}}_{GM,k}$, from the prediction step (4.91), (4.92), and (4.93). This encourages a consistent trajectory that respects the vehicle dynamics between the estimated poses. The prior errors are

$$\mathbf{e}_{v,k} = \begin{bmatrix} \ln(\mathbf{T}_{VM,k}\check{\mathbf{T}}_{VM,k}^{-1})^\vee \\ \boldsymbol{\varpi}_k - \check{\boldsymbol{\varpi}}_k \\ \ln(\mathbf{T}_{GM,k}\check{\mathbf{T}}_{GM,k}^{-1})^\vee \end{bmatrix}, \tag{4.99}$$

at the first time step, and

$$\mathbf{e}_{v,k} = \begin{bmatrix} \ln(\mathbf{T}_{VM,k}\hat{\mathbf{T}}_{VM,k-1}^{-1})^\vee - \Delta t_k\hat{\boldsymbol{\varpi}}_{k-1} \\ \boldsymbol{\mathcal{J}}(\ln(\mathbf{T}_{VM,k}\hat{\mathbf{T}}_{VM,k-1}^{-1})^\vee)^{-1}\boldsymbol{\varpi}_k - \hat{\boldsymbol{\varpi}}_{k-1} \\ \ln(\mathbf{T}_{GM,k}\hat{\mathbf{T}}_{GM,k-1}^{-1})^\vee \end{bmatrix}, \tag{4.100}$$

for the rest of the trajectory [2]. Note that we only update the estimations at the current time step k, so the quantities at time step $k-1$ in the prior errors are considered constants. Their values are taken from the results of the correction step at time step $k-1$.

Using the prior errors expressed above and the covariance matrix, $\check{\mathbf{P}}_k$, obtained from the prediction step, we can construct the prior cost term J_v as shown in (4.97).

Next, we seek to linearize the prior errors (4.99) and (4.100) about the operating point, $\mathbf{x}_{\text{op}} = \{\hat{\mathbf{T}}_{VM,\text{op},k},\ \hat{\boldsymbol{\varpi}}_{\text{op},k},\ \hat{\mathbf{T}}_{GM,\text{op},k}\}$. Using BCH approximation and perturbation, the inverse exponential mapping of a transformation matrix $\mathbf{T} = \exp(\delta\boldsymbol{\xi})\bar{\mathbf{T}}_{\text{op}}$

can be approximated as

$$\ln(\mathbf{T})^{\vee} \approx \ln(\bar{\mathbf{T}}_{\mathrm{op}})^{\vee} + \boldsymbol{\mathcal{J}}(\ln(\bar{\mathbf{T}}_{\mathrm{op}})^{\vee})^{-1}\delta\boldsymbol{\xi}, \tag{4.101}$$

which leads us to the following approximation:

$$\begin{aligned}
&\boldsymbol{\mathcal{J}}(\ln(\mathbf{T}_{VM,k}\hat{\mathbf{T}}_{VM,k-1}^{-1})^{\vee})^{-1}\boldsymbol{\varpi}_k \\
&\quad\approx \boldsymbol{\mathcal{J}}(\ln(\hat{\mathbf{T}}_{\mathrm{op},k,k-1})^{\vee} + \boldsymbol{\mathcal{J}}(\ln(\hat{\mathbf{T}}_{\mathrm{op},k,k-1})^{\vee})^{-1}\delta\boldsymbol{\xi}_{VM,k})^{-1}(\hat{\boldsymbol{\varpi}}_{\mathrm{op},k} + \delta\boldsymbol{\varpi}_k) \\
&\quad\approx \boldsymbol{\mathcal{J}}_{k,k-1}^{-1}\hat{\boldsymbol{\varpi}}_{\mathrm{op},k} - \frac{1}{2}(\boldsymbol{\mathcal{J}}_{k,k-1}^{-1}\delta\boldsymbol{\xi}_{VM,k})^{\curlywedge}\hat{\boldsymbol{\varpi}}_{\mathrm{op},k} + \boldsymbol{\mathcal{J}}_{k,k-1}^{-1}\delta\boldsymbol{\varpi}_k \\
&\quad= \boldsymbol{\mathcal{J}}_{k,k-1}^{-1}\hat{\boldsymbol{\varpi}}_{\mathrm{op},k} + \frac{1}{2}\hat{\boldsymbol{\varpi}}_{\mathrm{op},k}^{\curlywedge}\boldsymbol{\mathcal{J}}_{k,k-1}^{-1}\delta\boldsymbol{\xi}_{VM,k} + \boldsymbol{\mathcal{J}}_{k,k-1}^{-1}\delta\boldsymbol{\varpi}_k,
\end{aligned} \tag{4.102}$$

where $\hat{\mathbf{T}}_{\mathrm{op},k,k-1} = \hat{\mathbf{T}}_{VM,\mathrm{op},k}\hat{\mathbf{T}}_{VM,k-1}^{-1}$ and $\boldsymbol{\mathcal{J}}_{k,k-1} = \boldsymbol{\mathcal{J}}(\ln(\hat{\mathbf{T}}_{\mathrm{op},k,k-1})^{\vee})$.

Finally, we derive the linearized prior errors at the first time step:

$$\mathbf{e}_{v,k} \approx \begin{bmatrix} \ln(\hat{\mathbf{T}}_{VM,\mathrm{op},k}\check{\mathbf{T}}_{VM,k}^{-1})^{\vee} \\ \hat{\boldsymbol{\varpi}}_{\mathrm{op},k} - \check{\boldsymbol{\varpi}}_k \\ \ln(\hat{\mathbf{T}}_{GM,\mathrm{op},k}\check{\mathbf{T}}_{GM,k}^{-1})^{\vee} \end{bmatrix} - \mathbf{E}_k\delta\mathbf{x}, \tag{4.103}$$

where

$$\mathbf{E}_k = \begin{bmatrix} -\boldsymbol{\mathcal{J}}_{k,k-1}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{\mathcal{J}}(\ln(\hat{\mathbf{T}}_{GM,\mathrm{op},k}\hat{\mathbf{T}}_{GM,k-1}^{-1})^{\vee})^{-1} \end{bmatrix}. \tag{4.104}$$

For the rest of the time steps,

$$\mathbf{e}_{v,k} \approx \begin{bmatrix} \ln(\hat{\mathbf{T}}_{VM,\mathrm{op},k}\hat{\mathbf{T}}_{VM,k-1}^{-1})^{\vee} - \Delta t_k\hat{\boldsymbol{\varpi}}_{k-1} \\ \boldsymbol{\mathcal{J}}(\ln(\hat{\mathbf{T}}_{VM,\mathrm{op},k}\hat{\mathbf{T}}_{VM,k-1}^{-1})^{\vee})^{-1}\hat{\boldsymbol{\varpi}}_{\mathrm{op},k} - \hat{\boldsymbol{\varpi}}_{k-1} \\ \ln(\hat{\mathbf{T}}_{GM,\mathrm{op},k}\hat{\mathbf{T}}_{GM,k-1}^{-1})^{\vee} \end{bmatrix} - \mathbf{E}_k\delta\mathbf{x}, \tag{4.105}$$

where

$$\mathbf{E}_k = \begin{bmatrix} -\boldsymbol{\mathcal{J}}_{k,k-1}^{-1} & \mathbf{0} & \mathbf{0} \\ -\frac{1}{2}\hat{\boldsymbol{\varpi}}_{\mathrm{op},k}^{\curlywedge}\boldsymbol{\mathcal{J}}_{k,k-1}^{-1} & -\boldsymbol{\mathcal{J}}_{k,k-1}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{\mathcal{J}}(\ln(\hat{\mathbf{T}}_{GM,\mathrm{op},k}\hat{\mathbf{T}}_{GM,k-1}^{-1})^{\vee})^{-1} \end{bmatrix}. \tag{4.106}$$

The perturbation term is

$$\delta\mathbf{x} = \begin{bmatrix} \delta\boldsymbol{\xi}_{VM,k} \\ \delta\boldsymbol{\varpi}_k \\ \delta\boldsymbol{\xi}_{GM,k} \end{bmatrix}. \tag{4.107}$$

Measurement Cost Term

The overall measurement cost term J_y is a sum of the cost terms derived from the sensor measurements, including GPS, semantic cues, wheel encoders, and pseudo-measurements. For the i-th measurement, $\mathbf{e}_{y,k}^i$ is the measurement error computed with the measurement and the operating point, $\mathbf{x}_{\mathrm{op}} = \{\hat{\mathbf{T}}_{VM,\mathrm{op},k},\ \hat{\boldsymbol{\varpi}}_{\mathrm{op},k},\ \hat{\mathbf{T}}_{GM,\mathrm{op},k}\}$, and $\mathbf{R}^i$ is the associated observation covariance matrix. Depending on the type of measurement, they each take on one of the following forms:

- GPS (4.29):

$$\begin{cases} \mathbf{e}_{y,k}^i = \ln(\mathbf{T}_{VG,k}\hat{\mathbf{T}}_{VG,\mathrm{op},k}^{-1})^{\vee}, \\ \mathbf{R}^i = \mathbf{R}_{VG}, \end{cases} \tag{4.108}$$

  where

$$\hat{\mathbf{T}}_{VG,\mathrm{op},k} = \hat{\mathbf{T}}_{VM,\mathrm{op},k}\hat{\mathbf{T}}_{GM,\mathrm{op},k}^{-1}. \tag{4.109}$$

- Traffic Lights (4.40):

$$\begin{cases} \mathbf{e}_{y,k}^i = \mathbf{p}_{I,k}^j - \hat{\mathbf{p}}_{I,\mathrm{op},k}^j, \\ \mathbf{R}^i = \mathbf{R}_{\mathrm{light}}, \end{cases} \tag{4.110}$$

  where

$$\hat{\mathbf{p}}_{I,\mathrm{op},k}^j = \mathbf{g}(\mathbf{h}(\mathbf{p}_M^j, \hat{\mathbf{T}}_{VM,\mathrm{op},k})). \tag{4.111}$$

- Lane Markings (4.53):

$$\begin{cases} \mathbf{e}_{y,k}^{i} = \mathbf{x}_{k}^{j} - \hat{\mathbf{x}}_{\mathrm{op},k}^{j}, \\ \mathbf{R}^{i} = \mathbf{R}_{\mathrm{lane}}, \end{cases} \tag{4.112}$$

where

$$\hat{\mathbf{x}}_{\mathrm{op},k}^{j} = \mathbf{f}(\boldsymbol{\ell}_{M}^{j}, \hat{\mathbf{T}}_{VM,\mathrm{op},k}, \mathbf{y}^{j}). \tag{4.113}$$

- Wheel Encoders (4.64):

$$\begin{cases} \mathbf{e}_{y,k}^{i} = \boldsymbol{\varpi}_{\mathrm{wheel},k} - \hat{\boldsymbol{\varpi}}_{\mathrm{wheel,op},k}, \\ \mathbf{R}^{i} = \mathbf{R}_{\mathrm{lane}}, \end{cases} \tag{4.114}$$

where

$$\hat{\boldsymbol{\varpi}}_{\mathrm{wheel,op},k} = \mathbf{D}_{\varpi} \hat{\boldsymbol{\varpi}}_{\mathrm{op},k}. \tag{4.115}$$

- Pseudo-Measurement - Vehicle Elevation (4.71):

$$\begin{cases} \mathbf{e}_{y,k}^{i} = z_{\mathrm{pseudo},k} - \hat{z}_{\mathrm{op},k}, \\ \mathbf{R}^{i} = r_{z}, \end{cases} \tag{4.116}$$

where

$$\hat{z}_{\mathrm{op},k} = \mathbf{D}_{z1} \hat{\mathbf{T}}_{VM,\mathrm{op},k} \mathbf{D}_{z2}. \tag{4.117}$$

- Pseudo-Measurement - Vehicle Roll & Pitch (4.78):

$$\begin{cases} \mathbf{e}_{y,k}^{i} = \mathbf{D}_{I} \ln(\mathbf{C}_{\mathrm{pseudo},k} \hat{\mathbf{C}}_{VM,\mathrm{op},k}^{-1})^{\vee}, \\ \mathbf{R}^{i} = \mathbf{R}_{C}', \end{cases} \tag{4.118}$$

where $\hat{\mathbf{C}}_{VM,\mathrm{op},k} \in SO(3)$ is extracted from $\hat{\mathbf{T}}_{VM,\mathrm{op},k}$, and $\mathbf{D}_{I}$ removes the error term that corresponds to the vehicle yaw, as discussed before when formulating the observation model (4.78).

- Pseudo-Measurement - Vehicle Velocity (4.88):

$$\begin{cases} \mathbf{e}^i_{y,k} = v_{\text{pseudo},k} - \hat{v}_{\text{op},k}, \\ \mathbf{R}^i = r_v, \end{cases} \tag{4.119}$$

where

$$\hat{v}_{\text{op},k} = \mathbf{D}_v \hat{\boldsymbol{\varpi}}_{\text{op},k}. \tag{4.120}$$

M-Estimator

In order to minimize the impact of incorrect data association of semantic cues, a Cauchy M-estimator is deployed [16]. Because the inverse covariance matrix, $\mathbf{R}^{i^{-1}}$, acts as a weighting factor for each semantic cue measurement cost term in (4.98), we replace it by

$$\mathbf{Y}^{i^{-1}}_k = (1 + \mathbf{e}^{i^T}_{y,k} \mathbf{R}^{i^{-1}} \mathbf{e}^i_{y,k})^{-1} \mathbf{R}^{i^{-1}}. \tag{4.121}$$

Given a reasonably well initialized vehicle position, this scheme effectively prevents localization failures by scaling down the importance of outliers, which produce large measurement errors, via the associated $\mathbf{Y}^{i^{-1}}_k$.

Gauss-Newton Method

The cost function, $J = J_v + J_y$, is optimized using the standard Gauss-Newton approach. We start by initializing the operating point, $\mathbf{x}_{\text{op}} = \{\hat{\mathbf{T}}_{VM,\text{op},k}, \hat{\boldsymbol{\varpi}}_{\text{op},k}, \hat{\mathbf{T}}_{GM,\text{op},k}\}$, to the predicted means, $\check{\mathbf{T}}_{VM,k}$, $\check{\boldsymbol{\varpi}}_k$, and $\check{\mathbf{T}}_{GM,k}$, from the prediction step.

In each iteration, we first linearize the prior and measurement error terms, $\mathbf{e}_{v,k}$ and $\mathbf{e}^i_{y,k}$, about the operating point, $\mathbf{x}_{\text{op}}$. The resulting combined prior Jacobian matrix, $\mathbf{E}_k$, has been derived in (4.104) and (4.106). Similarly, the combined measurement Jacobian matrix, $\mathbf{G}_k$, is obtained by vertically stacking the Jacobian matrices derived earlier, which correspond to all the measurements at time step k, including $\mathbf{G}_{VG,k}$, $\mathbf{G}^j_{\text{light},k}$, $\mathbf{G}^j_{\text{lane},k}$, $\mathbf{G}_{\varpi}$, $\mathbf{G}_{z,k}$, $\mathbf{G}'_{C,k}$, and $\mathbf{G}_v$.

Next, we substitute the linearized error terms into the cost function and set its

derivative with respect to the perturbation term,

$$\delta\mathbf{x} = \begin{bmatrix} \delta\boldsymbol{\xi}_{VM,k} \\ \delta\boldsymbol{\varpi}_k \\ \delta\boldsymbol{\xi}_{GM,k} \end{bmatrix} \in \mathbb{R}^{18}, \tag{4.122}$$

to zero. This produces the update equation

$$(\mathbf{A}_{\text{pri}} + \mathbf{A}_{\text{meas}})\delta\mathbf{x}^* = \mathbf{b}_{\text{pri}} + \mathbf{b}_{\text{meas}}, \tag{4.123}$$

where

$$\mathbf{A}_{\text{pri}} = \mathbf{E}_k^T \check{\mathbf{P}}_k^{-1} \mathbf{E}_k, \tag{4.124}$$

$$\mathbf{A}_{\text{meas}} = \mathbf{G}_k^T \mathbf{R}_k^{-1} \mathbf{G}_k, \tag{4.125}$$

$$\mathbf{b}_{\text{pri}} = \mathbf{E}_k^T \check{\mathbf{P}}_k^{-1} \mathbf{e}_{v,k}, \tag{4.126}$$

$$\mathbf{b}_{\text{meas}} = \mathbf{G}_k^T \mathbf{R}_k^{-1} \mathbf{e}_{y,k}, \tag{4.127}$$

$\delta\mathbf{x}^*$ is the optimal perturbation, and $\mathbf{R}_k$ is a block diagonal matrix that combines all $\mathbf{R}^i$ and $\mathbf{Y}_k^i$. Solving for $\delta\mathbf{x}^*$, we update the operating point as follows:

$$\hat{\mathbf{T}}_{VM,\text{op},k} \leftarrow \exp((\delta\boldsymbol{\xi}_{VM,k}^*)^\wedge)\hat{\mathbf{T}}_{VM,\text{op},k}, \tag{4.128}$$

$$\hat{\boldsymbol{\varpi}}_{\text{op},k} \leftarrow \hat{\boldsymbol{\varpi}}_{\text{op},k} + \delta\boldsymbol{\varpi}_k^*, \tag{4.129}$$

$$\hat{\mathbf{T}}_{GM,\text{op},k} \leftarrow \exp((\delta\boldsymbol{\xi}_{GM,k}^*)^\wedge)\hat{\mathbf{T}}_{GM,\text{op},k}. \tag{4.130}$$

Finally, the results of the correction step, $\hat{\mathbf{T}}_{VM,k}$, $\hat{\boldsymbol{\varpi}}_k$, and $\hat{\mathbf{T}}_{GM,k}$, are output after convergence or a user-specified number of iterations. If the number of iterations is limited to one, this IEKF formulation simply becomes an EKF. Note that if the cost function, J, increases after an update, a backtracking operation is performed by undoing the update, halving the update step $\delta\mathbf{x}^*$, then reapplying the update. The

corresponding covariance is computed as

$$\hat{\mathbf{P}}_k = (\mathbf{A}_{\text{pri}} + \mathbf{A}_{\text{meas}})^{-1}, \tag{4.131}$$

from the last iteration.

4.5 Toy Example

To validate the effectiveness of our IEKF formulation, simulations were conducted with a simplified setup. First, a real-world vehicle trajectory of around 1 km was collected and used as the ground truth vehicle localization, $\mathbf{T}_{VM,k}$, in the simulation. Then, to simulate the semantic cues, we placed some traffic light point landmarks and lane boundaries along the vehicle trajectory as shown in the top left plot of Figure 4.2.

Based on this simple environment, the remaining information required for the simulation was generated: the ground truth motion trajectory of the GPS frame, $\mathbf{T}_{GM,k}$, was initialized at a pose not too far off from the semantic map frame, $\underrightarrow{\mathcal{F}}_M$, and then simulated by performing a random walk; the GPS measurements, $\mathbf{T}_{VG,k}$, were generated by compounding the simulated $\mathbf{T}_{GM,k}$ and $\mathbf{T}_{VM,k}$; the front-facing camera images were produced by projecting the traffic lights and lane boundaries into the image space at every time step k along the vehicle trajectory using the calibration information of a real camera. All the measurements are then injected with synthetic Gaussian noises. In this toy example, wheel encoders were not included because they are only necessary when GPS dropouts occur, and we also assume perfect data association of the semantic cue measurements by recording the correspondences between the semantic cue positions in the map and their image projections.

A snapshot of the simulation can be seen in Figure 4.2, where the overview of the vehicle trajectory, the simulated camera view, the current vehicle pose estimation, and the estimation of GPS-to-map offset are shown. The semantic cues are also visible as circles (traffic lights) and lines (lane boundaries). The resulting vehicle localization errors shown in Figure 4.3 indicates that the developed IEKF localization can achieve

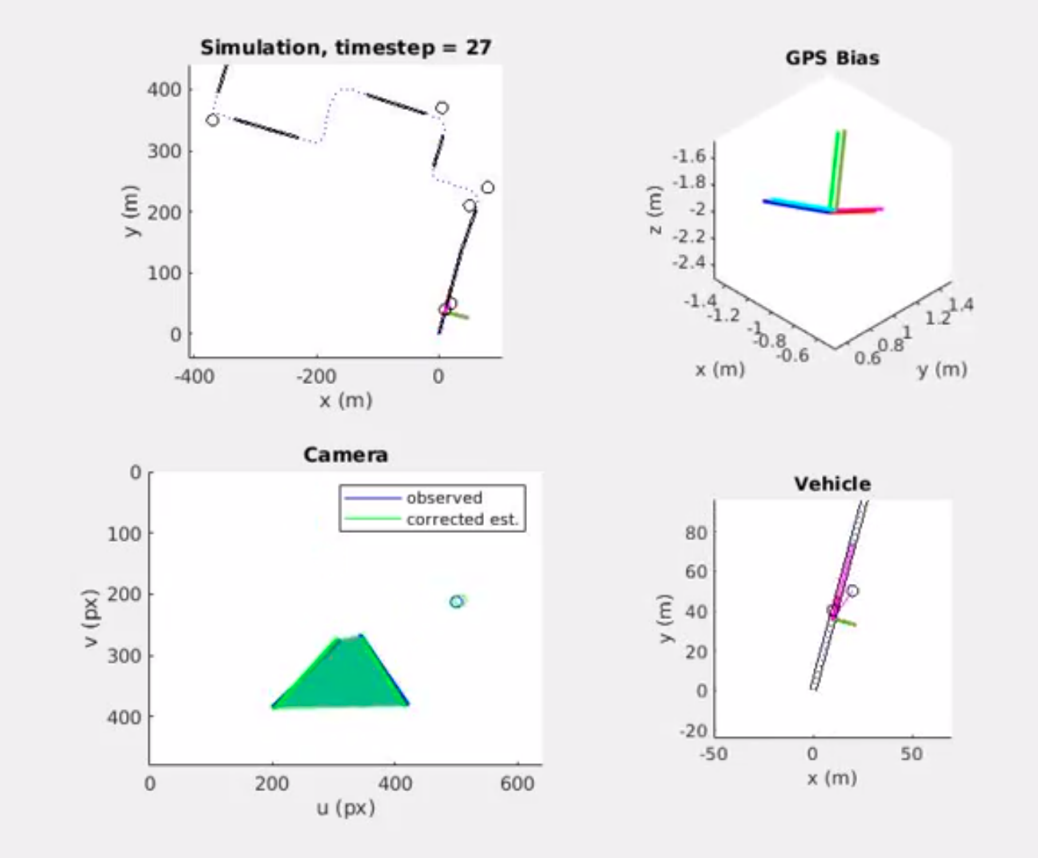


Figure 4.2: A view of the toy example simulation. Top left: the overview of the simulation, including the entire vehicle trajectory (dashed black line) and semantic cues represented as circles for traffic lights and solid black lines for lane boundaries. Note the stretch of trajectory where no lane boundary exists, which impacts the localization accuracy. Bottom left: the simulated camera view. The blue circle and quadrilateral are the detected traffic light and lane boundaries; the green ones are their predicted positions based on the estimated vehicle pose. Top right: the ground truth GPS frame is in red, blue, and green; the estimated GPS frame is in magenta, dark green, and cyan. Notice that they are very close, which indicates that the IEKF formulation is able to correct for GPS-to-map offset as designed. Bottom right: a zoomed-in view around the vehicle with the semantic cues. Magenta lines and shadings indicate detections of traffic lights and lane boundaries, respectively, by the vehicle.

decent performance with lateral errors mostly within 10cm, which is acceptable for autonomous vehicle navigation.

It is worth mentioning that this toy example was also used to evaluate and decide between EKF- and UKF-based algorithm during early development of the semantic localizer, when the correction step was yet to be made iterative. The two versions

	Scenario 1	Scenario 2	Scenario 3
EKF	3.3369 s	2.4323 s	2.2842 s
UKF	132.6152 s	88.7561 s	93.5365 s

Table 4.1: The runtime of vehicle localization algorithms in three different scenarios of the toy example. EKF-based algorithm is faster than UKF-based algorithm by orders of magnitude.

of localizer were applied to the toy example, along with two other scenarios consisting of different vehicle trajectories and semantic cue placements. However, as Table 4.1 shows, because of the sigma point generation process utilized by UKF, the computational time of the UKF-based localizer was orders of magnitude slower than EKF-based localizer, making it infeasible for real-time self-driving applications. Therefore, UKF was eliminated from our consideration, and EKF-based approach became the sole focus of all subsequent development of the localization algorithm from that point on.

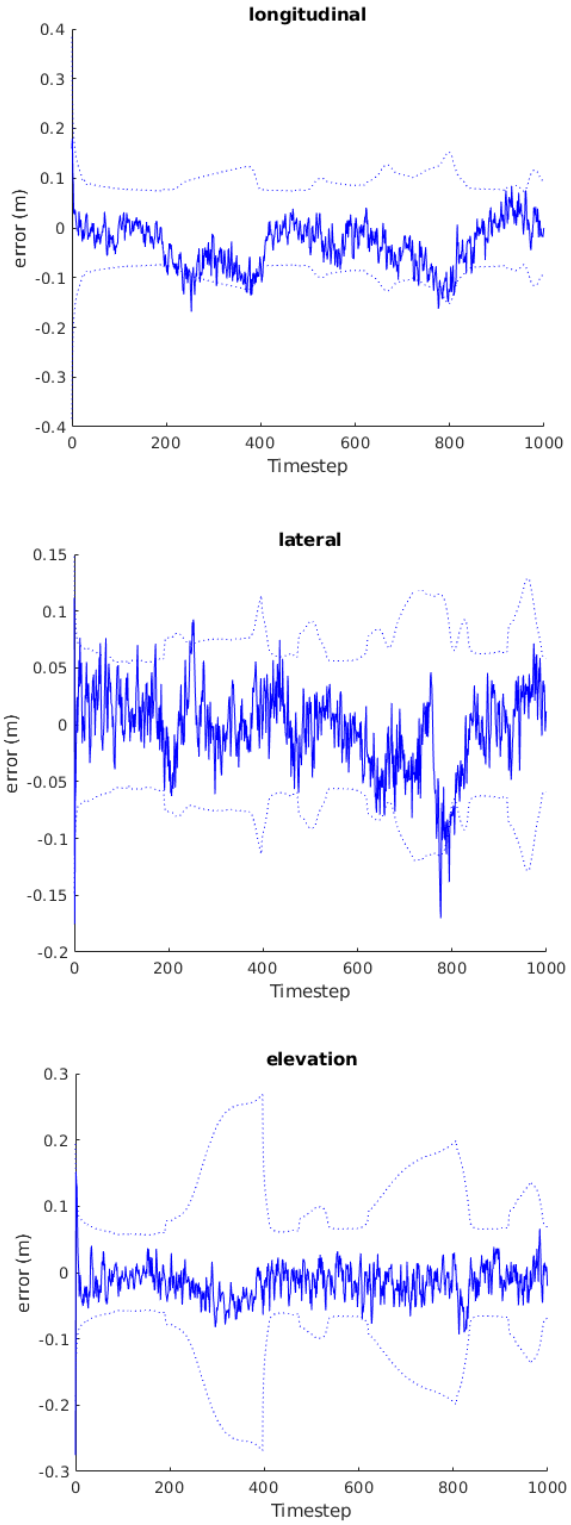


Figure 4.3: Vehicle localization errors of EKF-based algorithm in the toy example. The dotted lines form the uncertainty envelope. Note that the lateral error exceeds 10 cm at around time step 800, which corresponds to the stretch of the vehicle trajectory without any lane boundary.

Chapter 5

Experiments

The proposed semantic localization algorithm was tested in simulations followed by experiments using real-world datasets. However, significant difficulties were encounter when selecting datasets that could be used to reliably evaluate our localization method. As a compromise, quantitative results are presented based on the simulation, and anecdotal results gathered from two real-world datasets are shown to validate the feasibility of our approach in reality.

5.1 Carla Simulation

5.1.1 Simulation Setup and Process

Our localization algorithm was first tested using Carla, an autonomous driving simulator [11]. In particular, the simulations were conducted using the map "Town10HD", which offers a photorealistic urban driving environment and a perfect semantic map. The benefit of using a simulator is the availability of perfect semantic maps and ground truth, which simplifies the analysis of localization results. Due to Carla not supporting an offset in GPS measurements, we instead manually injected one with 2 m in both longitude and latitude. The time-dependency of the offset is ignored since it is negligible in the duration of a simulation run. With this setup, the simulation data was obtained from roughly 1 km of driving. The vehicle path comparing ground

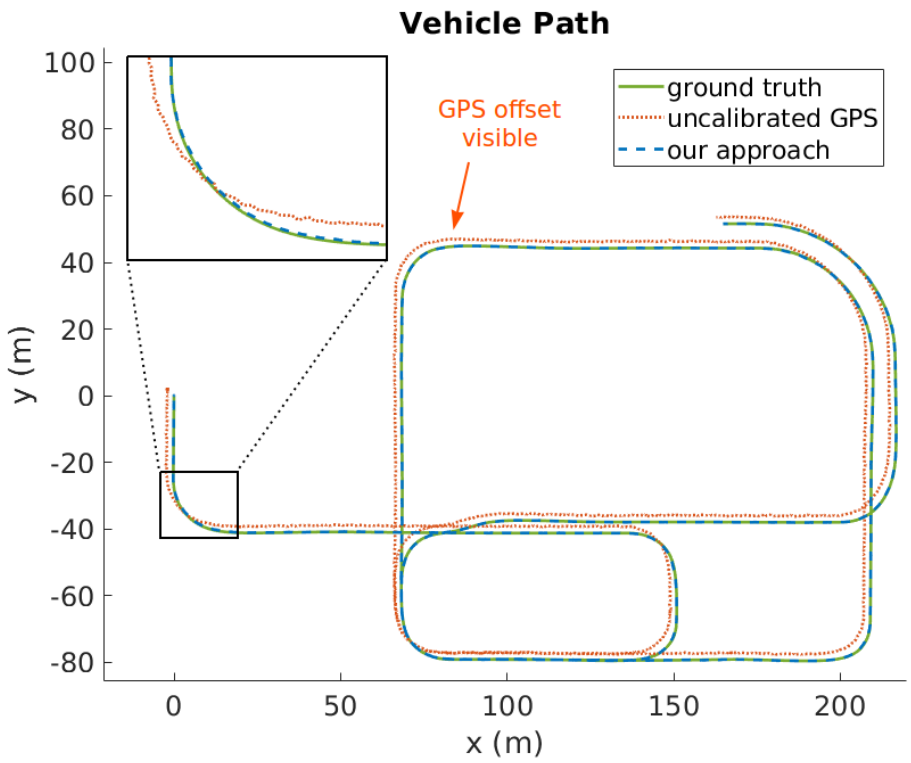


Figure 5.1: Vehicle path of Carla simulation with total length of roughly 1 km. The ground truth path is compared with results from uncalibrated GPS and our proposed approach. One of the turns is zoomed in to show that the estimated path using our approach very closely overlap with the ground truth path, while the uncalibrated GPS path significantly diverges from it.

truth with our method and uncalibrated GPS is shown in Figure 5.1.

5.1.2 Parameter Tuning

Our localization pipeline involves numerous parameters, including the outlier distance threshold in data association, and matrices related to the state estimator. In the process models, $\mathbf{Q}_C$ is associated with the state covariance of vehicle pose and velocity, and $\mathbf{Q}_{GM}$ affects the magnitude of the random walk of GPS frame. The observation noise parameters consist of $\mathbf{R}_{VG}$, $\mathbf{R}_{\text{light}}$, $\mathbf{R}_{\text{lane}}$, $\mathbf{R}_{\varpi}$, and $\mathbf{R}_{\text{pseudo}} = \{r_z, \mathbf{R}'_C, r_v\}$ that are associated with GPS, traffic light, lane markings, wheel encoders, and pseudo-measurements, respectively. These parameters are manually tuned by initializing them with reasonable values, followed by adjustments to achieve optimal performance evaluated on a validation dataset generated from Carla.

Table 5.1: Carla localization accuracy of the proposed IEKF localizer with & without GPS dropouts. Due to abundance of lane markings, little degradation of lateral and heading accuracy is observed when GPS dropouts occur.

	Experimental Scenarios					
	No Dropouts			GPS Dropouts		
Errors	Median	95%	99%	Median	95%	99%
Longitudinal (m)	0.053	0.145	0.185	0.069	0.370	0.504
Lateral (m)	0.031	0.104	0.172	0.032	0.158	0.270
Heading (rad)	0.004	0.014	0.025	0.004	0.015	0.028

5.1.3 Localization Results

The longitudinal, lateral, and heading localization errors computed using ground truth are summarized in Table 5.1. Our proposed method achieves highly accurate results with a median longitudinal error of 0.053 m, a median lateral error of 0.031 m, and a median heading error of 0.004 radians. When shown as a histogram in Figure 5.2, we observe that the longitudinal errors have a larger spread than lateral errors, and the vehicle heading always remains very accurate. This is in line with our expectations since lane markings, the most abundant type of semantic cues, only provide lateral and heading corrections. In contrast, longitudinal corrections offered by traffic lights are only available around road intersections.

GPS Offset Estimation

Being the key motivation for developing the proposed localization algorithm, achieving accurate estimation of the GPS-to-map offset is crucial. The blue line in Figure 5.3 shows the GPS offset estimation error. Starting from a poor initial guess, our localization algorithm successfully refines the estimates and drops the error down to just a few centimetres, with no manual calibration required.

GPS Dropouts

To evaluate the robustness of our localization algorithm, we introduced periodic GPS dropouts lasting for 30 seconds in every 60-second interval, i.e., half of the GPS

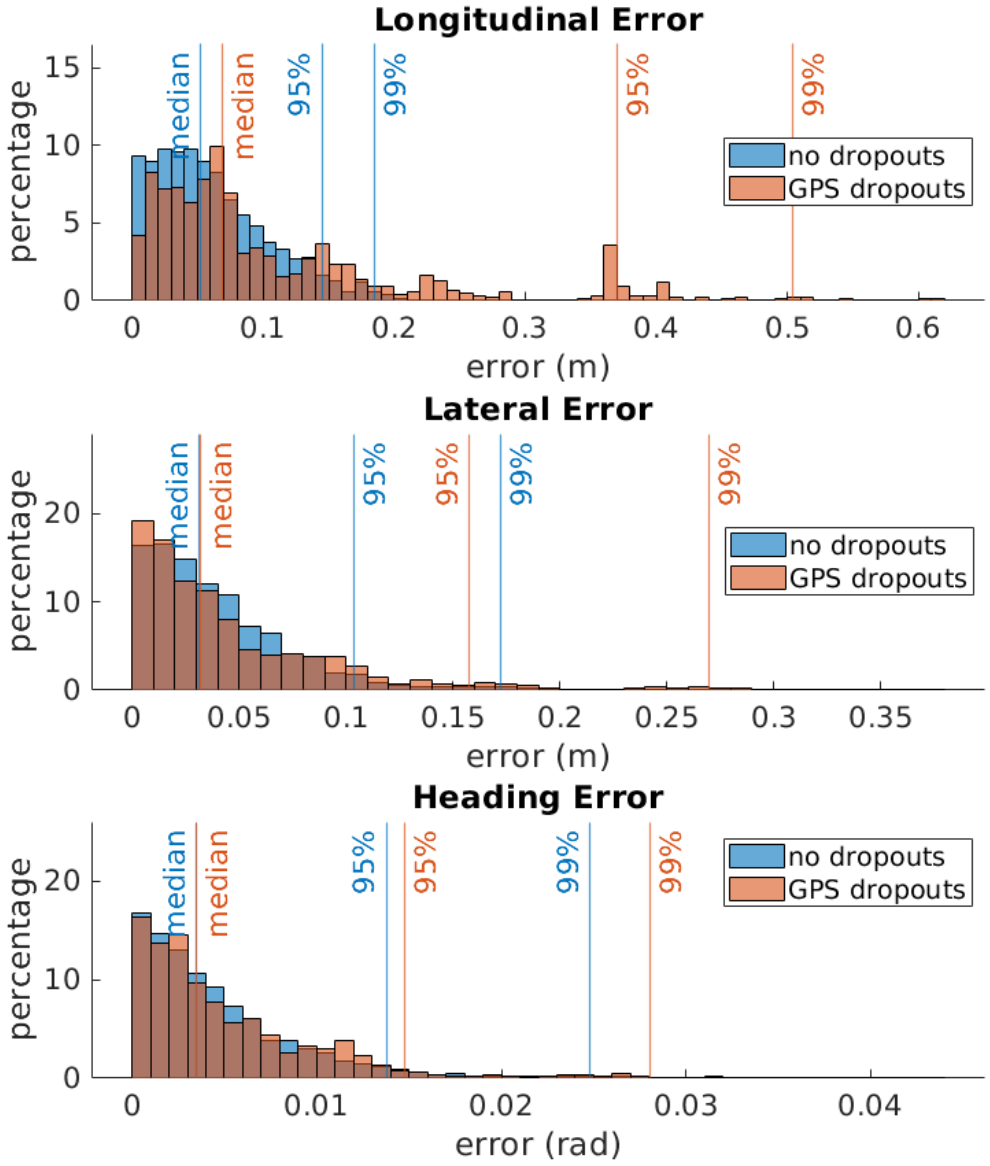


Figure 5.2: Histograms of longitudinal (top), lateral (middle), and heading (bottom) localization errors of our estimator on Carla simulation comparing scenarios with and without GPS dropouts.

measurements are lost. Under such conditions, the proposed approach is still able to achieve accurate estimation of the GPS offset as shown by the orange line in Figure 5.3, albeit at a slower pace. Furthermore, the localization results are shown in Figure 5.2 and summarized in Table 5.1. Compared to the scenario without any

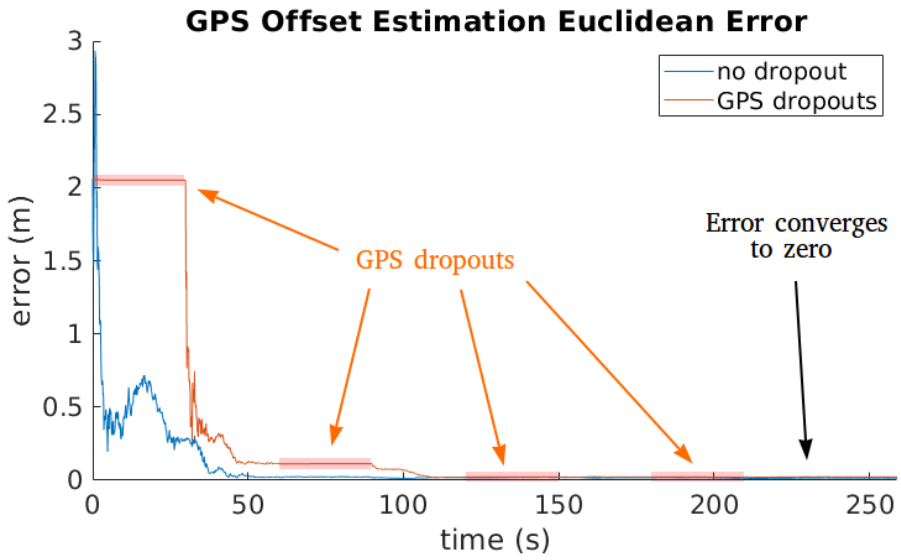


Figure 5.3: Euclidean error of GPS-to-map offset estimation of Carla simulation. By taking advantage of semantic cues, our localization algorithm is able to estimate the GPS measurement offset with decimetre-level accuracy even with the presence of periodic GPS dropouts.

GPS dropout, we observe virtually no increase in the median lateral and heading errors largely due to frequent occurrences of lane markings, which keep the vehicle in the correct lane. This highlights the importance and effectiveness of lane markings as a crucial type of semantic cues in semantic localization. On the other hand, there is a significant decline in performance over the worst case scenario in terms of longitudinal error, where it increases from 0.185 m to 0.504 m. This can be attributed to the infrequent appearance of traffic lights, which help with longitudinal localization, when road intersections are not nearby. In this case, the vehicle can only rely on wheel odometry for relative localization during GPS dropouts, which accumulates longitudinal errors. Nevertheless, the localization accuracy is still acceptable for autonomous driving. This demonstrates the robustness of our proposed approach against frequent GPS dropouts by leveraging semantic cues.

Table 5.2: Issues of the public datasets with semantic maps for our experiments.

Dataset	Issue
nuScenes [5]	Missing GPS data; vehicle trajectories are fragmented
Lyft Level 5 [21]	Missing GPS and wheel encoder data
Argoverse [6]	Missing traffic light locations; only provides lane centerline

5.2 Mcity Experiment

Unfortunately, the vast majority of the publicly available self-driving datasets do not provide semantic maps. Those that do all lack other components necessary for our experiments. For instance, the nuScenes dataset does not provide raw GPS data [5]. Table 5.2 summarizes the datasets with semantic maps and the components they lack for our experiments. Therefore, to demonstrate real-world feasibility, experiments were conducted using an internal dataset provided by aUToronto that was collected during the SAE AutoDrive competition at Mcity, where the incident of uncalibrated GPS occurred [4]. However, due to the lack of localization ground truth in the dataset for comparison, this will only serve as anecdotal results to verify the effectiveness of our approach. No further parameter tuning was made between the Carla simulation and the real-world experiments. Figure 1.1 provides a side-by-side comparison of a snapshot of the camera image overlaid with projection of the lane boundaries from the semantic map, which indicates the estimated location of the vehicle with respect to the map. Visually, we see that the GPS data is unusable on its own while our approach is able to self-calibrate for the GPS offset and has the projected lane boundaries well aligned with the lane markings to achieve accurate localization.

5.3 Boreas Experiment

After the Mcity experiment, a second attempt in real-world testing was made using another internal dataset that was collected by our laboratory's vehicle named "Boreas". The vehicle is equipped with all the sensors required for our experiment, and there is a postprocessed vehicle localization solution that we planned to use as

Figure 5.4: The lane lines (red) from the manually created semantic map for the Boreas dataset projected to the camera image using the postprocessed localization solution. The alignment between the projected lane lines and the lane markings is poor.

the ground truth for our localization results to compare against. However, the dataset lacked a semantic map. Therefore, we proceeded to make one using satellite images along the vehicle trajectory. The lane graph as well as the traffic light positions were all manually created based on the satellite images, and the elevation of traffic lights were extracted from LIDAR data of Boreas.

5.3.1 Semantic Map Refinement

The quality of the semantic map was evaluated by projecting it to the camera image space using the postprocessed localization solution and comparing its alignment with the semantic cues, mainly lane markings. As shown in Figure 5.4, the resulting

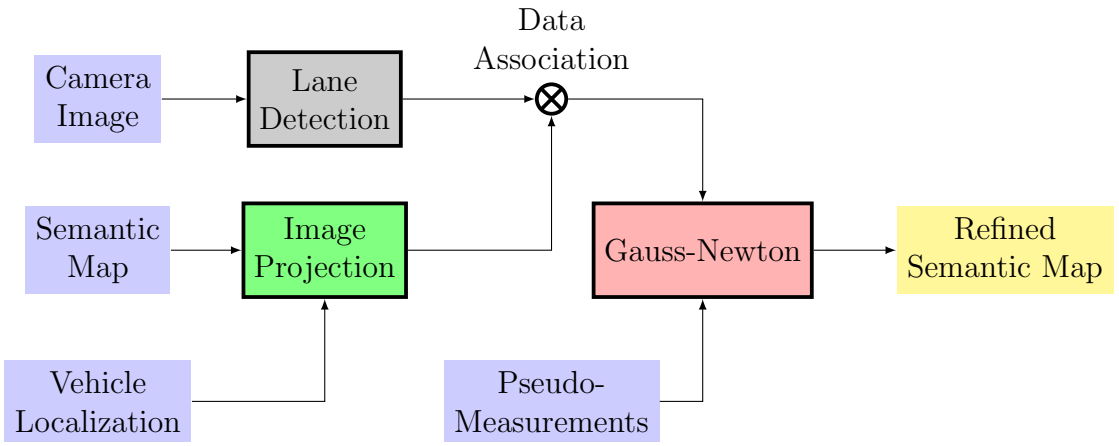


Figure 5.5: The proposed semantic map refinement pipeline. For each camera frame, the image is passed through the lane detector, and the semantic map is projected to the image space using the corresponding vehicle location. The data association step finds the correspondences between detection results and the semantic map projections. The results of the data association from all the camera frames are then fused with pseudo-measurements in a Gauss-Newton algorithm to produce the refined semantic map output.

alignment was poor, so no proper evaluation of the semantic localization method could be made using the dataset as it was.

In an attempt to improve the semantic map such that the dataset could become useful for our experiment, we proposed a semantic map refinement scheme. The goal is to refine a satellite-based semantic map — mainly the lane graph — so that it locally aligns with the postprocessed vehicle localization solution in the on-board camera image space. Keep in mind that not all semantic cues were observed by the camera in the vehicle path, but we still want their locations in the map to be updated along with the observed ones in order to preserve their relative positions. This map refinement scheme could potentially be a cheap method to produce semantic maps from satellite images.

Map Refinement Process

Given the original semantic map, the postprocessed vehicle localization solution, and the camera images captured along the vehicle trajectory, we formulate the map refinement process as a batch optimization problem solved through Gauss-Newton method.

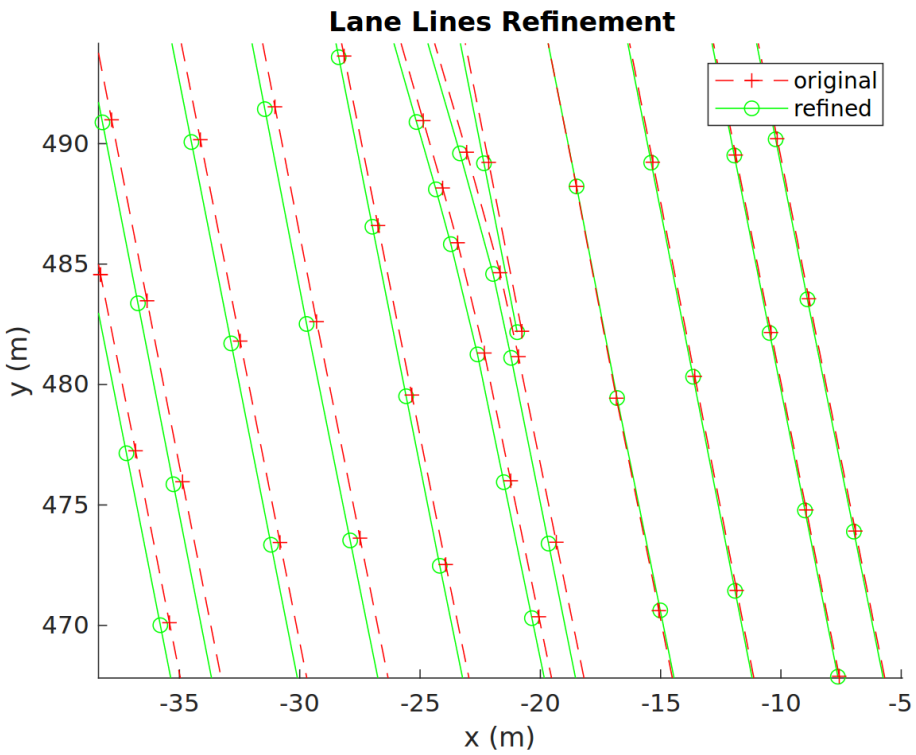
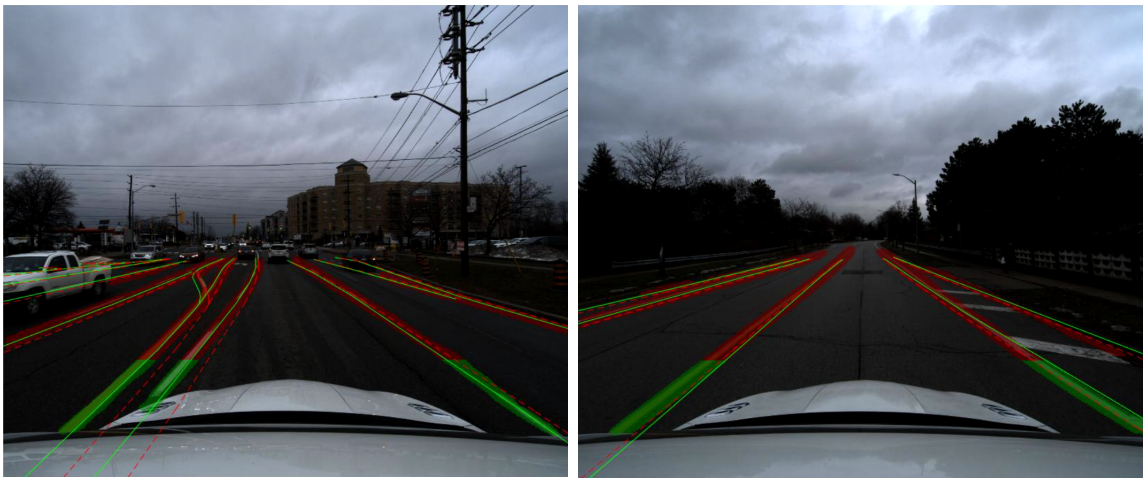


Figure 5.6: Original vs. refined lane graph of a multi-lane road. All of the lane lines, including those that are not observed by the vehicle, have been updated. Some lane lines are shifted more than the others depending on the camera observations.

The states to be updated are the positions of the nodes forming the polylines of the lane graph. They are initialized with the original semantic map input.

For every camera frame captured, the image goes through the same lane detection and data association process as in Section 3.3, except that the image projection involves a semantic map that is being updated rather than the vehicle location, which is a constant input in this algorithm.

To ensure that all the nearby lane lines are updated along with the observed ones, we introduce two types of pseudo-measurements. The first type applies to every pair of consecutive nodes belonging to the same lane line. For each pair, the pseudo-measurement is their difference in coordinates such that the each lane line will largely preserve its shape after the map refinement. The second type of pseudo-measurement involves nodes belong to different lane lines. First, the midpoint of every pair of consecutive nodes is computed. Next, for each pair of nearby midpoint belonging to two



(a) (b)

Figure 5.7: Original (red) vs. refined (green) lane lines projected to the camera image using the postprocessed localization solution. The lane markings identified by the lane detector are highlighted in green or red depending on the distance to the vehicle. Compared to the original lane lines, the refined lane lines overlap with the lane markings more in (a), but less so in (b).

different lane lines, their difference in coordinates becomes the pseudo-measurement. This helps maintain the relative position between lane lines such that the lane width can be preserved.

Lastly, all the data association results from the camera frames as well as the pseudo-measurements are passed into the Gauss-Newton method for iterative updates until convergence. Figure 5.5 summarizes the described map refinement pipeline.

Map Refinement Results

A sample of the refined semantic map compared to the original can be seen in Figure 5.6. Unfortunately, after the map refinement, although improvements were made in terms of map alignment in the camera images, there are still some places where the projected lane lines do not align with the lane markings well (see Figure 5.7), making the postprocessed localization solution unreliable as the ground truth. Therefore, we determined that the Boreas dataset is incapable of quantifying the localization accuracy of semantic localization method in a meaningful manner. Nevertheless, we still ran some experiments using the Boreas dataset to gain qualitative insights into

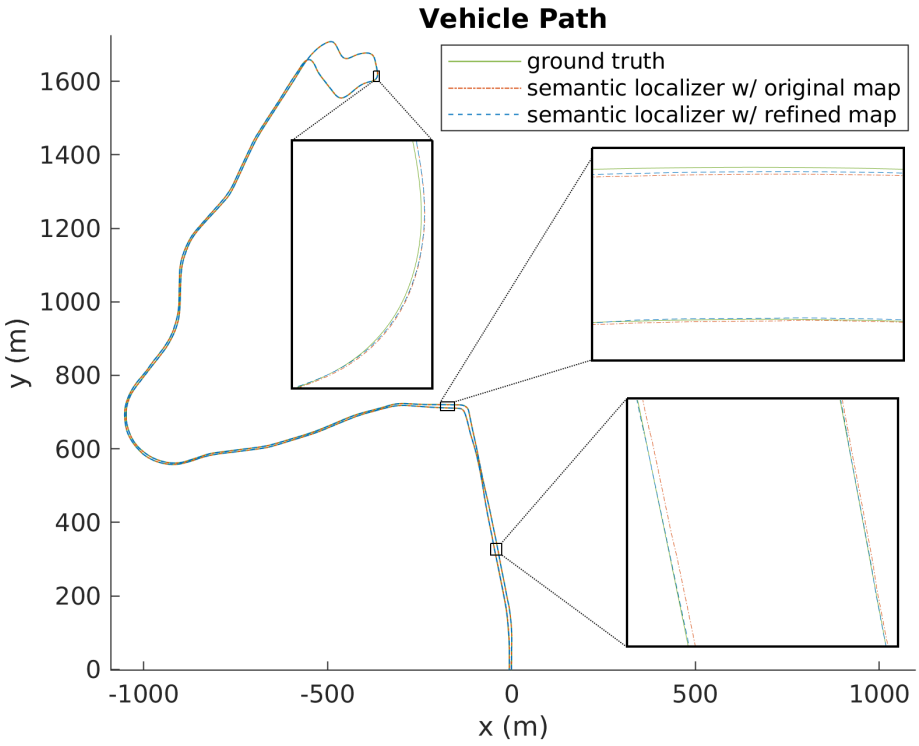


Figure 5.8: Vehicle path of Boreas experiment with total length of roughly 5 km. The ground truth path is compared with results from the semantic localizer using either the original semantic map or the refined one. Various locations along the vehicle path are zoomed in to show that the performance using the refined map is generally better than the original map, but there are places where the improvement is minimal.

the semantic localization algorithm. Figure 5.8 illustrates the vehicle path comparing the postprocessed localization solution, i.e., "ground truth", with the semantic localizer using either the original semantic map or the refined one. A histogram of the localization errors is also shown in Figure 5.9. From these figures, we can observe that generally speaking, the refined semantic map does produce better semantic localization results, especially in the lateral localization, where lane markings are mostly responsible for.

5.3.2 Qualitative Localization Results

Despite the challenges we encountered in utilizing the Boreas dataset, some qualitative analysis can still be conducted by treating the postprocessed localization solution as

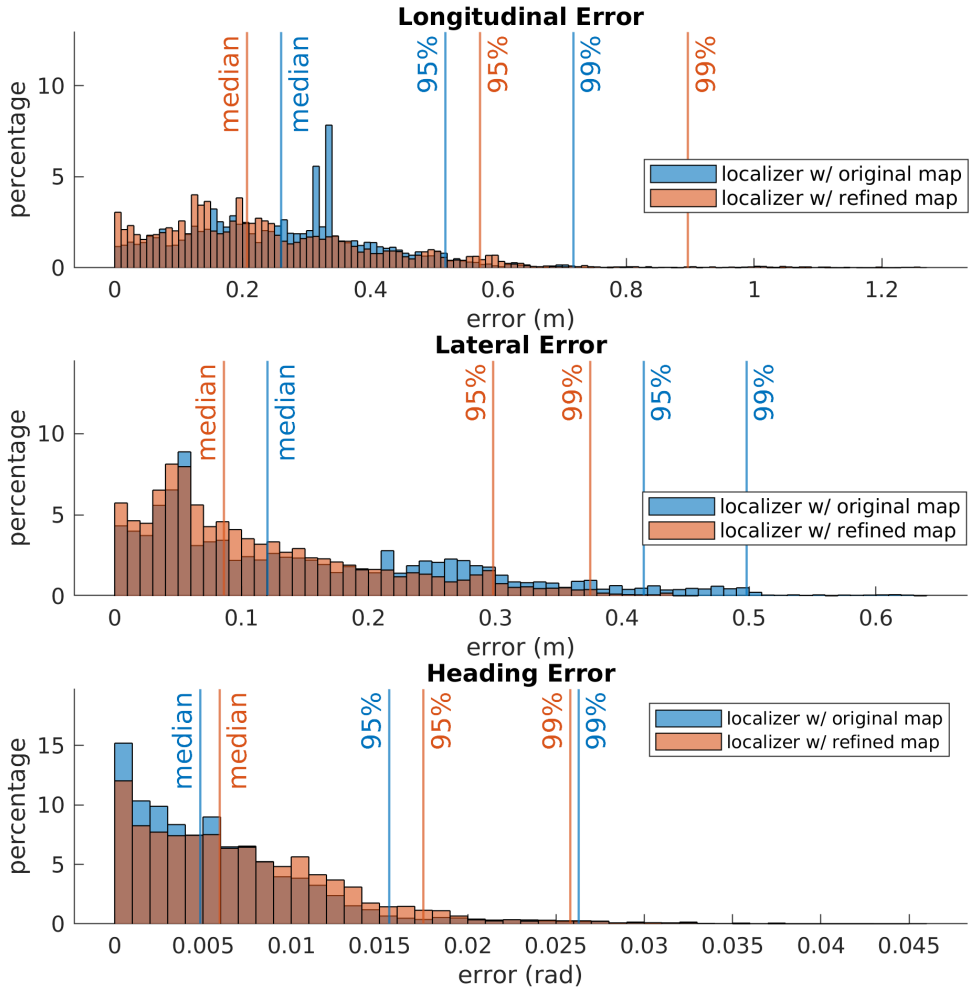


Figure 5.9: Histograms of longitudinal (top), lateral (middle), and heading (bottom) localization errors of Boreas experiment comparing localization results with the original map and the refined map.

the ground truth and comparing the semantic localizer's relative performance under various experimental scenarios. The following localization results are all generated using the refined semantic map.

Ablation Study

To observe the localization contribution made by each type of semantic cues, we repeat the same semantic localization, but with either the lane markings or traffic

Table 5.3: Boreas localization errors comparing scenarios with both types of semantic cues present vs. only one type of semantic cues.

	Experimental Scenarios								
	All Semantic Cues			Only Lane Markings			Only Traffic Lights		
Errors	Median	95%	99%	Median	95%	99%	Median	95%	99%
Longitudinal (m)	0.213	0.591	0.911	0.219	0.599	0.905	0.393	0.760	0.962
Lateral (m)	0.093	0.355	0.455	0.094	0.338	0.463	0.191	0.574	1.061
Heading (rad)	0.006	0.017	0.026	0.006	0.018	0.025	0.005	0.014	0.022

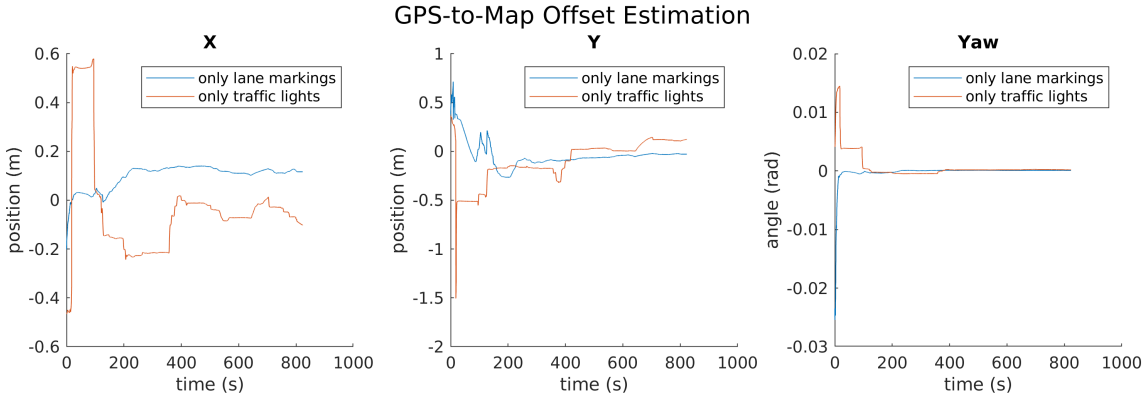


Figure 5.10: The GPS-to-map offset estimation results when only lane markings (blue) or traffic lights (orange) are used in the semantic localization. Notice that the blue lines converge to some relatively constant values faster than orange lines.

lights disabled in the semantic localizer. The localization results are presented in Table 5.3. As expected, the overall localization accuracy with only lane markings is better than with only traffic lights, which can be explained by their drastic difference in the frequency of occurrence. More interestingly, when comparing against the full semantic localization, there is very little degradation in accuracy for the scenario with only lane markings, including the longitudinal direction. This can be attributed to the abundance of lane markings, which allows the estimation of GPS-to-map offset to quickly converge as shown in Figure 5.10, thus achieving GPS calibration faster and producing better GPS measurements for more reliable lateral and longitudinal localization. This also suggests that the lane markings have a disproportionately big contribution to the semantic localization algorithm compared to the traffic lights.

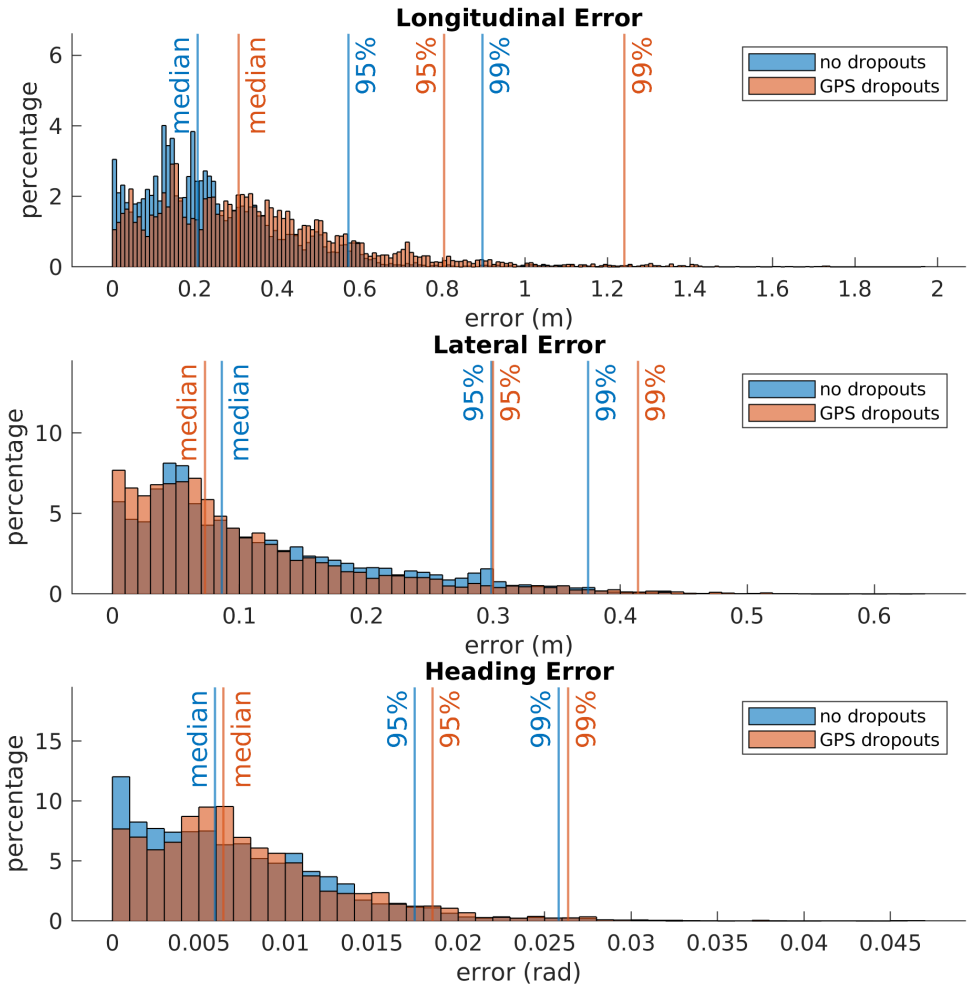


Figure 5.11: Histograms of longitudinal (top), lateral (middle), and heading (bottom) localization errors of Boreas experiment comparing scenarios with and without GPS dropouts.

GPS Dropouts

The histogram in Figure 5.11 compares the semantic localization performance with and without GPS dropouts. Similar to the Carla simulation, even with the loss of half of the GPS measurements, we observe that lateral localization accuracy barely degrades, while the longitudinal accuracy suffers more due to the relative sparsity of traffic lights compared to lane markings. Again, this highlights the effectiveness of utilizing lane markings to keep the vehicle in the correct lane.

Chapter 6

Conclusion and Future Work

6.1 Summary of Contributions

In this thesis, we proposed a method capable of localizing an autonomous vehicle while self-calibrating for an offset between live GPS and semantic map frames. This is achieved by using a lightweight semantic map containing locations of lane boundaries and traffic lights, which are complementary in correcting for lateral and longitudinal position of the vehicle. These semantic cues are detected via a monocular camera and integrated with GPS and wheel encoders.

In Chapter 3, we introduced the semantic map as well as the preprocessing steps of the semantic cues. The semantic map is shown to simply be made up of a lane graph and positions of sparsely distributed traffic lights, making it very lightweight compared to a typical LIDAR map. We then discussed the two corresponding classes of semantic cue: lane markings and traffic lights, which we assumed to be detected by an on-board camera that is commonplace on self-driving vehicles. The computational cost-effectiveness of our proposed algorithm was highlighted with the fact that these common yet crucial semantic cues are most definitely tracked by the self-driving system already, thus no additional implementation of CNN detectors is necessary. Lastly, for the proposed localization algorithm to utilize the semantic cues, we outlined the data association process that corresponds the semantic cue detections with their known positions in the semantic map.

In Chapter 4, we laid out the detailed mathematical formulation of our proposed semantic localization algorithm. We first properly defined the localization problem for which to solve, which includes the estimation of three quantities: vehicle pose, velocity, and the GPS-to-map offset. Next, the expression and linearization of the process models as well as the measurement models were shown. In addition to GPS and semantic cues, the wheel encoders and some pseudo-measurements pertaining to vehicle's physical constraints are also included in the measurement models. Finally, we deployed a modified IEKF to solve the localization problem. The IEKF is modified by replacing the iterative correction step with Gauss-Newton method. A toy example was implemented to validate the developed semantic localizer.

Chapter 5 documents the results of simulations and experiments conducted, part of which has previously been published in "Self-Calibration of the Offset Between GPS and Semantic Map Frames for Robust Localization" by Tseng and Barfoot [39]. Our approach was first evaluated using Carla simulator, which demonstrated robustness against GPS dropouts in addition to achieving decimetre-level accuracy. We encountered great difficulties when attempting to conduct real-world experiments. Due to the nature of our algorithm, semantic map is an integral part of the experiments. However, few public datasets have semantic maps. We subsequently turned to two different internal datasets. The first dataset was collected by a vehicle with its GPS not calibrated to the semantic map, but it is missing a ground truth for performance analysis. The semantic map of the second dataset lacks the accuracy required for proper semantic localization, which led to the devising of a map refinement scheme. But the quality of the refined semantic map was determined to be insufficient still. Therefore, these experiments only serve to show some qualitative analysis, including the real-world feasibility of our approach, as well as the robustness against GPS dropouts.

6.2 Future Work

Our proposed semantic localization algorithm currently only utilizes two types of semantic cues: lane markings and traffic lights. To decrease the gap between semantic cue observations due to sparsity, other types of semantic cues such as traffic signs and stop lines can be included as well. In particular, the vehicle's longitudinal accuracy is always worse than the lateral accuracy due to the relative sparsity of traffic lights compared to lane markings. Diversifying the types of semantic cues can also make the algorithm more robust under various driving environments where particular types of semantic cues might not exist.

The real-world experiments carried out in Chapter 5 are somewhat unsatisfactory due to the lack of a comprehensive dataset for a quantitative analysis of the localization performance. The Boreas experiments especially highlight the importance of an accurate semantic map and its impact on the quality of the semantic localization. Even though the proposed map refinement pipeline failed to produce a map that would enable quantitative analysis of the localization algorithm, it does show some promise in improving the quality of the semantic maps. Therefore, with further development and fine-tuning, the map refinement pipeline can potentially achieve its goal and make the Boreas dataset better suited for our experiments. We can alternatively look for a better dataset to conduct the experiments with. To fully demonstrate the usefulness of our approach, the dataset needs to contain raw uncalibrated GPS data, front-facing camera images, wheel encoders, and a high quality semantic map that has lane graph, traffic lights, and preferably other types of semantic cues. Such a semantic map can likely be obtained through commercial mapping services upon request. Additionally, the dataset must have a localization ground truth such that the projected semantic map closely aligns with all the semantic cues in the camera images. A possible option is to generate it via a LIDAR-based localization method, which is capable of highly accurate localization.

If satisfactory experimental results can be obtained through a better dataset, the next step would be the open-loop and even the closed-loop implementation of the

semantic localization system on an autonomous vehicle. Another possible direction of research would be the development of detection and mitigation strategies when the semantic map disagrees with the observations made by the onboard sensors in situations such as an outdated map.

Bibliography

[1] Naoki Akai et al. "Autonomous driving based on accurate localization using multilayer LiDAR and dead reckoning". In: *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. 2017, pp. 1–6.

[2] S. Anderson and T. D. Barfoot. "Full STEAM ahead: Exactly sparse Gaussian process regression for batch continuous-time trajectory estimation on SE(3)". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 157–164.

[3] Timothy D. Barfoot. *State Estimation for Robotics*. 1st. New York, NY, USA: Cambridge University Press, 2017. DOI: 10.1017/9781316671528.

[4] Keenan Burnett et al. "Zeus: A system description of the two-time winner of the collegiate SAE autodrive competition". In: *Journal of Field Robotics* 38.1 (2021), pp. 139–166.

[5] Holger Caesar et al. "nuScenes: A multimodal dataset for autonomous driving". In: *arXiv preprint arXiv:1903.11027* (2019).

[6] Ming-Fang Chang et al. "Argoverse: 3D Tracking and Forecasting with Rich Maps". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[7] F. Chausse, J. Laneurit, and R. Chapuis. "Vehicle localization on a digital map using particles filtering". In: *IEEE Proceedings. Intelligent Vehicles Symposium*. 2005, pp. 243–248.

[8] Kyoungtaek Choi, Jae Kyu Suhr, and Ho Gi Jung. "FAST pre-filtering-based real time road sign detection for low-cost vehicle localization". In: *Sensors* 18.10 (2018), p. 3590.

[9] M. J. Choi et al. "Low-Cost Precise Vehicle Localization Using Lane Endpoints and Road Signs for Highway Situations". In: *IEEE Access* 7 (2019), pp. 149846–149856. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2947287.

[10] Z. J. Chong et al. "Synthetic 2D LIDAR for precise vehicle localization in 3D urban environment". In: *IEEE International Conference on Robotics and Automation*. 2013, pp. 1554–1559.

[11] Alexey Dosovitskiy et al. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[12] Philipp Egger et al. "Posemap: Lifelong, multi-environment 3D lidar localization". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 3430–3437.

[13] D. Fontanelli, L. Ricciato, and S. Soatto. "A Fast RANSAC-Based Registration Algorithm for Accurate Localization in Unknown Environments using LIDAR Measurements". In: *IEEE International Conference on Automation Science and Engineering*. 2007, pp. 597–602.

[14] D. Gruyer, R. Belaroussi, and M. Revilloud. "Map-aided localization with lateral perception". In: *IEEE Intelligent Vehicles Symposium Proceedings*. 2014, pp. 674–680.

[15] Dominique Gruyer, Rachid Belaroussi, and Marc Revilloud. "Accurate lateral positioning from map data and road marking detection". In: *Expert Systems with Applications* 43 (2016), pp. 1–8.

[16] Frank R Hampel et al. *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 1986.

[17] Alberto Y Hata and Denis F Wolf. "Feature detection for vehicle localization in urban environments using a multilayer LIDAR". In: *IEEE Transactions on Intelligent Transportation Systems* 17.2 (2015), pp. 420–429.

[18] Jun-Hyuck Im, Sung-Hyuck Im, and Gyu-In Jee. "Vertical corner feature based precise vehicle localization using 3D LIDAR in urban area". In: *Sensors* 16.8 (2016), p. 1268.

[19] K. Jo, K. Chu, and M. Sunwoo. "GPS-bias correction for precise localization of autonomous vehicles". In: *IEEE Intelligent Vehicles Symposium (IV)*. June 2013, pp. 636–641. DOI: 10.1109/IVS.2013.6629538.

[20] K. Jo et al. "Precise Localization of an Autonomous Car Based on Probabilistic Noise Models of Road Surface Marker Features Using Multiple Cameras". In: *IEEE Transactions on Intelligent Transportation Systems* 16.6 (2015), pp. 3377–3392.

[21] R. Kesten et al. *Level 5 Perception Dataset 2020*. https://level-5.global/level5/data/. 2019.

[22] Jean Laneurit, Roland Chapuis, and Frédéric Chausse. "Accurate vehicle positioning onto a numerical map". In: *International Journal of Control, Automation and Systems* 3 (2005), pp. 15–31.

[23] Cedric Le Gentil, Teresa Vidal-Calleja, and Shoudong Huang. "IN2LAAMA: Inertial Lidar Localization Autocalibration and Mapping". In: *IEEE Transactions on Robotics* (2020).

[24] Byung-Hyun Lee et al. "GPS/DR Error Estimation for Autonomous Vehicle Localization". In: *Sensors* 15 (Aug. 2015), p. 20779. DOI: 10.3390/s150820779.

[25] H. Li, F. Nashashibi, and G. Toulminet. "Localization for intelligent vehicle by fusing mono-camera, low-cost GPS and map data". In: *13th International IEEE Conference on Intelligent Transportation Systems*. 2010, pp. 1657–1662.

[26] Hang Liu et al. "A precise and robust segmentation-based lidar localization system for automated urban driving". In: *Remote Sensing* 11.11 (2019), p. 1348.

[27] Weixin Lu et al. “L3-net: Towards learning based lidar localization for autonomous driving”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6389–6398.

[28] Wei-Chiu Ma et al. “Exploiting Sparse Semantic HD Maps for Self-Driving Vehicle Localization”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, pp. 5304–5311.

[29] S. Nedevschi et al. “Accurate Ego-Vehicle Global Localization at Intersections Through Alignment of Visual Data With Digital Map”. In: *IEEE Transactions on Intelligent Transportation Systems* 14.2 (2013), pp. 673–687.

[30] X. Qu, B. Soheilian, and N. Paparoditis. “Vehicle localization using mono-camera and geo-referenced traffic signs”. In: *IEEE Intelligent Vehicles Symposium (IV)*. June 2015, pp. 605–610. DOI: 10.1109/IVS.2015.7225751.

[31] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *arXiv:1804.02767* (2018). arXiv: 1804.02767 [cs.CV].

[32] M. Schreiber, C. Knöppel, and U. Franke. “LaneLoc: Lane marking based localization using highly accurate maps”. In: *IEEE Intelligent Vehicles Symposium (IV)*. June 2013, pp. 449–454. DOI: 10.1109/IVS.2013.6629509.

[33] M. Sefati et al. “Improving vehicle localization using semantic and pole-like landmarks”. In: *IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 13–19.

[34] K. Shunsuke, G. Yanlei, and L. Hsu. “GNSS/INS/On-board Camera Integration for Vehicle Self-Localization in Urban Canyon”. In: *IEEE 18th International Conference on Intelligent Transportation Systems*. 2015, pp. 2533–2538.

[35] J. K. Suhr et al. “Sensor Fusion-Based Low-Cost Vehicle Localization System for Complex Urban Environments”. In: *IEEE Transactions on Intelligent Transportation Systems* 18.5 (2017), pp. 1078–1086.

[36] Towaki Takikawa et al. “Gated-SCNN: Gated Shape CNNs for Semantic Segmentation”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 5229–5238.

[37] Z. Tao and P. Bonnifait. "Road invariant Extended Kalman Filter for an enhanced estimation of GPS errors using lane markings". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2015, pp. 3119–3124. DOI: 10.1109/IROS.2015.7353808.

[38] Z. Tao et al. "Mapping and localization using GPS, lane markings and proprioceptive sensors". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Nov. 2013, pp. 406–412. DOI: 10.1109/IROS.2013.6696383.

[39] Wei-Kang Tseng, Angela P. Schoellig, and T. Barfoot. "Self-Calibration of the Offset Between GPS and Semantic Map Frames for Robust Localization". In: *2021 18th Conference on Robots and Vision (CRV)* (2021), pp. 173–180.

[40] Rafael Vivacqua, Raquel Vassallo, and Felipe Martins. "A low cost sensors approach for accurate vehicle localization and autonomous driving application". In: *Sensors* 17.10 (2017), p. 2359.

[41] A. Vu et al. "Real-Time Computer Vision/DGPS-Aided Inertial Navigation System for Lane-Level Vehicle Navigation". In: *IEEE Transactions on Intelligent Transportation Systems* 13.2 (2012), pp. 899–913.

[42] C. Wang et al. "Vehicle Localization at an Intersection Using a Traffic Light Map". In: *IEEE Transactions on Intelligent Transportation Systems* 20.4 (Apr. 2019), pp. 1432–1441. ISSN: 1558-0016. DOI: 10.1109/TITS.2018.2851788.

[43] A. Welzel, P. Reisdorf, and G. Wanielik. "Improving Urban Vehicle Localization with Traffic Sign Recognition". In: *IEEE 18th International Conference on Intelligent Transportation Systems*. Sept. 2015, pp. 2728–2732. DOI: 10.1109/ITSC.2015.438.

[44] R. W. Wolcott and R. M. Eustice. "Fast LIDAR localization using multiresolution Gaussian mixture maps". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 2814–2821.

[45] T. Wu and A. Ranganathan. "Vehicle localization using road markings". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2013, pp. 1185–1190.

[46] Keisuke Yoneda et al. “Lidar scan feature for localization with highly precise 3-D map”. In: *IEEE Intelligent Vehicles Symposium Proceedings*. 2014, pp. 1345–1350.

[47] Fisher Yu et al. “BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.