Self-Localization for Autonomous Driving Using Vector Maps and Multi-Modal Odometry

by

Ehsan Mohammadbagher

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2023

© Ehsan Mohammadbagher 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Ryozo Nagamune Professor, Mechanical Engineering University of British Columbia
Supervisor(s):	Amir Khajepour Professor, Mechanical and Mechatronics Engineering
	Ehsan Hashemi Assistant Professor, Mechanical Engineering University of Alberta
Internal Member:	William Melek Professor, Mechanical and Mechatronics Engineering
Internal Member:	Baris Fidan Professor, Mechanical and Mechatronics Engineering
Internal-External Member:	Nasser Lashgarian Azad Associate Professor, Systems Design Engineering

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

One of the fundamental requirements in automated driving is having accurate vehicle localization. It is because different modules such as motion planning and control require accurate location and heading of the ego-vehicle to navigate within the drivable region safely. Global Navigation Satellite Systems (GNSS) can provide the geolocation of the vehicle in different outdoor environments. However, they suffer from poor observability and even signal loss in GNSS-denied environments such as city canyons. Map-based self-localization systems are the other tools to estimate the pose of the vehicle in known environments. The main purpose of this research is to design a real-time self-localization system for autonomous driving.

To provide short-term constraints over the self-localization system a multi-modal vehicle odometry algorithm is developed that fuses an Inertial Measurement Unit (IMU), a camera, a Lidar, and a GNSS through an Error-State Kalman Filter (ESKF). Additionally, a Machine-Learning (ML)-based odometry algorithm is developed to compensate for the self-localization unavailability through kernel-based regression models that fuse IMU, encoders, and a steering sensor along with recent historical measurement data. The simulation and experimental results demonstrate that the vehicle odometry can be estimated with good accuracy.

Based on the main objective of the thesis, a novel computationally efficient self-localization algorithm is developed that uses geospatial information from High-Definition (HD) maps along with observation of nearby landmarks. This approach uses situation- and uncertainty-aware attention mechanisms to select "suitable" landmarks at any drivable location within the known environment based on their observability and level of uncertainty. By using landmarks that are invariant to seasonal changes and knowing "where to look" proactively, robustness and computational efficiency are improved. The developed localization system is implemented and experimentally evaluated on WATonoBus, the University of Waterloo's autonomous shuttle. The experimental results confirm excellent computational efficiency and good accuracy.

Acknowledgements

First and foremost, I would like to express my profound and sincerest gratitude to my PhD supervisors, Prof. Amir Khajepour and Prof. Ehsan Hashemi, for their outstanding support, encouragement, and guidance for my research and education during my PhD program at the University of Waterloo.

I would like to further thank Dr. Ryozo Nagamune for accepting to hold the external examiner position for my PhD defense examination. I am also thankful to my committee members Dr. William Melek, Dr. Baris Fidan, and Dr. Nasser Lashgarian Azad for agreeing to be in my PhD committee.

I gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC), Ontario Research Fund (ORF), and the financial and technical support of General Motors (GM) in this work. Special thanks to Dr. Bakhtiar Litkouhi and Dr. Alireza Kasaizadeh at the GM Research and Development Center in Warren, MI, USA, and Dr. Hojjat Izadi, and Dr. Mansoor Alghooneh at GM Canadian Technical Center for their technical support and valuable feedback. I would like to thank the technicians in the MVS laboratory, Jeff Graansma, Aaron Sherratt, and Adrian Neill for making the experimental tests possible.

I am also grateful to my dear friends, Reza Hajiloo and Amin Habibnejad Korayem, for their endless friendship. I have had an amazing time at Waterloo because of some great colleagues and friends who have been valuable sources of assistance, affection, and friendship. I highly appreciate my friends including Neel P. Bhatt, Ruihe Zhang, Mobin Khamooshi, Ahmad R. Alghooneh, Pouya Panahandeh, Hamed Jamshidifar, Ahmad Mozaffari, Mehdi Zabihi, Ali Shahidi, and MohammadReza Ghorbani, at the University of Waterloo.

Last but not least, I would like to thank my parents, Ahmad and Maryam, my elder brother, Amin, my sister, Fatemeh, and my little brother, Sajjad, for all the support and endless love through this long journey.

Dedication

Dedicated to my beloved parents.

Table of Contents

List of Figures **xi**

List of Tables **xvi**

1 Introduction **2**

1.1 Motivation 2

1.2 Objectives 3

1.3 Thesis Contributions 4

1.4 Thesis Outline 4

2 Background and Literature **6**

2.1 Introduction 6

2.2 Literature Review on Map-Based Vehicle Self-Localization 6

2.2.1 Uncertainty Modelling 9

2.2.2 Uncertainty in Localization 11

2.3 Literature Review on Vehicle Odometry 13

2.3.1 VINS for Ground Vehicles 15

2.4 Summary 17

3 Vehicle Odometry 19
3.1 Introduction 19
3.2 Model-Based Multi-Modal Vehicle Odometry 20
3.2.1 IMU Measurement Model 21
3.2.2 Decomposition of (True) States into the Nominal States and the Error States 22
3.2.3 Camera Measurement Model 23
3.2.4 GNSS Measurement Model 26
3.2.5 GNSS Alignment 26
3.2.6 Lidar Measurement Model 27
3.2.7 Estimator Structure 28
3.2.8 Results and Discussion 33
3.2.9 GNSS Sensitivity Analysis 39
3.3 Machine-Learning-Based Vehicle Odometry 42
3.3.1 Input Feature Selection 42
3.3.2 Results and Discussion 46
3.4 Summary 46
4 Vehicle Self-localization 51
4.1 Introduction 51
4.2 Landmark-Based Self-Localization 52
4.2.1 Parametrization of Drivable Space 52
4.2.2 Geometrical Model of Landmarks 53
4.2.3 Definition of HD Vector Map 55
4.2.4 Landmark Extraction by Region-of-Interest 56
4.2.5 Robust Model Fitting of Landmarks 57
4.2.6 Measurement and Motion Models 58
4.2.7 Uncertainty-Aware Fusion 60

4.3 Map-Based Lidar Extrinsic Calibration 61
4.4 Uncertainty Quantification 63
4.4.1 Pole Detection Uncertainty 63
4.4.2 Pole Detection Outlier Rejection by Number of Falling Points Test . 65
4.4.3 Plane Detection Uncertainty 66
4.4.4 GNSS Noise Characteristics 66
4.4.5 Situation- and Uncertainty-Aware Attention Mechanism 66
4.5 Summary 67

5 Evaluation and Experimental Studies 69
5.1 Introduction 69
5.2 Experimental Setup 69
5.3 Planimetric Landmark Mapping 70
5.3.1 Data Acquisition and Registration 72
5.3.2 Landmark Extraction and Geometrical Modelling 73
5.3.3 Mapping Results 77
5.4 Result of Map-Based Lidar Extrinsic Calibration 77
5.5 Uncertainty Quantification Results 77
5.5.1 Pole Detection Uncertainty 77
5.5.2 Plane Detection Uncertainty 82
5.6 Real-time Performance 85
5.6.1 Metrics 85
5.6.2 Baseline Algorithm 85
5.7 Localization Results 85
5.7.1 Runtime Analysis 88
5.8 Summary 88

6 Conclusion and Future Work **89**
6.1 Conclusions 89
6.1.1 Future Works 91

References **93**

List of Figures

2.1 Categorization of Uncertainty 9

2.2 High-level schematic of loosely coupled vs tightly-coupled visual-inertial odometry 14

3.1 The goal of odometry is to estimate the changes in the position and orientation over time. Although the goal of self-localization is to localize with respect to the reference frame, having vehicle odometry is necessary to find the position of the vehicle relative to the last position in case the localization becomes unreliable. 19

3.2 Camera measurement block, which receives three consecutive images, detects and matches features, and outputs the measurement vector corresponding to each observed feature. 25

3.3 The GNSS measurements are expressed in a global coordinate system whereas the IMU states are expressed in a local navigation frame. A transformation can align the two coordinate systems together. 27

3.4 Lidar Measurement Block. It receives point clouds in two consecutive frames and does registration in addition to some processing 28

3.5 Factor-graph Representation of the multi-modal odometry estimation problem. The state vector of the odometry at the time t_k is determined based on the edge of visual measurements to the last two states (denoted in yellow), the edge of Lidar measurements and the last state (denoted in green), the edge of IMU measurements and last state (denoted in gray), and the edge of GNSS measurement at the current time (denoted in red). 29

3.6 The schematic diagram of the ESUKF structure. The state is odometry is estimated based on the IMU kinematics and then updated upon receiving measurements from the camera, GNSS and Lidar. 34

3.7 The results of odometry estimation for two scenarios. The scenario in (a) contains a vehicle driving in a highway environment at higher speeds. The scenario in (b) contains driving in an urban environment with lower speeds. 37

3.8 The velocity measurement of scenario b reported from the RTK-GNSS-INS system that is used for ground truth. The accuracy is around 0.05m/s accuracy which corresponds to 0.005m accuracy of displacement given that the time step is 0.1s. 38

3.9 MATLAB/Simulink provides the functionality of generating and running a driving scenario in Unreal Engine. 39

3.10 The odometry results based on simulation experiments. Throughout the first 4 seconds of the maneuver, the estimated states converge close to the ground truth values. 40

3.11 A visual representation of different GNSS measurement signals with different sampling frequencies ($f = 1,\ 2,\ 5$ Hz). 41

3.12 The AIC results for the input feature selection for odometry estimation. . . 44

3.13 The schematic diagram of the machine-learning-based odometry estimator. Given the historical measurement from a database, the odometry is estimated based on the IMU measurements, wheel encoder, and steering angle. 45

3.14 To evaluate the performance of the developed ML-based odometry system, a test dataset is collected from WATonoBus while operating around the ring road at the University of Waterloo campus. The WATonoBus is designed as a low-speed shuttle that operates at low speeds and small steering angles that result in relatively low accelerations. 47

3.15 The results of estimating the vehicle odometry using the developed ML regression model. Accordingly, the developed regression model can estimate the vehicle lateral displacement (Δx), longitudinal displacement (Δy), and yaw angle change ($\Delta \psi$) accurately 48

3.16 The procedure of data collection and model implementation. When the localization is reliable, the odometry system receives localization output along with measurements of internal sensors, to learn odometry. Then at a time that the localization becomes unreliable, the odometry system would use sensor measurements in addition to the collected reference data to relatively self-localize until the self-localization system becomes reliable once again. 49

4.1 A bird-eye view of the ego-vehicle navigating within a known environment. The objective is self-localizing the frame {V} with respect to frame {R}. The self-localization is performed through relatively localizing nearby stationary landmarks, including light poles $\{p_1, p_2, p_3\}$, and building planes $\{\pi_1, \pi_2, \pi_3, \pi_4\}$. 52

4.2 Parametrization of drivable space with a path coordinate system by defining equally distributed discrete s-coordinates along the s-curve. At each s-coordinate, only a subset of landmarks is observable. 53

4.3 A vertical plane which is modeled as a line in xy-plane is parametrized by its distance to the origin and its angle 54

4.4 Definition of the Region-of-interest around a façade plane. A cuboid is constructed that is aligned with the plane and covers the points that fall onto the plane with added margin. 56

4.5 The normal distance to the curb is computed by using a curb unit tangent vector and a point on the curb 59

4.6 Range and bearing measurements for pole detection. Different levels of uncertainty is associated with pole detection as the vehicle observes the pole at various bearing angles and ranges 64

5.1 System diagram of HD-LOC. It contains two ROS nodes that communicate within the ROS environments. The ROIs are formed around the expected location of landmarks according to the HD Vector map and *a priori* self-localization of the ego-vehicle. Landmark model fitting is performed over ROI clusters. Finally, the landmark models in the Lidar frame and in the reference frame are used in the least-square problem to estimate the self-location of the vehicle. 70

5.2 The WATonoBus platform is used for experimental studies of the developed self-localization system. The vehicle is equipped with a long-range 32-beam Lidar with a vertical Field of View (FOV) of 40° and horizontal FOV of 180° with up to 200 m range and ±3 cm accuracy. A short-range 32-beam blind spot Lidar is used with 360° horizontal FOV, 90° vertical FOV, and a range of 30m with up to ±3 cm typical range accuracy. The short-range Lidar is used for scanning nearby curbs. The long-range Lidar scans building planes and poles. The vehicle is equipped with an RTK-GNSS-INS with centimeter-level accuracy that provides the ground truth information of the self-localization. 71

5.3 The scan of the long-range Lidar (left) and short-range blind spot Lidar (right). A Long-range Lidar is used for scanning bigger landmarks, including building planes and light poles, and a short-range Lidar is used to scan nearby curbs. 72

5.4 Result of Point cloud collection captured at the Ring Road at the University of Waterloo Campus. It contains scans of Lidar that are registered into a fixed local reference frame. The points that correspond to the ground are removed based on thresholding. 73

5.5 Manual selection and labeling of suitable landmarks. A landmark is considered suitable if it is observable from a portion of the road. Poles are represented as a single point and planes are represented by their two endpoints. 74

5.6 Bird-eye-view of a Lidar scan (denoted in blue) and detected light poles p_1 and p_2 and building planes π_1 and π_2 (denoted in red) 75

5.7 The process of collecting curb control points. A curb detection module provides the nearby curb points (denoted in red). A curb control point ($\mathbf{c}_n$) is obtained by projecting the nearest s-coordinate ($\mathbf{s}_n$) on the fitted curb line (l). 76

5.8 The result of mapping the landmarks at the University of Waterloo Ring Road, including 107 light poles (blue), 177 building planes (black), and curb points (red) with 2.63 km of length. The location of landmarks is expressed in the navigation frame (The local Easting axis (x) is perpendicular to gravity, perpendicular to the local Northing axis and is in the east direction. The local Northing axis (y) is perpendicular to the gravity vector and in the direction of the north pole along the earth's surface. The up axis (z) is co-axial with the gravity vector and positive in the up direction.) 78

5.9 By minimizing the distance between the observed light poles (ovals) and their expected location acquired from the HD vector map (squares), the Lidar to Vehicle extrinsic calibration is refined. The conformity of the actual and expected location of the poles confirms the effectiveness of the developed Lidar calibration algorithm. 79

5.10 Uncertainty quantification of pole detection k-NN regression of residuals (a) the bearing residuals for pole detection (b) the mean and standard deviation of bearing residuals for pole detection. The decreasing standard deviation shows the heteroskedasticity of uncertainty in estimating the pole bearing. 80

5.11 Uncertainty quantification of pole detection k-NN regression of residuals (a) the range residuals for pole detection (b) the mean and standard deviation of range residuals for pole detection. The increasing standard deviation shows the heteroskedasticity of uncertainty in estimating the pole range. 81

5.12 The result of pole outlier rejection based on regression over the number of falling points on the poles versus the range. The number of falling points is decreasing because when detecting a pole from farther away, a smaller subset of Lidar laser beams collides with the body of the pole. Any detected pole that is outside of a 97.5% confidence band is considered an outlier. . . 83

5.13 The results for (a) variation of plane distance error versus the number of points that fall on a plane and (b) the variation of the standard deviation of the distance error versus distance to the plane. 84

5.14 Longitudinal error of self-localization over the Ring Road. Accordingly, HD-LOC estimates the self-localization of WATonoBus with good consistency over the entire Ring Road. On the other hand, the accuracy of NDT map matching deteriorates in some part of the Ring Road. 86

5.15 The environment contains mostly temporarily parked vehicles while lacking enough longitudinal and lateral excitation. HD-LOC solely attends to the light poles (denoted in red) to localize the vehicle while filtering out the rest of the point cloud. 87

List of Tables

2.1 A summary of studies using attention mechanisms 12

3.2 RMSE of Vehicle Odometry estimation for the real world experiments . . . 38

3.3 The result for analyzing the sensitivity of odometry estimation accuracy to the sampling frequency of GNSS. The sampling frequency of 1Hz is considered the baseline for the comparison. 40

3.4 The result for analyzing the sensitivity of odometry estimation accuracy to the standard deviation of GNSS noise. The noise level of 1m is considered the baseline for the comparison. 41

3.5 RMSE of Vehicle Odometry Prediction 49

5.1 Quantitative comparison of NDT results with the HD-LOC 87

Notation Conventions

Throughout the thesis, scalars are denoted by small letters (such as a) vectors are denoted by small and bold letters (e.g. $\mathbf{b}$), matrices are denoted by capital bold letters (e.g. $\mathbf{A}$), and points are denoted by capital letters (e.g. M). Coordinate frames are denoted by a capital letter inside curly brackets (e.g. $\{V\}$). If $\{A\}$ and $\{B\}$ are two arbitrary frames in the space, the rigid body transformation that transforms a vector from $\{A\}$ to $\{B\}$ is a transformation in (i.e. the transformation in the Lie group of SE(3)) is denoted as follows:

$$
{}^{B}\mathbf{T}_{A} = \begin{bmatrix} {}^{B}\mathbf{R}_{A} & {}^{B}\mathbf{p}_{A} \\ 0,0,0 & 1 \end{bmatrix} \tag{1}
$$

where ${}^{B}\mathbf{R}_{A}$ is a 3D rotation matrix from $\{A\}$ to $\{B\}$ and ${}^{B}\mathbf{p}_{A}$ is the position of $\{A\}$ with respect to $\{B\}$ while expressed in $\{B\}$.

Chapter 1

Introduction

In automated driving, a vehicle localization system is necessary for the safe operation of other modules like motion planning and control systems. Although the geolocation of the vehicle in various outside settings can be provided by GNSS, however, GNSS suffers from poor observability in GNSS-denied environments. Therefore, localization systems are needed for estimating the accurate location and heading of the ego-vehicle.

1.1 Motivation

Given the availability of maps of the environment, map-based localization systems have become a viable option for automated driving. They fuse multiple landmarks information with the map for solving the localization problem. However, fusing several noisy measurements in a probabilistic state estimation framework (e.g. Kalman filter) requires proper uncertainty models of the measurements. In some sensors, the source of noise may arise from the physical sensing mechanism of the sensor itself. For example, the noise of an IMU, which measures the angular velocities and linear accelerations by measuring physical quantities, is statistically interpretable and determined by the manufacturer. However, sensors such as Lidars and cameras that require data processing to obtain an estimate or measure a physical quantity (e.g. distance to the landmark) need increasingly complex and potentially hard-to-specify uncertainty models. It is because numerous factors such as the presence of dynamic objects, environment structure, level of salient features, the mutual pose of the sensor to the landmark, etc., contribute to the uncertainty of processing the data to arrive at a physical estimation. Therefore, there is a need to develop some uncertainty models that maps the low-level raw measurements into their uncertainty level. Having the uncertainty model will also enable the system to select a subset of the least uncertain (most informative) measurements, which not only increases the estimation accuracy but also reduces the computational complexity of the localization.

A map-based localization system relies on the availability of salient landmarks in the environment. However, challenging conditions such as the sparsity of landmarks in some areas or occlusion might lead to localization failure that needs to be addressed by estimating the vehicle's ego-motion (i.e. change in position over time) as a short-term solution until the localization becomes reliable. Odometry systems, such as wheel odometry, enable vehicles to navigate in the environment when localization is not available. Visual Odometry (VO), particularly, has emerged as a viable substitute for wheel odometry, since it delivers more precise trajectories with relative position errors (ranging from 0.1 to 2% [1]) than wheel odometry because it is not impacted by wheel slip in challenging road conditions. VO techniques, however, fall short in low-textured settings, poor lighting conditions, and circumstances when visibility is impaired by adverse weather conditions. Utilizing other sensor modalities, such as IMU, GNSS, and Lidar, can help mitigate some of these constraints. The information these sensors provide is often fused in a probabilistic framework to improve the observability of the problem where the sensors work cooperatively to provide information that is impossible from individual ones. For instance, in the monocular Visual-Inertial Navigation Systems (VINS), the motion scale and the IMU's biases become observable. Moreover, it is desirable to increase the redundancy to make the whole system robust to sensors' failures. Finally, sensor modalities have different noise characteristics; hence they can work in a complementary fashion to compensate for each other's noisy measurements. For instance, in the Lidar-radar-camera odometry system, the Lidar complements the camera in low-light conditions, and the radar complements the camera and Lidar in adverse weather conditions. A reliable fusion algorithm that can adequately account for individual sensor characteristics has been a challenge in multi-sensor localization.

1.2 Objectives

The main objective of this thesis is to develop a localization algorithm by achieving the following sub-objectives:

1. Developing an efficient map-based vehicle localization system that takes into account the uncertainty of environmental landmarks.

2. Developing a Multi-modal vehicle odometry system to aid the localization system by estimating the vehicle's ego-motion.

1.3 Thesis Contributions

The main contributions of this thesis are the following:

1. Developing an efficient HD-map-based landmark detection and geometrical modeling.
2. Quantifying and incorporating landmarks' uncertainties into the landmark-based localization problem.
3. Developing a multi-modal odometry system based on a tightly-coupled fusion of camera and IMU measurements along with a loosely-coupled fusion of Lidar and GNSS measurements.
4. Developing a novel learning-based vehicle odometry algorithm that uses the vehicle's conventional sensors to aid the self-localization system.

1.4 Thesis Outline

Chapter 2 provides a literature review of techniques/algorithms used for estimating vehicle self-localization and vehicle odometry. The main features, limitations, and assumptions for each estimation method are discussed.

In Chapter 3, vehicle odometry systems are proposed. In Section 3.2, a model-based multi-modal vehicle odometry system is proposed to improve vehicle self-localization in general settings by estimating the vehicle ego-motion. This algorithm fuses measurements of different sensors, including IMU, camera, GNSS, and Lidar, to estimate the odometry of the vehicle (i.e. short-term relative displacement and rotation of the vehicle) through an Error-State Kalman Filter (ESKF). In this section, the proposed model-based odometry estimator is validated by experimental and realistic simulation experiments. In Section 3.3, a machine-learning-based odometry estimation system is developed that uses recent historical data to compensate for the unavailability of reliable self-localization. The performance of the developed odometry system is validated using experimental results with discussion in Section 3.3.

Chapter 4 presents a new self-localization algorithm that uses geometric information of reliable landmarks while using the developed situation- and uncertainty-aware attention mechanisms to fuse various sources of information according to the uncertainty level. Section 4.2 formulates the self-localization task as an optimization problem. It contains the proposed approach for modeling and detecting landmarks along with the integration of motion models. A landmark-based extrinsic calibration of Lidar is developed and discussed in Section 4.3. Section 4.4 presents the developed uncertainty quantification procedure for landmarks detection along with the developed attention mechanisms.

Chapter 5 provides the results and discussions for implementing the self-localization system using real-world experiments. The details of the experimental setup are presented in Section 5.2. In Section 5.3, the procedure and the results of obtaining the light HD Vector map of landmarks are presented. The result of the developed landmark-based algorithm for Lidar extrinsic calibration is presented in Section 5.4. The results of uncertainty quantification for landmarks are discussed in Section 5.5. The real-time performance of the developed self-localization algorithm is presented in Section 5.6. The qualitative and quantitative results for the developed self-localization system are provided in Section 5.7.

Finally, conclusions are made in Chapter 6 with suggestions for future works.

Chapter 2

Background and Literature

2.1 Introduction

While there is a great catalog of work concerned with map-based self-localization, this chapter examines related approaches that incorporate some uncertainty models in multi-sensor fusion in self-localization as well as vehicle odometry. The main features, limitations, and assumptions for each estimation method are discussed. Additionally, some background materials and terminologies that are used in the thesis will be covered.

This chapter is organized as follows: In Section 2.2, the literature on map-based vehicle self-localization and specifically uncertainty-aware approaches are reviewed. Moreover, the details of vehicle odometry techniques focusing on visual-inertial navigation systems are reviewed in Section 2.3. Conclusions are provided in Section 2.4.

2.2 Literature Review on Map-Based Vehicle Self-Localization

According to the availability of maps of the environment, autonomous vehicles may use map-free or map-based approaches. In the map-free approach, a map that provides accurate information about the road features is not available. In this approach, a perception module must locate important nearby objects/features relative to the ego-vehicle while expressed in the vehicle frame. This is done by fusing the sensors' online measurements through the perception module in real-time. In this approach, the perception unit must provide the drivable space by detecting static road boundaries accurately and reliably such that the controller can drive the vehicle safely inside the drivable space [2]. Having a vehicle-centric self-localization is all that is needed for the safe operation of an Autonomous Vehicle (AV). In the map-based approach, however, the accurate location of static road features is known in the map frame. In this approach, an accurate self-localization in the map frame is necessary for knowing where the road boundary is around the vehicle to obtain the drivable space without the need to perceive it in real-time. Additionally, the location of other objects with a fair amount of accuracy is needed for obstacle avoidance [3]. In comparison, map-based operation of AVs needs an accurate global self-localization. On the other hand, a map-free approach needs an accurate and highly reliable real-time perception system.

Recent developments in developing HD maps of urban environments have resulted in using map-based self-localization algorithms [4, 5, 6, 7, 8]. The idea is to compare information from existing HD maps with the observed landmarks in the sensors' online measurements. Measurements from different sensor modalities have been used in the literature for the self-localization task.

The camera provides 2D location and RGB information of environmental features in the image domain. Visual features in an image can be used for map-based self-localization [9, 10, 11]. However, they are not robust to appearance changes due to different lighting conditions, seasonal changes, and lack of enough performance in texture-less environments. Additionally, the nonlinear projection on the image plane makes the problem highly nonlinear and complex to solve. Moreover, imperfect calibration models (e.g. due to image distortion) can result in additional systematic uncertainty to the system. It is also important to note that the robustness of visual features to small movements of the camera makes them a good choice for odometry estimation of the vehicle.

Lidar sensors provide geometric information about the environment in form of the 3D location of point clouds and their intensity level. Processing Lidar measurement is simpler since it is given in vector space and no nonlinear projection is involved. Moreover, scans of a 3D structure are less affected by view angle and illumination variance, which makes them good for long-term localization without the need to update the map frequently.

One of the popular approaches for HD-map-based self-localization is to register online Lidar 3D point clouds with pre-recorded 3D HD point cloud maps to perform self-localization. Iterative Closest Points (ICP) and Normal Distribution Transform (NDT) are two well-used map-matching algorithms [6]. However, processing large-size 3D HD point cloud maps and large real-time Lidar data streams makes them computationally expensive with high storage needs. The idea of matching sub-maps is introduced in [12] to account for changes in the environment; however, it yields a significant increase in the overall computation time. Additionally, 3D HD point cloud maps are prone to change over time due to constructions that need to update the map frequently.

One potential approach to reduce the computational time and the storage need of map-based self-localization systems is to use sparse landmarks by filtering out unnecessary information from online perception data and HD maps while focusing on a subset of landmarks for the self-localization task. Landmark-based localization is a very well-studied problem [13]. Visual semantic cues on the road surface such as lane markings can be used to construct a map for self-localization [4, 8]. Authors in [8] propose a localization algorithm for parking lots based on HD vector maps. In this approach, road markings are extracted using semantic segmentation of bird-eye-view of multiple cameras. Then detected and matched landmarks are fused with IMU in an error-state Kalman filter. Authors in [4] propose a tightly-coupled fusion of visual odometry and vector HD maps. The algorithm uses visual and vector HD map landmarks. The disadvantage of using visual landmarks is that their appearance may change over time by repainting or by being covered by snow. Road signs can be also used as a sparse semantic map for self-localization [14]. However, the effectiveness of such approaches depends on the availability of the signs. There also exist works on vector-map-based self-localization using Lidar measurements.

Urban areas contain a density of various landmarks such as buildings, light poles, road markings, sidewalks, vegetation, traffic signs, etc., which can be used for the self-localization task. Buildings and light poles are the two types of static landmarks that are of higher quality for self-localization [7, 15]. Buildings and light poles are relatively tall; therefore, they are more likely to be visible from farther distances while not being obstructed by other objects such as trucks and buses. As a result, the perception system can observe them throughout a larger section of the road with less chance of losing the line of sight. Second, the geometry and the appearance of buildings and light poles are less prone to seasonal and temporal changes in contrast to other landmarks such as trees and road markings whose geometry and appearance may vary over time. Therefore, building planes and poles can be observed with consistent simple geometry and appearance over time. On the other hand, instead of representing buildings and light poles by point clouds, they can be represented by some simple geometrical models as they are usually in simple geometrical shapes. Buildings that are constructed mostly by vertical flat walls can be represented by a set of planes and light poles that have a cylinder-shaped body and can be represented by lines. As a result, processing their geometrical models is computationally more cost-effective and requires less space to store compared to unstructured landmarks in the environment.

2.2.1 Uncertainty Modelling

Uncertainty refers to situations involving imperfect or unknown knowledge. In the mathematical modeling of physical phenomena, the goal is to devise a model with a specific structure and set of parameters that can predict the desired output given observation data. There might be some situations that lead to uncertainty on the model's prediction values (i.e. low confidence) [16]:

- *Noisy data*: the observed data might be noisy, which leads to the *aleatoric* uncertainty
- *Uncertainty in model parameters*: the designer might be uncertain in selecting the best set of parameters for the mode
- *Structure uncertainty*: the designer might be unsure which model structure to use

The latter two uncertainties can be grouped under *model uncertainty* (also referred to as *epistemic uncertainty*). The aleatoric and epistemic uncertainties are involved in producing *predictive uncertainty*, the uncertainty in the model's prediction.

The noise in the data can be classified into two groups, *homoscedastic aleatoric uncertainty*, when all the observation data involve identical observation noise, and *heteroscedastic aleatoric uncertainty* when the observation noise can vary with observation data [17]. Figure 2.1 shows the categorization of uncertainty.

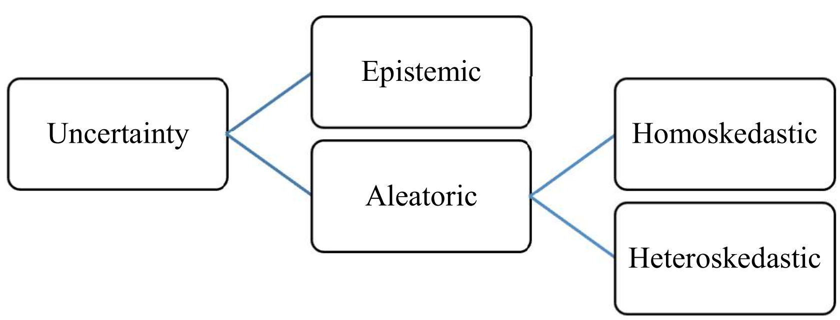


Figure 2.1: Categorization of Uncertainty

An uncertainty model is a function that maps the inputs from the measurement space to the prediction uncertainty (usually formed as a covariance matrix) associated with the model's output in the prediction space.

In regression analysis, heteroscedasticity (sometimes spelled heteroskedasticity) refers to the unequal scatter of residuals or error terms. Specifically, it refers to the case where there is a systematic change in the spread of the residuals over the range of measured values. Heteroscedasticity in the data is problematic in ordinary least squares (OLS) regression since it assumes that the residuals come from a population that has homoscedasticity, which means constant variance.

In some sensors, sources of noise may arise from the physical mechanisms of the sensor itself. For instance, an IMU sensor has an almost constant accelerometer and gyro noise characteristics because it measures the angular velocity and linear acceleration by directly measuring a physical quantity. Therefore, IMU noise is homoscedastic and is usually given by the manufacturer who had carried out rigorous experiments. Hence, knowing this information suffices for the algorithm designer to choose a sensor that meets the desired noise level. Additionally, there is no further need to change the noise characteristic of the sensor in real-time. In more complicated sensors such as vision and Lidar sensors, the model of uncertainty is more complex. The reason is two-fold; first, the Lidar measurement is a multi-dimensional vector of hundreds of ranges and intensities, which makes it complex and hard to be mapped into its uncertainty. Second, the vehicle itself undergoes different maneuvers in different environments with different levels of light, dynamic/static objects, structured/unstructured objects, etc. Therefore, the uncertainty level in the Lidar is non-uniform in different conditions, so it is heteroskedastic.

Visual attention, in general, is a bio-inspired mechanism, which is selecting the relevant data from the whole perception information. Authors in [18] comprehensively reviewed visual attention mechanisms with emphasis on the biological aspects. Attention mechanisms can be divided into two categories; *hard attention mechanisms* that filter out the uncertain region of the measurement completely; and *Soft attention mechanisms* that attend non-uniformly to different regions of the uncertain measurement.

In visual localization, there is often a great amount of redundancy in the camera measurements, which gives the algorithm the flexibility to attend to only a sub-region of the whole image in real-time while the localization remains observable; for instance, having five-point correspondences across two image frames makes the estimation of relative rotation and translation observable [19]. Therefore, the system can filter out uncertain regions of the image by using a visual hard attention mechanism. On the other hand, the system can attend to different regions of the image non-uniformly, based on their level of uncertainty; this is known as the visual soft attention mechanism.

2.2.2 Uncertainty in Localization

Some of the localization algorithms considered using attention mechanisms to meet two objectives, to reduce computational complexity, and to neglect uncertain measurements to improve localization accuracy. The following will particularly go over the recent visual SLAM, which uses hard and soft attentional mechanisms for the aforementioned objectives.

Visual hard attention, often known as active feature selection, has been used in most of the works to enhance the performance of visual localization and odometry algorithms. Authors in [20] and [21] use a trained heuristic model of the quality of the visual features by mapping to a scalar value that quantifies the features' quality for the localization; in [21] the measurement covariance associated with each observation is rescaled accordingly. Authors in [22] propose a method to select salient landmarks and constructed a topological map. Authors in [23] use an appearance-based metric of visual saliency (features that "stand out") in the Simultaneous Localization and Mapping (SLAM) loop closing. Authors in [24] select a minimal subset of landmarks in a graph based on a co-visibility criterion which is to be seen from multiple camera frames. Authors in [25] propose an attention module for landmark selection and active gaze control; In their work, feature selection includes a bottom-up and top-down attention mechanism, for considering features' saliency and task performance, respectively. Authors in [26] propose a landmark selection mechanism based on the covered area to reduce the map size. Authors in [27] propose a mechanism for landmarks subset selection through a reinforcement learning algorithm. Authors in [28] and [29] incorporate prior information to inform feature matching, and to increase the efficiency consequently.

On the other hand, Visual soft attention mechanisms have been used more recently in visual odometry and SLAM. Authors in [21] consider multiple criteria, including angular velocity and acceleration as the level of image degradation, local image entropy as the quality of textures, a blur metric introduced in [30], optical flow variance score as a metric for moving objects, and image frequency composition as the level of texture in the image. They use these sets of criteria as the features to learn the uncertainty using a k-nearest neighborhood (KNN) approach. They further develop this approach in [31], in which they use the same set of features in a kernel function to estimate the covariance matrix in the least square problem in visual odometry. Authors in [32], convert an end-to-end visual re-localization neural network [33] into a Bayesian neural network that can estimate the predictive uncertainty in addition to the network's point estimate. They average Monte Carlo dropout samples from the posterior of the network's weights to estimate the network's output distribution and noise characteristics. More recently, authors in [34] propose an end-to-end visual-Lidar SLAM that incorporates a heuristically designed attention layer.

Table 2.1: A summary of studies using attention mechanisms

Application	**Attention Mechanism**	**Criteria**	**Metric**
Visual Odometry	Hard	Prior-informed Matching	Vehicle-environment configuration (depth) [38]
			Motion-model [28, 29, 39]
		Distribution of the image	Bucketing [38, 40]
			Mutual information [41]
			Observability [42]
			Orthogonality Index [43]
		Quality	Semantic Information [36, 44, 45]
			Tracking age [38]
			Prediction Age [39]
		Outlier rejection	Normalized reprojection error [46]
		Localization error	Reinforcement learning [27]
	Soft	Quality	Angular velocity, acceleration, image entropy, etc. [21, 47]
		Data-driven	Deep features [33, 32]
Visual Localization and SLAM	Hard	Topology	Well-distributed in the environment [26]
			Higher Co-visibility [9, 24]
			Information gain [48]
		Quality	Uniqueness [21, 22, 23, 25]
			Static/dynamic [45, 44]
Other Applications	Soft	Data-driven	Deep features [35, 49, 37]

Soft attention has been of interest in other applications as well. Authors in [35] provide an attention layer in a deep learning framework for probabilistic pixel-wise semantic segmentation. Authors in [36] propose an attention module that is used in a multi-task learning structure, including semantic segmentation, depth estimation, and surface normal estimation. Authors in [37] propose an attention-aware temporal weighted convolutional neural network (CNN) for the action recognition problem. Table 2.1 summarizes the aforementioned works done in active feature selection and visual attention mechanisms.

According to Table 2.1, some research studies use various metrics as features that can interpret the heteroskedastic uncertainty of the visual information. These are some domain-knowledge-based features that try to express the uncertainty of the visual measurements based on various criteria. One can interpret the idea of including more features (metrics) as an attempt to reduce the epistemic uncertainty of the system (the uncertainty in the model structure and parameters).

Approaches in Table 2.1, do not cover multiple essential ideas that can be considered as gaps in the literature. Developing a soft attention mechanism in the localization problem is one of the main objectives of the proposed thesis, which has not been done for Lidar localization, according to Table 2.1. Moreover, most of the works, considered vehicle-oriented measurements for the attention mechanism while ignoring the effect of vehicle-map mutual configuration. There is a need to include the map topology from the vehicle's perspective (e.g. depth of the features, angle of view, etc.) to proactively attend to uncertain regions of the environment.

One approach to improve the overall uncertainty of the localization problem has been using sensor fusion by fusing the output of multiple sensor modalities. There have been two general paradigms in multi-sensor fusion in SLAM, tightly-coupled and loosely-coupled approaches. In the loosely-coupled state estimation approach, the output of different state estimators based on different sensor modalities is fused. On the other hand, in tightly-coupled fusion, the states that are being estimated in the estimation problem include not only the pose of the vehicle but also the internal states of the different sensor modalities (e.g. IMU bias); therefore, they outperform the estimation accuracy but need more implementation effort. Figure 2.2 illustrates a high-level schematic of visual-inertial odometry in these approaches.

From the optimization perspective, localization algorithms are classified into filtering and optimization (Maximum a Posteriori, fix-lag smoothing) approaches. In the filtering approach, a window of recent states is estimated in a Kalman filter [50, 51]. On the other hand, in the optimization-based approach, a sparse set of states is estimated through a nonlinear optimization problem which yields better real-time performance [52, 53].

2.3 Literature Review on Vehicle Odometry

In recent years, visual odometry has been developed significantly due to its low cost, small size, and easy hardware layout properties [9, 53, 54, 55, 56]. However, the weak robustness of visual measurements in challenging environments with high uncertainty has been a problem of pure visual odometry and SLAM. One solution to this problem is fusing an inertial measurement unit (IMU) with the camera, which leads to the visual-inertial navigation system (VINS) and SLAM algorithms [52, 51, 57, 10]. In VINS, the motion

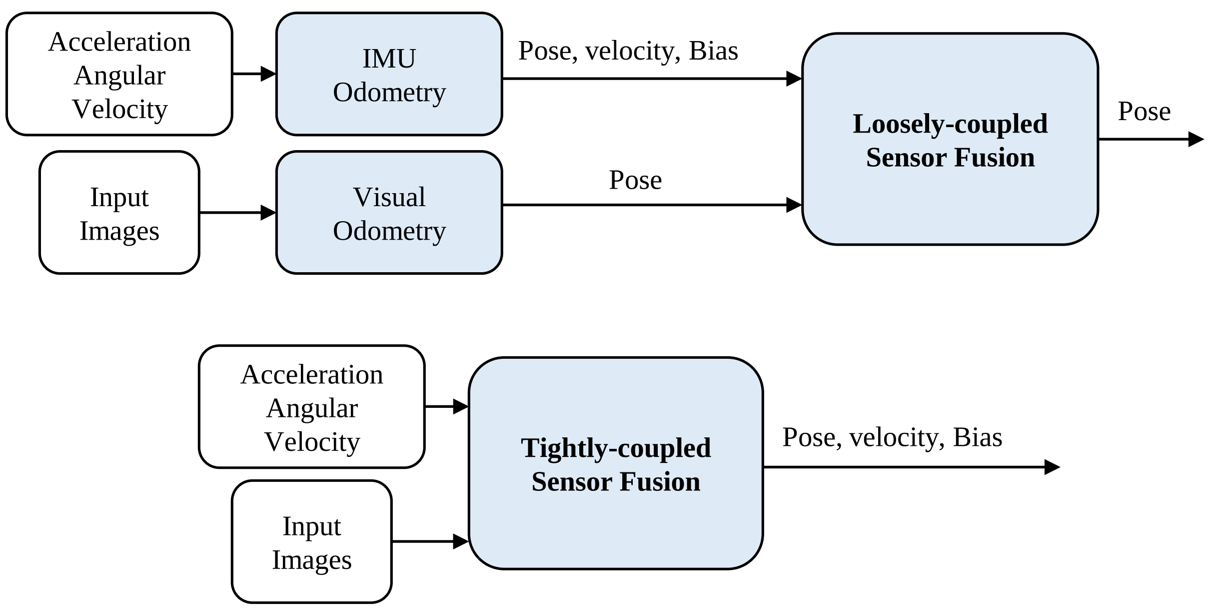


Figure 2.2: High-level schematic of loosely coupled vs tightly-coupled visual-inertial odometry

scale of monocular SLAM becomes observable and the inertial readings impose some short-term motion constraints that yield more robust performance under uncertain low-quality visual readings, such as fast motion or strong illumination change.

In [51], a multi-state constraint Kalman filter (MSCKF) is proposed that uses an error-state extended Kalman filter to estimate the states using an IMU and a monocular camera. In this work, the 3D position of features needs to be estimated through a least square problem. In [58] a visual-inertial odometry architecture is developed that uses a trifocal tensor as the constraint between three camera images. Therefore, no estimation of the 3D position of features is required which reduces the computational complexity. The fusion of the GNSS with VINS is also a popular topic. In [59], an MSCKF-based estimator is used to optimally fuse inertial, camera, and asynchronous GNSS measurements. In [60], global positional information is fused with visual-inertial measurement in a tightly coupled optimization-based estimator. In this work, the states contain the vehicle's states and the position of 3D landmarks, making the system computationally heavy if the position of 3D visual landmarks is not needed.

The most similar approach to this thesis is [61], in which they fuse IMU, GNSS, radar, Lidar, and camera through an error-state extended Kalman filter (ESEKF). They used a deep neural network, PWC-net, to estimate the optical flow followed by essential matrix extraction. Therefore, it is not CPU real-time. On the other hand, it uses two camera images (instead of the three camera images that are used in this work) to do the visual odometry. A one-step numerical differentiation is done to fuse the output of Visual and Lidar odometry to obtain the estimated velocity, which is not accurate and is prone to noise. On the other hand, in this work, since the last pose of the vehicle is included in the state variables, the Lidar measurements are fused based on the relative position of the current vehicle frame and the last vehicle frame. Additionally, they do not include IMU biases in the state variables, which can lead to significant drift. They used a 3DOF IMU through a 6DOF kinematic model. Therefore, in highly dynamic driving or on roads with bank angles and grades, the readings of 3DOF IMU are not reliable because it contains a large projection of gravitational acceleration.

2.3.1 VINS for Ground Vehicles

When it comes to ground vehicle applications, there are still some challenges in using VINS. It has been shown that VINS algorithms suffer from weak observability in a degenerate type of ground vehicle's motion [62]. A vehicle approximately has a planar motion, usually over an arc or straight line. Therefore, IMU is mostly measuring only 3 degrees of freedom

(DOF) of the vehicle motion. This 3 DOF often degenerates to 1 Degree of Freedom (DOF) or 2 DOF in urban and highway driving conditions. As a result, when the vehicle has no acceleration (going in a straight line with constant speed), the motion scale and the acceleration biases become unobservable [62]. Nevertheless, wheel odometry can make the motion scale observable; however, there is a need to fully exploit the vehicle motion characteristics in the estimation problem.

The aforementioned challenges of VINS can be mitigated by incorporating the vehicle's motion dynamics to limit the solution space the vehicle can undergo. Existing literature shows that knowledge of robot dynamics, especially vehicle dynamics, can improve estimation. However, using vehicle dynamics has been limited mostly to the filtering algorithms, such as Kalman filters [63].

On the other hand, in optimization-based approaches, utilizing dynamics in the estimation is challenging in real-time for two reasons; first, the rate of the dynamic measurements, e.g. steering wheel angle, is fast compared to the speed of the optimization step, which implies that measurements need to be processed at a faster rate than the optimizer can handle. Second, those measurements are highly coupled with the states, meaning that multiple measurements cannot easily be integrated, as the estimates of the states presumably must change at each optimization step.

To fuse high-frequency sampled inertial measurement with low-frequency sampled visual measurement of the camera, the pre-integration method has been extensively used by researchers [64, 57] after it was introduced in [65]. In this approach choosing the body frame as the reference frame decouples measurement from the estimation states. As a result, it enables integrating the inertial measurement data once and using the integration result in every optimization step without the need for recalculation. Given the fact that inertial measurements essentially provide odometry data (the local motion of the vehicle), one can generalize the idea of pre-integration to any odometry-based information, specifically dynamical equations of the motion. Pre-integrated dynamic factors were introduced in [47] for the visual-inertial odometry of unmanned aerial vehicles (UAVs). It utilized a point-mass quadrotor model to create dynamic factors for transitional dynamics only and used IMU measurement for the rotational dynamics factor. This work was followed by [66] by jointly estimating UAV states and external forces. By estimating external forces, the method introduced in this work was able to reduce the discrepancy between motion prediction based on UAV dynamics and actual motion.

In addition to UAVs, the dynamics of the vehicle have been also considered in some VINS algorithms. In some earlier works [67, 68, 69], while assuming the planar motion of the vehicle, the motion is estimated by solving a homography matrix. In [70, 71] a one-point random sample consensus (RANSAC) outlier rejection based on the vehicle's non-holonomic constraints is introduced to increase the estimation efficiency. The Ackermann vehicle model is used in [72] to recover the scale of monocular visual odometry. However, this simple vehicle model did not consider the influence of the tire cornering stiffness and sideslip angles. The above methods specifically designed for vehicle motion estimation only focus on adding motion constraints to visual odometry or data association, rather than adding vehicle model constraints to the optimization backend. In mVINS [73], wheel-encoder measurements were incorporated into VINS trying to properly model the vehicle's almost-planar motion; however, it didn't benefit fully from vehicle dynamics by tightly coupling vehicle motion information with other measurements.

More recently, authors in [74] introduced the idea of tightly-coupled integration of vehicle dynamics into VINS. They applied the pre-integration idea to the application of ground vehicles. In this work, by considering the availability of high-frequency vehicle inputs (i.e. longitudinal velocity and steering wheel angle over the CAN Bus), the 2D lateral vehicle bicycle model is implemented under the steady-state assumption. This approach does not consider the epistemic uncertainties originating from the uncertainties in vehicle model structure and parameters, nonlinearities in the tire model, vehicle non-planar motions, and external disturbances from the road.

2.4 Summary

To evaluate the novelty of the proposed thesis, this chapter reviewed previous research studies in multi-sensor fusion localization from the uncertainty-awareness perspective. General definitions and a short description of different types of uncertainties were followed by reviewing the localization algorithms that take into account measurement uncertainty to some extent by using soft and hard attention mechanisms. It was shown that there is a need to first, quantify the uncertainty of measurements, and second, incorporate the quantified uncertainty into multi-modal data fusion localization, which has not been fully discovered in the literature so far.

Additionally, the general paradigms in vehicle odometry were reviewed with special consideration to the visual-inertial navigation system (VINS). The recent progress in incorporating vehicle dynamics into VINS was discussed and the need to consider a data-driven approach was discussed, something that has not been considered in the literature yet.

According to the reviewed literature and the thesis objectives, some new approaches have been developed, which are discussed in the following chapters. A new approach for model-based odometry and learning-based odometry is introduced in Chapter 3. In Chapter 4, a vehicle self-localization approach is developed that takes into account measurement uncertainties in the problem.

Chapter 3

Vehicle Odometry

3.1 Introduction

Given the vehicle's last position and measurements, the objective of vehicle odometry is to find the next position of the vehicle. Vehicle odometry plays an important role in self-localization systems, especially when an absolute self-localization measurement (e.g. GNSS) is not reliable. A variety of approaches have been proposed for odometry estimation in the literature.

Conventional vehicle state estimators who have access just to the standard vehicle dynamics sensors suffer from weak observability or non-observability due to the highly nonlinear behavior of the vehicle, particularly on slippery roads or while driving at the limits of vehicle handling. Currently, vehicles with automated driving features are equipped with

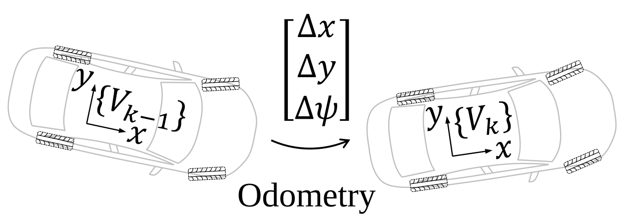


Figure 3.1: The goal of odometry is to estimate the changes in the position and orientation over time. Although the goal of self-localization is to localize with respect to the reference frame, having vehicle odometry is necessary to find the position of the vehicle relative to the last position in case the localization becomes unreliable.

additional types of sensors, such as GNSS, cameras, Lidar, radar, etc, that can be used to estimate the vehicle states with higher accuracy and reliability. On the other hand, model-based observers suffer from parameter mismatch and drift between the model's parameters and the actual vehicle's parameters. On the other hand, kinematic-based observers are prone to drifts due to time-varying biases in the IMU sensor.

In this chapter, a tightly-coupled estimation approach is used that involves the IMU's biases in the state vector, which is estimated through fusion with other exteroceptive (i.e. environmental) sensors. Therefore, the observer does not rely on vehicle dynamics parameters and does not suffer from drift due to the IMU biases. This odometry system is designed to aid the self-localization system by imposing constraints on the displacement of the vehicle. Additionally, a machine learning-based odometry system is developed that uses proprioceptive (i.e. internal) sensors that account for the measurement uncertainty through reference data. This odometry system is designed to compensate for localization failures. In particular, there are some key contributions to this work. The first contribution is developing a generic framework that fuses measurement of IMU and camera in a tightly coupled manner, which includes the estimation of IMU's biases along with loosely-coupled fusion with GNSS and Lidar. The second contribution is to design a learning-based odometry system that is robust to sensor uncertainties.

This chapter is organized as follows: In Section 3.2, a model-based vehicle odometry system is developed based on an Error-State Unscented Kalman Filter (ESUKF) that fuses a monocular camera, a GNSS, and a Lidar with IMU. In this section, the proposed model-based odometry estimator is validated by experimental and realistic simulation data. In Section 3.3, a machine-learning odometry estimator is developed through kernel-based non-parametric regression models using effective input feature selection. Finally, the proposed machine learning-based odometry estimator is evaluated based on experimental tests along with discussion in Section 3.3.

3.2 Model-Based Multi-Modal Vehicle Odometry

In this section, the overall structure of sensor fusion is discussed based on the Unscented Kalman Filter (UKF) approach. The reason for utilizing unscented transform is that having highly nonlinear camera measurement functions makes the Extended Kalman Filter(EKF) inaccurate and inconvenient.

3.2.1 IMU Measurement Model

The 6-axis IMU contains an accelerometer and a gyroscope that measures 3D accelerations and 3D angular velocities, respectively. The IMU measurements are modeled by the following equations:

$$\begin{aligned} \mathbf{a}_m &= \left({}^N\mathbf{R}_I\right)^\top \left({}^N\mathbf{a} +^N \mathbf{g}\right) + \mathbf{b}_a + \mathbf{n}_a \\ \omega_m &=^I\omega + \mathbf{b}_g + \mathbf{n}_g \ , \end{aligned} \tag{3.1}$$

where $\mathbf{a}_m \in \mathbb{R}^3$ is the measured acceleration of the vehicle that is expressed in the vehicle frame $\{V\}$, ${}^N\mathbf{a} \in \mathbb{R}^3$ is the true acceleration expressed in the navigation frame $\{N\}$, ${}^N\mathbf{g} = [0,\ 0,\ 9.81]^\top$ [m /s^2] is the gravitational acceleration vector, $\mathbf{n}_a \in \mathbb{R}^3$ is the accelerometer's zero-mean Gaussian noise, $\omega_m \in \mathbb{R}^3$ is the measured angular velocity, ${}^I\omega$ is the true angular velocity, and $\mathbf{n}_g \in \mathbb{R}^3$ is the gyroscope's zero-mean Gaussian noise. Note that the effect of the Earth's rotation is neglected on the measured acceleration by simply assuming that $\{N\}$ is stationary.

The accelerometer biases $\mathbf{b}_a \in \mathbb{R}^3$ and the gyroscope biases $\mathbf{b}_g \in \mathbb{R}^3$ are modeled as Gaussian random walk processes driven by zero-mean Gaussian noise $\mathbf{n}_{\text{ba}}$ and $\mathbf{n}_{\text{bg}}$, respectively:

$$\begin{aligned} \dot{\mathbf{b}}_\mathbf{a} &=\mathbf{n}_{\text{ba}} \\ \dot{\mathbf{b}}_\mathbf{g} &=\mathbf{n}_{\text{bg}} \ . \end{aligned} \tag{3.2}$$

Accordingly, the IMU kinematics can be written as follows:

$$\begin{aligned} {}^\mathbf{N}\dot{\mathbf{p}}_\mathbf{I} &=^\mathbf{N} \mathbf{v}_\mathbf{I} \\ {}^\mathbf{N}\dot{\mathbf{q}}_\mathbf{I} &= \frac{\mathbf{1}}{\mathbf{2}}\left({}^\mathbf{N}\mathbf{q}_\mathbf{I}\right) \otimes \left[\mathbf{0} \quad \omega_\mathbf{m}^\top - \mathbf{b}_\mathbf{g}^\top - \mathbf{n}_\mathbf{g}^\top\right]^\top \\ {}^N\dot{v} &= {}^N\mathbf{R}_I\left(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a\right) -^N \mathbf{g} \\ \dot{\mathbf{b}}_\mathbf{a} &= \mathbf{n}_{\text{ba}} \\ \dot{\mathbf{b}}_\mathbf{g} &= \mathbf{n}_{\text{bg}} \ , \end{aligned} \tag{3.3}$$

where ${}^N\mathbf{v}_I$ is the velocity of $\{I\}$ with respect to $\{N\}$ and expressed in $\{N\}$, ${}^\mathbf{N}q_\mathbf{I}$ is the unit quaternion corresponding to the rotation from $\{I\}$ to $\{N\}$, and $\otimes$ denotes the operation of quaternion multiplication.

The state vector that fully describes the entire state of the IMU model at any time is comprised of the position, orientation, velocity, and biases of the IMU, which are as follows:

$$\mathbf{x}_{\mathrm{IMU}} = \begin{bmatrix} {}^{N}\mathbf{p}^{\top} & {}^{N}\mathbf{q}^{\top} & {}^{N}\mathbf{v}^{\top} & \mathbf{b}_a^{\top} & \mathbf{b}_g^{\top} \end{bmatrix}^{\top} . \tag{3.4}$$

3.2.2 Decomposition of (True) States into the Nominal States and the Error States

It is beneficial to decompose the state (called also as a true state here) into the nominal state and error-state vector for multiple reasons; first, it reduces the dimension of the state vector by parameterizing the rotation using three Euler angles instead of a quaternion (which has 4 components) and second, it enables the linearization of the system. The true, nominal, and error state vectors of IMU are:

$$\begin{aligned}
\textbf{\textit{True state:}}\ \mathbf{x}_{\mathrm{IMU}} &= \begin{bmatrix} {}^{\mathbf{N}}\mathbf{p}^{\top} & {}^{\mathbf{N}}\mathbf{q}^{\top} & {}^{\mathbf{N}}\mathbf{v}^{\top} & \mathbf{b}_{\mathbf{a}}^{\top} & \mathbf{b}_{\mathbf{g}}^{\top} \end{bmatrix}^{\top} \in \mathbb{R}^{\mathbf{16}} \\
\textbf{\textit{Nominal state:}}\ \widehat{\mathbf{x}}_{\mathrm{IMU}} &= \begin{bmatrix} {}^{\mathbf{N}}\widehat{\mathbf{p}}^{\top} & {}^{\mathbf{N}}\widehat{\mathbf{q}}^{\top} & {}^{\mathbf{N}}\widehat{\mathbf{v}}^{\top} & \widehat{\mathbf{b}}_{\mathbf{a}}^{\top} & \widehat{\mathbf{b}}_{\mathbf{g}}^{\top} \end{bmatrix}^{\top} \in \mathbb{R}^{\mathbf{16}} \\
\textbf{\textit{Error state:}}\ \widetilde{\mathbf{x}}_{\mathrm{IMU}} &= \begin{bmatrix} {}^{\mathbf{N}}\widetilde{\mathbf{p}}^{\top} & {}^{\mathbf{N}}\delta\theta & {}^{\mathbf{N}}\widetilde{\mathbf{v}}^{\top} & \widetilde{\mathbf{b}}_{\mathbf{a}}^{\top} & \widetilde{\mathbf{b}}_{\mathbf{g}}^{\top} \end{bmatrix}^{\top} \in \mathbb{R}^{\mathbf{15}} .
\end{aligned} \tag{3.5}$$

The true state is considered as the composition of the nominal state and error state whose relationships are as follows:

$$\begin{aligned}
{}^{\mathbf{N}}\mathbf{p} &= {}^{\mathbf{N}}\widehat{\mathbf{p}} + {}^{\mathbf{N}}\widetilde{\mathbf{p}} \\
{}^{\mathbf{N}}\mathbf{q} &= {}^{\mathbf{N}}\widehat{\mathbf{q}} \otimes \begin{bmatrix} 1 & \frac{1}{2}\delta\theta^{\top} \end{bmatrix}^{\top} \\
{}^{\mathbf{N}}\mathbf{v} &= {}^{\mathbf{N}}\widehat{\mathbf{v}} + {}^{\mathbf{N}}\widetilde{\mathbf{v}} \\
\mathbf{b}_a &= \widehat{\mathbf{b}}_a + \widetilde{\mathbf{b}}_a \\
\mathbf{b}_g &= \widehat{\mathbf{b}}_g + \widetilde{\mathbf{b}}_g .
\end{aligned} \tag{3.6}$$

The nominal state kinematics is considered the same as the true state kinematics in (3.6) when assuming no uncertainty is present (setting all the noises to zero). Therefore,

$$
\begin{aligned}
{}^{\mathbf{N}}\dot{\widehat{\mathbf{p}}}_{\mathbf{I}} &= {}^{\mathbf{N}}\widehat{\mathbf{v}} \\
{}^{\mathbf{N}}\dot{\widehat{\mathbf{q}}} &= \frac{\mathbf{1}}{\mathbf{2}}{}^{\mathbf{N}}\widehat{\mathbf{q}} \otimes \begin{bmatrix}\mathbf{0} & \left({}^{\mathbf{I}}\widehat{\omega}_{\mathbf{m}}{}^{\top} - \widehat{\mathbf{b}}_{\mathbf{g}}^{\top}\right)\end{bmatrix}^{\top} \\
{}^{\mathbf{N}}\dot{\widehat{\mathbf{v}}} &= R\left({}^{\mathbf{N}}\widehat{\mathbf{q}}\right)(\mathbf{a}_{\mathbf{m}} - \widehat{\mathbf{b}}_{\mathbf{a}}) - {}^{\mathbf{N}}\mathbf{g} \\
\dot{\widehat{\mathbf{b}}}_{\mathbf{a}} &= \mathbf{0}_{\mathbf{3}\times\mathbf{1}} \\
\dot{\widehat{\mathbf{b}}}_{\mathbf{g}} &= \mathbf{0}_{\mathbf{3}\times\mathbf{1}} \ .
\end{aligned} \tag{3.7}
$$

Based on the nominal state kinematics in the composition of states in (3.6), the error state kinematics can be derived as follows (see [75] for the full derivation):

$$
\begin{aligned}
{}^{\mathbf{N}}\dot{\widetilde{\mathbf{p}}} &= {}^{\mathbf{N}}\widetilde{\mathbf{v}} \\
{}^{\mathbf{N}}\delta\dot{\theta} &= -\left\lfloor {}^{\mathbf{I}}\widehat{\omega}_{\mathbf{m}}{}^{\top} - \widehat{\mathbf{b}}_{\mathbf{g}}^{\top} \right\rfloor_{\times} {}^{\mathbf{N}} - \widetilde{\mathbf{b}}_{\mathbf{g}} - \mathbf{n}_{\mathbf{g}} \\
{}^{\mathbf{N}}\dot{\widetilde{\mathbf{v}}} &= -R\left({}^{\mathbf{N}}\widehat{\mathbf{q}}\right)\left(\left\lfloor \mathbf{a}_{\mathbf{m}} - \widehat{\mathbf{b}}_{\mathbf{a}} \right\rfloor_{\times} {}^{\mathbf{N}}\delta\theta + \widetilde{\mathbf{b}}_{\mathbf{a}} + \mathbf{n}_{\mathbf{a}}\right) \\
\dot{\widetilde{\mathbf{b}}}_{\mathbf{a}} &= \mathbf{n}_{\mathrm{ba}} \\
\dot{\widetilde{\mathbf{b}}}_{\mathbf{g}} &= \mathbf{n}_{\mathrm{bg}} \ ,
\end{aligned} \tag{3.8}
$$

where $\lfloor\cdot\rfloor_{\times}$ is the skew-symmetric matrix defined as:

$$
\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\times} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \ . \tag{3.9}
$$

Now, the error state kinematics of IMU is fully derived and ready to be used in the ESUKF, which is discussed in the following sections.

3.2.3 Camera Measurement Model

The camera provides a sequence of images at some frame rate. The camera might observe common visual features across multiple frames when the vehicle navigates through the environment. Each feature observed in multiple camera frames imposes a constraint on the camera's pose at the capture time of the frames. Consider $\{\mathbf{m}_{k-2}, \mathbf{m}_{k-1}, \mathbf{m}_k\}_i$ are three observations of a feature i in three consecutive camera frames $\{C_{k-2}, C_{k-1},\ C_k\}$. The camera measurement model contains a constraint for the feature i $\in \{1, 2, \ldots, n\}$ that is observed in the last three poses of the camera based on the trifocal tensor, and a constraint for any feature that is observed in two consecutive poses (i.e. $\{C_{k-2}, C_{k-1}\}$ or $\{C_{k-1}, C_k\}$) based on the epipolar constraint [58]:

$$\begin{aligned}
\mathbf{z_{c,i}}{=}\mathbf{h_c}\left(\mathbf{X_k}, \{\mathbf{m_{k-2}},\mathbf{m_{k-1}},\mathbf{m_k}\}_\mathbf{i}\right) &= \begin{bmatrix} \widetilde{\mathbf{m}}_\mathbf{k-1}^\top \mathbf{R}_\mathbf{12}^\top \lfloor \mathbf{t_{12}} \rfloor_\times \widetilde{\mathbf{m}}_\mathbf{k-2}^\top \\ \widetilde{\mathbf{m}}_\mathbf{k}^\top \mathbf{R}_\mathbf{23}^\top \lfloor \mathbf{t_{23}} \rfloor_\times \widetilde{\mathbf{m}}_\mathbf{k-1}^\top \\ \mathbf{K}\left(\sum_\mathbf{i} \widetilde{\mathbf{m}}_\mathbf{1i}^\top \mathbf{T}_\mathbf{i}^\top\right)\mathbf{l_2} \end{bmatrix} + \mathbf{n_c}, \qquad (3.10)\\
\mathbf{i} &= 1, 2, \ldots, n,\\
n_c &\sim N(0, \Sigma_c),
\end{aligned}$$

where

$$\begin{aligned}
\widetilde{\mathbf{m}}_{k-2} &= \mathbf{K^{-1}}\mathbf{m}_{k-2} \qquad (3.11)\\
\widetilde{\mathbf{m}}_{k-1} &= \mathbf{K^{-1}}\mathbf{m}_{k-1}\\
\widetilde{\mathbf{m}}_{k} &= \mathbf{K^{-1}}\mathbf{m}_{k}\\
\mathbf{R}_{12} &= \left({}^G\mathbf{R}_{C_{k-2}}\right)^\top {}^G\mathbf{R}_{C_{k-1}}\\
\mathbf{R}_{23} &= \left({}^G\mathbf{R}_{C_{k-1}}\right)^\top {}^G\mathbf{R}_{C_{k}}\\
\mathbf{t}_{12} &= \left({}^\mathbf{N}\mathbf{R}_{C_{k-2}}\right)^\top \left({}^\mathbf{N}\mathbf{p}_{C_{k-1}} - {}^\mathbf{N}\mathbf{p}_{C_{k-2}}\right)\\
\mathbf{t}_{23} &= \left({}^\mathbf{N}\mathbf{R}_{C_{k-1}}\right)^\top \left({}^\mathbf{N}\mathbf{p}_{C_{k}} - {}^\mathbf{N}\mathbf{p}_{C_{k-1}}\right)\\
{}^\mathbf{N}\mathbf{R}_{C_{k-2}} &= R\left({}^\mathbf{N}\mathbf{q}_{I_{k-2}}\right) {}^I\mathbf{R}_C\\
{}^\mathbf{N}\mathbf{p}_{C_{k-2}} &= {}^\mathbf{N}\mathbf{p}_{I_{k-2}} + R\left({}^\mathbf{N}\mathbf{q}_{I_{k-2}}\right) {}^I\mathbf{p}_C,\\
{}^\mathbf{N}\mathbf{R}_{C_{k-1}} &= R\left({}^\mathbf{N}\mathbf{q}_{I_{k-1}}\right) {}^I\mathbf{R}_C\\
{}^\mathbf{N}\mathbf{p}_{C_{k-1}} &= {}^\mathbf{N}\mathbf{p}_{I_{k-1}} + R\left({}^\mathbf{N}\mathbf{q}_{I_{k-1}}\right) {}^I\mathbf{p}_C,\\
{}^\mathbf{N}\mathbf{R}_{C_{k}} &= R\left({}^\mathbf{N}\mathbf{q}_{I_{k}}\right) {}^I\mathbf{R}_C\\
{}^\mathbf{N}\mathbf{p}_{C_{k}} &= {}^\mathbf{N}\mathbf{p}_{I_{k}} + R\left({}^\mathbf{N}\mathbf{q}_{I_{k}}\right) {}^I\mathbf{p}_C\\
\mathbf{l}_2 &= \left(l_{e2} - l_{e1},\ -\widetilde{m}_{2u} l_{e2} + \widetilde{m}_{2v} l_{e1}\right)^\top,\\
\mathbf{R}_{12}^\top \lfloor \mathbf{t}_{12} \rfloor_\times \widetilde{\mathbf{m}}_1 &= \left(l_{e1}, l_{e2}, l_{e3}\right)^\top\\
\widetilde{\mathbf{m}}_2 &= \left(\widetilde{m}_{2u}, \widetilde{m}_{2v}, 1\right)^\top\ .
\end{aligned}$$

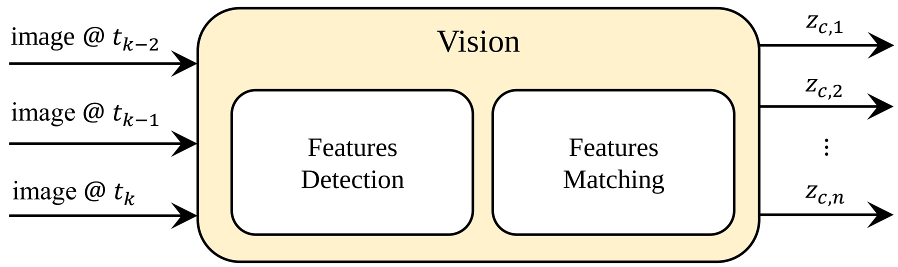


Figure 3.2: Camera measurement block, which receives three consecutive images, detects and matches features, and outputs the measurement vector corresponding to each observed feature.

Figure 3.2 shows the block diagram of the camera measurement model; first, it receives three consecutive image frames; then after detecting features and matching n features across the three consecutive frames, it outputs n measurement vectors corresponding to each matched visual feature.

Note that the nonlinearity inside the camera measurement model in (3.10) is considered by using sigma points in the UKF.

The main benefit of using epipolar constraint and trifocal tensor here is that they are just functions of 3D camera observations and camera pose while they are not a function of the feature position in 3D space (the mapping). Therefore, there is no need to have an extra step of structure estimation using the least square problem (same as [51]) and no need to include the position of numerous features inside the state vector. This is desirable since the state vector remains short and the computation complexity to estimate the structure is avoided. This is aligned with the main objective which is to estimate the vehicle's state rather than estimating the mapping.

On the other hand, because the environment's appearance is not robust to illumination and seasonal changes, visual features are not stored and used for localization objectives. However, during a short time interval of three consecutive frames, the environment's appearance does not change significantly; therefore, visual features used for odometry remain robust.

3.2.4 GNSS Measurement Model

In this project, a low-grade GNSS sensor is considered used in the sensory suite that measures its global position. The global coordinate system is the Universal Transverse Mercator (UTM) coordinate system that expresses the location as northing, easting, and height:

$$\mathbf{z}_G = \begin{bmatrix} \text{northing} \\ \text{easting} \\ \text{height} \end{bmatrix} . \tag{3.12}$$

The GNSS measures the position in $\{G\}$ frame which is assumed to be contaminated with a zero-mean Gaussian noise. Accordingly, the GNSS measurement model is as follows:

$$\begin{aligned} \mathbf{z}_G = h_G\left(\mathbf{X}_k\right) =^G \mathbf{p}_N + \left({}^G\mathbf{R}_N\right)^N \mathbf{p} + \left({}^G\mathbf{R}_N\right)\left({}^N\mathbf{R}\right)^I \mathbf{p}_{\text{GNSS}} + \mathbf{n}_{\text{GNSS}}, \\ \mathbf{n}_{\text{GNSS}} \sim N(0, \mathbf{\Sigma}_G) \end{aligned} \tag{3.13}$$

where ${}^G\mathbf{p}_N$ and ${}^G\mathbf{R}_N$ are the GNSS alignment, which includes the position and orientation of the navigation frame with respect to the GNSS frame, ${}^I\mathbf{p}_{\text{GNSS}}$ is the lever arm of the GNSS sensor to the IMU sensor on the vehicle (which is a calibration parameter), and $\mathbf{\Sigma}_G = diag(\sigma_{\text{GNSS},x}, \sigma_{\text{GNSS},y}, \sigma_{\text{GNSS},z})$.

3.2.5 GNSS Alignment

The measurement model in (3.13) relates the GNSS measurements to multiple quantities, including GNSS alignment ${}^G\mathbf{p}_N$ and ${}^G\mathbf{R}_N$. This is because the reference of the GNSS positioning is a global frame $\{G\}$, whereas the IMU states in (3.4) are expressed in a local stationary frame $\{N\}$. Therefore, to use the GNSS measurement model in (3.13), there is a need to find the GNSS alignment which is the transformation that aligns the local coordinate system $\{N\}$ to the global coordinate system $\{G\}$ (Figure 3.3).

As both $\{N\}$ and $\{G\}$ frames are already aligned with gravity, there is a need to estimate a 4-DOF transformation between these frames, including rotation around the gravity vector (z-axis) and a 3D translation. Given a set of nominal estimations from the state estimation system and the GNSS measurements, one can solve a nonlinear least-square problem to find the alignment:

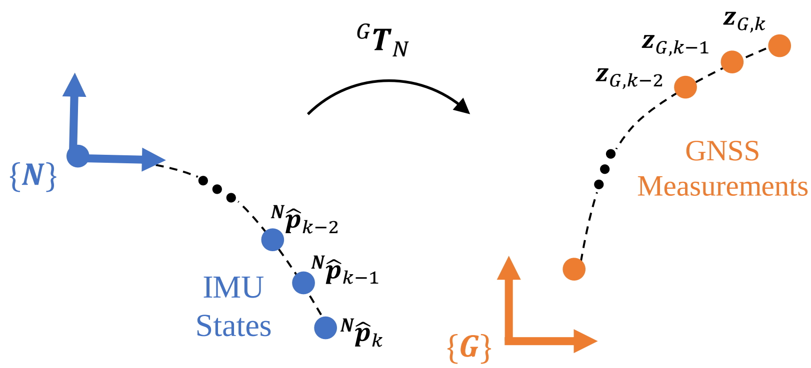


Figure 3.3: The GNSS measurements are expressed in a global coordinate system whereas the IMU states are expressed in a local navigation frame. A transformation can align the two coordinate systems together.

$$ {}^{G}\mathbf{T}_N{}^{\star} = \underset{{}^{G}\mathbf{T}_N}{\operatorname{argmin}} \sum_j \left\| z_{G,j} - f\left({}^{G}\mathbf{T}_N, {}^{N}\mathbf{p}_j , {}^{N}\mathbf{R}_j \right) \right\|_{\Omega}. \tag{3.14} $$

Note that the GNSS alignment requires the system to wait for some time to get a window of GNSS measurements with enough displacement to estimate this alignment. After getting this alignment, the system can fuse the measurements from GNSS.

3.2.6 Lidar Measurement Model

The Normal Distribution Transformation (NDT) map-matching technique [6] is used to match the Lidar points in the current frame to the Lidar points in the last frame. The NDT algorithm is a registration algorithm that uses standard optimization techniques applied to statistical models of 3D points to determine the most probable registration between two point clouds.

Moreover, some point cloud sanitization techniques are performed. The points on the ground are removed based on the thresholding approach. To improve the efficiency and accuracy of the registration algorithm, the point clouds are down-sampled using random sampling with a sample ratio, which reduces the total number of points. Finally, through annular region selection, sparse distance points, as well as points on the vehicle itself, are removed from the point cloud of Lidar.

The Lidar measurement model is the constraint over two consecutive Lidar point clouds, which is imposed by the output of the Lidar point cloud registration:

$$\mathbf{z}_L = \mathbf{h}_L\left(\mathbf{X}_k\right) + \mathbf{n}_L = \begin{bmatrix} \left({}^N\mathbf{R}_{k-1}\right)^{-1}\left({}^N\mathbf{p}_k - {}^N\mathbf{p}_{k-1}\right) \\ \theta\{\left({}^N\mathbf{q}_{k-1}\right)^{-1} \bigotimes \left({}^N\mathbf{q}_k\right)\} \end{bmatrix} + \mathbf{n}_L,\ \mathbf{n}_L \sim N(0, \mathbf{\Sigma}_L)\ , \tag{3.15}$$

where $\theta\{\cdot\}$ is the Euler angles vector containing roll, pitch, and yaw, given the rotation in quaternion, and $\mathbf{n}_L$ is assumed to be a zero-mean Gaussian noise. Figure 3.4 illustrates the inputs and outputs of the Lidar measurement model and the intermediate steps.

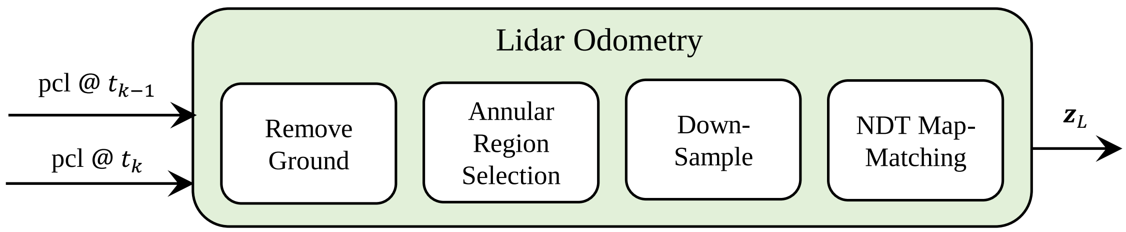


Figure 3.4: Lidar Measurement Block. It receives point clouds in two consecutive frames and does registration in addition to some processing

3.2.7 Estimator Structure

Filter State Vector

Figure 3.5 illustrates the factor-graph representation of the observer system. Accordingly, a camera measurement (detected feature) imposes a constraint over three consecutive states, Lidar and IMU measurements impose a constraint over two consecutive states and GNSS imposes a constraint over each state. Therefore, the filter state vector should contain the last two poses of the vehicle in addition to the IMU states at the current time. The filter nominal state at time step k is:

$$\widehat{\mathbf{x}}_k = \begin{bmatrix} \widehat{\mathbf{x}}_{IMU,k}^{\top} & {}^N\widehat{\mathbf{p}}_{k-1}{}^{\top} & {}^N\widehat{\mathbf{q}}_{k-1}{}^{\top} & {}^N\widehat{\mathbf{p}}_{k-2}{}^{\top} & {}^N\widehat{\mathbf{q}}_{k-2}{}^{\top} \end{bmatrix}^{\top} \in \mathbb{R}^{30}\ . \tag{3.16}$$

Accordingly, the filter error state is:

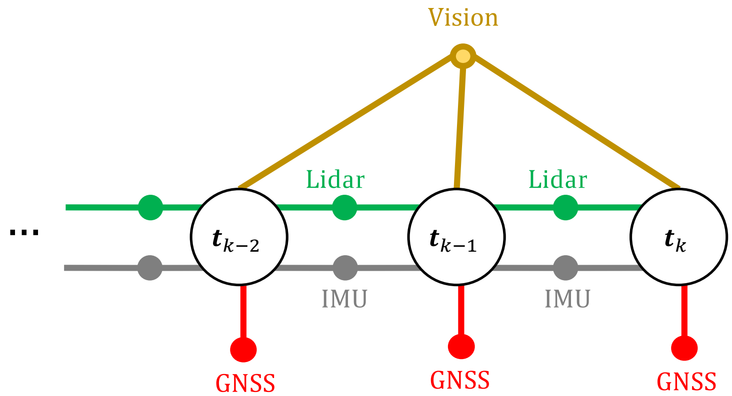


Figure 3.5: Factor-graph Representation of the multi-modal odometry estimation problem. The state vector of the odometry at the time t_k is determined based on the edge of visual measurements to the last two states (denoted in yellow), the edge of Lidar measurements and the last state (denoted in green), the edge of IMU measurements and last state (denoted in gray), and the edge of GNSS measurement at the current time (denoted in red).

$$\widetilde{\mathbf{x}}_k = \left[\widetilde{\mathbf{x}}_{IMU,k}^\top \quad {}^N\widetilde{\mathbf{p}}_{k-1}{}^\top \quad {}^N\delta\theta_{k-1}{}^\top \quad {}^N\widetilde{\mathbf{p}}_{k-2}{}^\top \quad {}^N\delta\theta_{k-2}{}^\top\right]^\top \in \mathbb{R}^{27}\ . \tag{3.17}$$

Filter Prediction

Given that the last two poses are stationary, the filter propagation model of the nominal state is as follows:

$$\begin{aligned}
{}^{\mathbf{N}}\dot{\widehat{\mathbf{p}}}_{\mathbf{I}} &= {}^{\mathbf{N}}\widehat{\mathbf{v}} \\
{}^{\mathbf{N}}\dot{\widehat{\mathbf{q}}} &= \frac{\mathbf{1}}{\mathbf{2}}{}^{\mathbf{N}}\widehat{\mathbf{q}} \otimes \begin{bmatrix}\mathbf{0} & \left({}^{\mathbf{I}}\widehat{\omega}_{\mathbf{m}}{}^{\top} - \widehat{\mathbf{b}}_{\mathbf{g}}^{\top}\right)\end{bmatrix}^{\top} \\
{}^{\mathbf{N}}\dot{\widehat{\mathbf{v}}} &= R\left({}^{\mathbf{N}}\widehat{\mathbf{q}}\right)(\mathbf{a}_{\mathbf{m}} - \widehat{\mathbf{b}}_{\mathbf{a}}) - {}^{\mathbf{N}}\mathbf{g} \\
\dot{\widehat{\mathbf{b}}}_{\mathbf{a}} &= \mathbf{0}_{\mathbf{3}\times\mathbf{1}} \\
\dot{\widehat{\mathbf{b}}}_{\mathbf{g}} &= \mathbf{0}_{\mathbf{3}\times\mathbf{1}} \\
{}^{N}\dot{\widehat{\mathbf{p}}}_{k-1} &= 0 \\
{}^{N}\dot{\widehat{\mathbf{q}}}_{k-1} &= 0 \\
{}^{N}\dot{\widehat{\mathbf{p}}}_{k-2} &= 0 \\
{}^{N}\dot{\widehat{\mathbf{q}}}_{k-2} &= 0\ .
\end{aligned} \tag{3.18}$$

Additionally, the filter propagation model of the error state is as follows:

$$\begin{aligned}
{}^{\mathbf{N}}\dot{\widetilde{\mathbf{p}}} &= {}^{\mathbf{N}}\widetilde{\mathbf{v}} \\
{}^{\mathbf{N}}\delta\dot{\theta} &= -\left\lfloor {}^{\mathbf{I}}\widehat{\omega}_{\mathbf{m}}{}^{\top} - \widehat{\mathbf{b}}_{\mathbf{g}}^{\top}\right\rfloor_{\times} {}^{\mathbf{N}} - \widetilde{\mathbf{b}}_{\mathbf{g}} - \mathbf{n}_{\mathbf{g}} \\
{}^{\mathbf{N}}\dot{\widetilde{\mathbf{v}}} &= -R\left({}^{\mathbf{N}}\widehat{\mathbf{q}}\right)\left(\left\lfloor \mathbf{a}_{\mathbf{m}} - \widehat{\mathbf{b}}_{\mathbf{a}}\right\rfloor_{\times} {}^{\mathbf{N}}\delta\theta + \widetilde{\mathbf{b}}_{\mathbf{a}} + \mathbf{n}_{\mathbf{a}}\right) \\
\dot{\widetilde{\mathbf{b}}}_{\mathbf{a}} &= \mathbf{n}_{\mathrm{ba}} \\
\dot{\widetilde{\mathbf{b}}}_{\mathbf{g}} &= \mathbf{n}_{\mathrm{bg}} \\
{}^{N}\dot{\widetilde{\mathbf{p}}}_{k-1} &= 0 \\
{}^{N}\delta\dot{\theta}_{k-1} &= 0 \\
{}^{N}\dot{\widetilde{\mathbf{p}}}_{k-2} &= 0 \\
{}^{N}\delta\dot{\theta}_{k-2} &= 0\ .
\end{aligned} \tag{3.19}$$

Accordingly, the continuous error-state kinematic model is:

$$\dot{\widetilde{\mathbf{x}}} = \mathbf{F}_c\widetilde{\mathbf{x}} + \mathbf{G}_c\mathbf{n}_{\text{IMU}},\ \mathbf{n}_{\text{IMU}}= \begin{bmatrix}\mathbf{n}_a^\top & \mathbf{n}_g^\top & \mathbf{n}_{\text{ba}}^\top & \mathbf{n}_{\text{bg}}^\top\end{bmatrix}\ , \tag{3.20}$$

where

$$\mathbf{F}_c = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & -\left\lfloor {}^I\widehat{\omega}_m{}^\top - \widehat{\mathbf{b}}_g^\top \right\rfloor_\times & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & -\mathbf{R}\left({}^N\widehat{\mathbf{q}}\right)\left\lfloor \mathbf{a}_m - \widehat{\mathbf{b}}_a \right\rfloor_\times & \mathbf{0}_{3\times3} & -\mathbf{R}\left({}^N\widehat{q}\right) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times12} \end{bmatrix}, \tag{3.21}$$

$$\mathbf{G}_c = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\mathbf{R}\left({}^N\widehat{\mathbf{q}}\right) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \\ \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} \\ \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} \end{bmatrix}.$$

The zero-order-hold discretized error-state kinematics is:

$$\widetilde{\mathbf{x}}_k = \mathbf{F}_d\widetilde{\mathbf{x}}_{k-1} + \mathbf{G}_d n_{\text{IMU}}\ , \tag{3.22}$$

where

$$\mathbf{F_d} = \exp\left(\mathbf{F}_c \mathrm{t}\right) = \mathbf{I}_{27\times27} + \mathbf{F}_c \Delta t + \frac{1}{2!}\mathbf{F}_c^2 \Delta t^2 + \ldots = \begin{bmatrix} \mathbf{I}_3 & \mathbf{F}_1 & \mathbf{I}_3 & -\frac{1}{2}\mathbf{R}\left({}^N\widehat{\mathbf{q}}\right)\Delta t^2 & \mathbf{F}_4 & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & \mathbf{F}_2 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{F}_5 & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & \mathbf{F}_3 & \mathbf{I}_3 & -\mathbf{R}\left({}^N\widehat{\mathbf{q}}\right)\mathrm{t} & \mathbf{F}_6 & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times12} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times12} \\ \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{0}_{12\times3} & \mathbf{I}_{12} \end{bmatrix} \tag{3.23}$$

$$\begin{aligned}
\mathbf{F}_1 &= -\mathbf{R}\left({}^N\widehat{\mathbf{q}}\right)\left\lfloor \mathbf{a}_m - \widehat{\mathbf{b}}_a \right\rfloor_\times \left(-\frac{1}{2}\Delta t^2 \mathbf{I}_3 + \frac{1}{6}\left\lfloor {}^I\widehat{\omega}_m{}^\top - \widehat{\mathbf{b}}_g^\top \right\rfloor_\times \Delta t^3 - \frac{1}{24}\left\lfloor {}^I\widehat{\omega}_m{}^\top - \widehat{\mathbf{b}}_g^\top \right\rfloor_\times^2 \Delta t^4\right) \\
\mathbf{F}_2 &= \mathbf{I}_3 - \left\lfloor {}^I\widehat{\omega}_m{}^\top - \widehat{\mathbf{b}}_g^\top \right\rfloor_\times \Delta t + \frac{1}{2}\left\lfloor {}^I\widehat{\omega}_m{}^\top - \widehat{\mathbf{b}}_g^\top \right\rfloor_\times^2 \Delta t^2 \\
\mathbf{F}_3 &= \mathbf{R}\left({}^N\widehat{\mathbf{q}}\right)\left\lfloor \mathbf{a}_m - \widehat{\mathbf{b}}_a \right\rfloor_\times \left(-\Delta t \mathbf{I}_3 + \frac{1}{2}\left\lfloor {}^I\widehat{\omega}_m{}^\top - \widehat{\mathbf{b}}_g^\top \right\rfloor_\times \Delta t^2 - \frac{1}{16}\left\lfloor {}^I\widehat{\omega}_m{}^\top - \widehat{\mathbf{b}}_g^\top \right\rfloor_\times^2 \Delta t^3\right) \\
\mathbf{F}_4 &= \mathbf{R}\left({}^N\widehat{\mathbf{q}}\right)\left\lfloor \mathbf{a}_m - \widehat{\mathbf{b}}_a \right\rfloor_\times \left(\frac{1}{6}\Delta t^3 \mathbf{I}_3 - \frac{1}{24}\left\lfloor {}^I\widehat{\omega}_m{}^\top - \widehat{\mathbf{b}}_g^\top \right\rfloor_\times \Delta t^4 - \frac{1}{120}\left\lfloor {}^I\widehat{\omega}_m{}^\top - \widehat{\mathbf{b}}_g^\top \right\rfloor_\times^2 \Delta t^5\right) \\
\mathbf{F}_5 &= \Delta t \mathbf{I}_3 + \frac{1}{2}\left\lfloor {}^I\widehat{\omega}_m{}^\top - \widehat{\mathbf{b}}_g^\top \right\rfloor_\times \Delta t^2 - \frac{1}{6}\left\lfloor {}^I\widehat{\omega}_m{}^\top - \widehat{\mathbf{b}}_g^\top \right\rfloor_\times^2 \Delta t^3 \\
\mathbf{F}_6 &= -\mathbf{F}_1 \ .
\end{aligned}$$

The covariance of the noise is

$$\mathbf{Q}_c = \mathrm{var}\left(\mathbf{n}_{\mathrm{IMU}}\right) = \mathrm{diag}\left(\sigma_g \mathbf{I}_3, \sigma_a \mathbf{I}_3, \sigma_{\mathrm{ba}} \mathbf{I}_3, \sigma_{\mathrm{bg}} \mathbf{I}_3\right) \ , \tag{3.24}$$

where σ_g, σ_a, σ_{ba}, σ_{bg} are scalars. The propagated covariance matrix is:

$$\begin{aligned}
\mathbf{Q}_d &= \int_{\mathbf{\Delta t}} \mathbf{F}_d(\tau)\mathbf{G_c}\mathbf{Q}_c\mathbf{G}_c^\top \mathbf{F}_d(\tau)^\top \mathrm{d} \qquad (3.25) \\
\mathbf{P}_{k|k-1} &= \mathbf{F}_d \mathbf{P}_{k-1|k-1}\mathbf{F}_d^\top + \mathbf{Q}_d \ .
\end{aligned}$$

Marginalization of State Vector

When the new set of measurements arrives, the last vehicle pose is discarded, and the new pose is added to the state vector to keep just the last three vehicle poses. The nominal state, error state, and covariance are marginalized as follows:

$$
\begin{aligned}
\widehat{\mathbf{x}}_k &= \mathbf{T}_n \widehat{\mathbf{x}}_k \\
\widetilde{\mathbf{x}}_k &= \mathbf{T}_n \widetilde{\mathbf{x}}_k \\
\mathbf{P}_{k|k} &= \mathbf{T}_e \mathbf{P}_{k|k} \mathbf{T}_e^\top \ ,
\end{aligned} \tag{3.26}
$$

where

$$
\mathbf{T}_n = \begin{bmatrix} \mathbf{I}_{7\times7} & \mathbf{0}_{7\times9} & \mathbf{0}_{7\times7} & \mathbf{0}_{7\times7} \\ \mathbf{0}_{9\times7} & \mathbf{I}_{9\times9} & \mathbf{0}_{9\times7} & \mathbf{0}_{9\times7} \\ \mathbf{0}_{7\times7} & \mathbf{0}_{7\times9} & \mathbf{0}_{7\times7} & \mathbf{I}_{7\times7} \\ \mathbf{I}_{7\times7} & \mathbf{0}_{7\times9} & \mathbf{0}_{7\times7} & \mathbf{0}_{7\times7} \end{bmatrix}, \ \mathbf{T}_e = \begin{bmatrix} \mathbf{I}_{6\times6} & \mathbf{0}_{6\times9} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{9\times6} & \mathbf{I}_{9\times9} & \mathbf{0}_{9\times6} & \mathbf{0}_{9\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times9} & \mathbf{0}_{6\times6} & \mathbf{I}_{6\times6} \\ \mathbf{I}_{6\times6} & \mathbf{0}_{6\times9} & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \end{bmatrix} \ . \tag{3.27}
$$

Observability

According to [76], for a visual-inertial observer with a known gravity vector and 6-axis IMU, the necessary conditions for full observability of a visual-inertial observer are as follows:

1. To estimate the observable modes the vehicle cannot move at a constant linear speed
2. To estimate the observable modes the camera must perform at least three observations

The second condition is simply met after receiving three camera frames. However, the first condition requires the motion of the vehicle to be sufficiently rich to capture accelerations in the IMU measurements with enough excitations. In the developed augmented state estimation, the presence of other sensor measurements improves the observability of the system and compensates for such conditions.

The overall structure of the state estimation system is shown in Figure 3.6 and the main steps are summarized in Algorithm 1.

3.2.8 Results and Discussion

The performance of the developed augmented odometry system is evaluated in real experiments. Accordingly. the KITTI dataset is used to evaluate the state estimation performance that contains synchronized measurements of a vehicle equipped with different

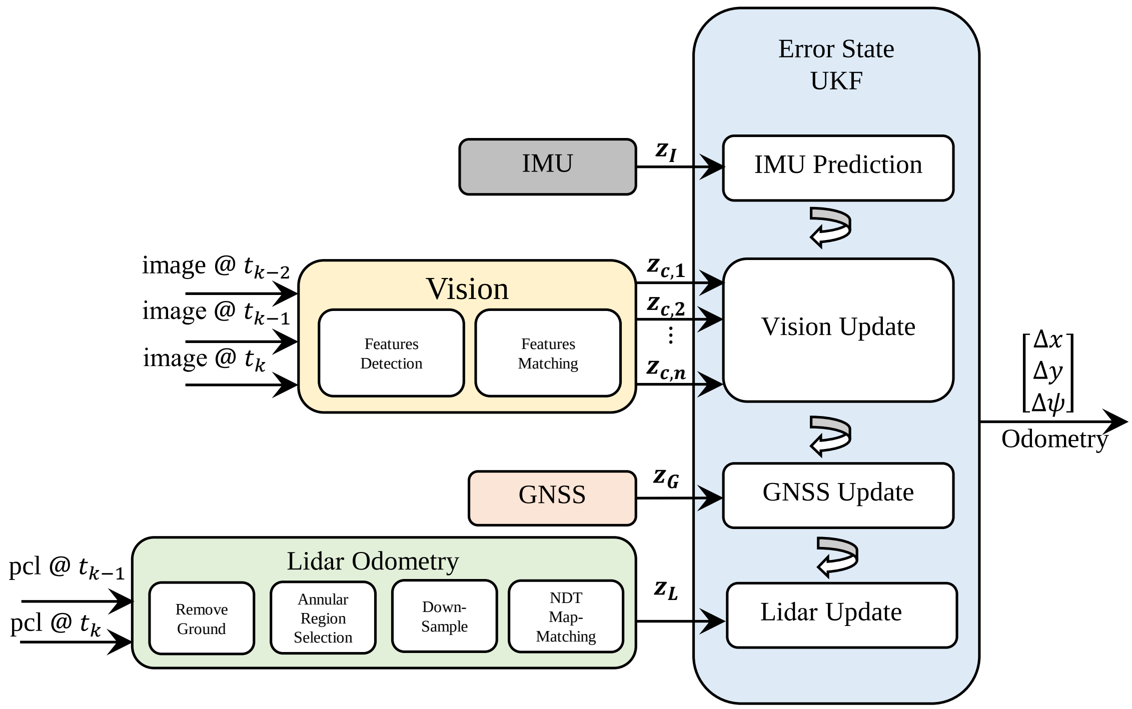


Figure 3.6: The schematic diagram of the ESUKF structure. The state is odometry is estimated based on the IMU kinematics and then updated upon receiving measurements from the camera, GNSS and Lidar.

Algorithm 1: Augmented Vehicle State Estimation

Initialize $\widehat{\mathbf{x}}_\mathbf{0}$, $\mathbf{P_{0|0}}$, and $\widetilde{\mathbf{x}}_\mathbf{0|0}=\mathbf{0_{27\times 1}}$
for k
{Filter Prediction}
Compute $\mathbf{F}_d$ and $\mathbf{Q}_d$ by (3.23) and (3.25)
$\widetilde{\mathbf{x}}_\mathbf{k|k-1}=\mathbf{0_{27\times 1}}$, $\mathbf{P}_{k|k-1}=\mathbf{F}_d\mathbf{P}_{k-1|k-1}\mathbf{F}_d^\top+\mathbf{Q}_d$
Predict $\widehat{\mathbf{x}}_\mathbf{k}$ by (3.18) and 4-th order Runge-Kutta method
Predict $\widetilde{\mathbf{x}}_\mathbf{k}$ by (3.22)
{Camera Measurement Update}
Match feature points in the last three frames to get $\{\mathbf{m_{k-2}},\mathbf{m_{k-1}},\mathbf{m_k}\}_\mathbf{i}$
Use RANSAC to find inliers
%%% Generate Sigma points and predict all the camera measurements %%%
$\widetilde{\mathbf{X}}^l_{k|k-1}=\mathbf{0}_{27\times 1}\pm\left(\sqrt{(L+\lambda)\mathbf{P}_{k|k-1}}\right)_l$
$\mathbf{Z}^l_i=\mathbf{h_c}\left(\mathbf{X_k},\{\mathbf{m_{k-2}},\mathbf{m_{k-1}},\mathbf{m_k}\}_\mathbf{i}\right)$, $\widehat{\mathbf{z}}_{c,i}=\sum_{l=0}^{2L}\mathbf{W}^l_s\mathbf{Z}^l_i$
%%% Update Error State and Error Covariance %%%
$\mathbf{P}_{\mathbf{z_{c,i}},\mathbf{z_{c,i}}}=\sum_{l=0}^{2L}\left(\mathbf{Z}^l_i-\widehat{\mathbf{z}}_{c,i}\right)\left(\mathbf{Z}^l_i-\widehat{\mathbf{z}}_{c,i}\right)^\top+\mathbf{R}$
$\mathbf{P}_{\mathbf{xz_{c,i}}}=\sum_{l=0}^{2L}\mathbf{W}^l_c\left(\widetilde{\mathbf{X}}^l_{k|k-1}-\mathbf{0}_{27\times 1}\right)\left(\mathbf{Z}^l_i-\widehat{\mathbf{z}}_{c,i}\right)^\top$
$\mathbf{K_k}=\mathbf{P_{xz_i}}\mathbf{P}^{-1}_{\mathbf{z_i z_i}}$
$\widetilde{\mathbf{x}}_{k|k}=\widetilde{\mathbf{x}}_{k|k-1}+\mathbf{K}_k(\mathbf{z}_i-\mathbf{P_{xz_i}})$
{GNSS Measurement Update}
if GNSS is not initialized
initialize GNSS using (3.14)
else
%%% Generate Sigma points and predict camera measurement %%%
$\widetilde{\mathbf{X}}^l_{k|k-1}=\mathbf{0}_{27\times 1}\pm\left(\sqrt{(L+\lambda)\mathbf{P}_{k|k-1}}\right)_l$
$\mathbf{Z}^l_G=\mathbf{h}_G\left(\mathbf{X}_k\right)$, $\widehat{\mathbf{z}}_G=\sum_{l=0}^{2L}\mathbf{W}^l_s\mathbf{Z}^l_G$
%%% Update Error State and Error Covariance %%%
$\mathbf{P}_{\mathbf{z_G},\mathbf{z_G}}=\sum_{l=0}^{2L}\left(\mathbf{Z}^l_G-\widehat{\mathbf{z}}_G\right)\left(\mathbf{Z}^l_G-\widehat{\mathbf{z}}_G\right)^\top+\mathbf{R}$
$\mathbf{P_{xz_G}}=\sum_{l=0}^{2L}\mathbf{W}^l_c\left(\widetilde{\mathbf{X}}^l_{k|k-1}-\mathbf{0}_{27\times 1}\right)\left(\mathbf{Z}^l_G-\widehat{\mathbf{z}}_G\right)^\top$
$\mathbf{K_k}=\mathbf{P_{xz_G}}\mathbf{P}^{-1}_{\mathbf{z_G z_G}}$
$\widetilde{\mathbf{x}}_{k|k}=\widetilde{\mathbf{x}}_{k|k}+\mathbf{K}_k(\mathbf{z}_G-\mathbf{P_{xz_G}})$
endif
{Lidar Measurement Update}
Remove points on the ground plane
Select points in the annular region
Random-down sample points
Perform NDT to register pcl @ t_{k-1} to pcl @ t_k
%%% Generate Sigma points and predict camera measurement %%%
$\widetilde{\mathbf{X}}^l_{k|k-1}=\mathbf{0}_{27\times 1}\pm\left(\sqrt{(L+\lambda)\mathbf{P}_{k|k-1}}\right)_l$
$\mathbf{Z}^l_L=\mathbf{h}_L\left(\mathbf{X}_k\right)$, $\widehat{\mathbf{z}}_L=\sum_{l=0}^{2L}\mathbf{W}^l_s\mathbf{Z}^l_L$
%%% Update Error State and Error Covariance %%%
$\mathbf{P}_{\mathbf{z_L},\mathbf{z_L}}=\sum_{l=0}^{2L}\left(\mathbf{Z}^l_L-\widehat{\mathbf{z}}_L\right)\left(\mathbf{Z}^l_L-\widehat{\mathbf{z}}_L\right)^\top+\mathbf{R}$
$\mathbf{P_{xz_L}}=\sum_{l=0}^{2L}\mathbf{W}^l_c\left(\widetilde{\mathbf{X}}^l_{k|k-1}-\mathbf{0}_{27\times 1}\right)\left(\mathbf{Z}^l_L-\widehat{\mathbf{z}}_L\right)^\top$
$\mathbf{K_k}=\mathbf{P_{xz_L}}\mathbf{P}^{-1}_{\mathbf{z_L z_L}}$
$\widetilde{\mathbf{x}}_{k|k}=\widetilde{\mathbf{x}}_{k|k}+\mathbf{K}_k(\mathbf{z}_L-\mathbf{P_{xz_L}})$
{State Marginalization}
%%% replace the old state with the current state and revise the covariance matrix %%%
$\widehat{\mathbf{x}}_k=\mathbf{T}_n\widehat{\mathbf{x}}_k$, $\widetilde{\mathbf{x}}_k=\mathbf{T}_n\widetilde{\mathbf{x}}_k$, $\mathbf{P}_{k|k}=\mathbf{T}_e\mathbf{P}_{k|k}\mathbf{T}_e^\top$
endfor

sensor modalities, including an OXTS RT3003 GNSS-Inertial Navigation System (INS), a 360-degree Lidar, and 4 cameras. More information about the specifications of the sensors is available on the corresponding website [77].

To model a low-grade GNSS, a zero-mean Gaussian noise to the ground truth data is added. The sensitivity of the odometry to GNSS sampling rate and accuracy is analyzed and discussed in Section 3.2.9. Additionally, a bias of $[0.01,\ -0.01,\ 0.02]$ rad/s is artificially added to the gyroscope measurements and a bias of $[0.1,\ -0.1,\ 0.2]\ \mathrm{m}/\mathrm{s}^2$ is added to the accelerometer measurements to represent a low-grade production-level IMU.

To evaluate the odometry system, the result of estimating the longitudinal displacement, lateral displacement, and the yaw angle change is examined,

$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = {}^{k-1}\,\mathbf{t}_k\ , \quad \Delta\psi = \psi_k - \psi_{k-1}\ , \tag{3.28}$$
$$\text{where}\ \begin{bmatrix}{}^{k-1}\mathbf{R}_k & {}^{k-1}\mathbf{t}_k\\ 0\ 0\ 0 & 1\end{bmatrix} = {}^{k-1}\,\mathbf{T}_k = \left({}^{N}\mathbf{T}_{k-1}\right)^{-1}\left({}^{N}\mathbf{T}_k\right)\ ,$$

${}^{N}T_k \in SE(4)$ is the transformation from the vehicle frame to the navigation frame, and ψ is the yaw angle with respect to the navigation frame. Note that the upward displacement of the vehicle (Δz) is not important for vehicle localization.

Figure 3.7 shows the odometry estimation results in two different maneuvers. In Figure 3.7a, the vehicle undergoes a maneuver with higher speeds and small steering angles in a highway environment. The environment contains mostly unstructured landmarks (e.g. vegetation) with distant visual features. This fact makes visual feature detection and matching hard. On the other hand, smaller steering angles lead to smaller excitation in the lateral acceleration measurements in IMU which is the input to the IMU dynamics in (3.3). As a result, it makes the estimation of odometry challenging. In Figure 3.7b, the vehicle is in the urban environment with lower speeds, larger steering angles, and multiple acceleration and deceleration events. Table 3.2 summarizes the accuracy of odometry estimation in the two cases. Accordingly, odometry estimation is more accurate in scenario b in which the environment is urban.

According to the results in Figure 3.7, the developed odometry system can estimate the longitudinal displacement and yaw angle change with very good accuracy. However, the result of the vehicle's lateral displacement seems to be not very accurate when compared to the ground truth. In fact, the accuracy of the ground truth signal is not enough to be comparable to the lateral displacement which has relatively low magnitudes for the following reasons. The ground truth signals are provided by an RTK-GNSS-INS system.

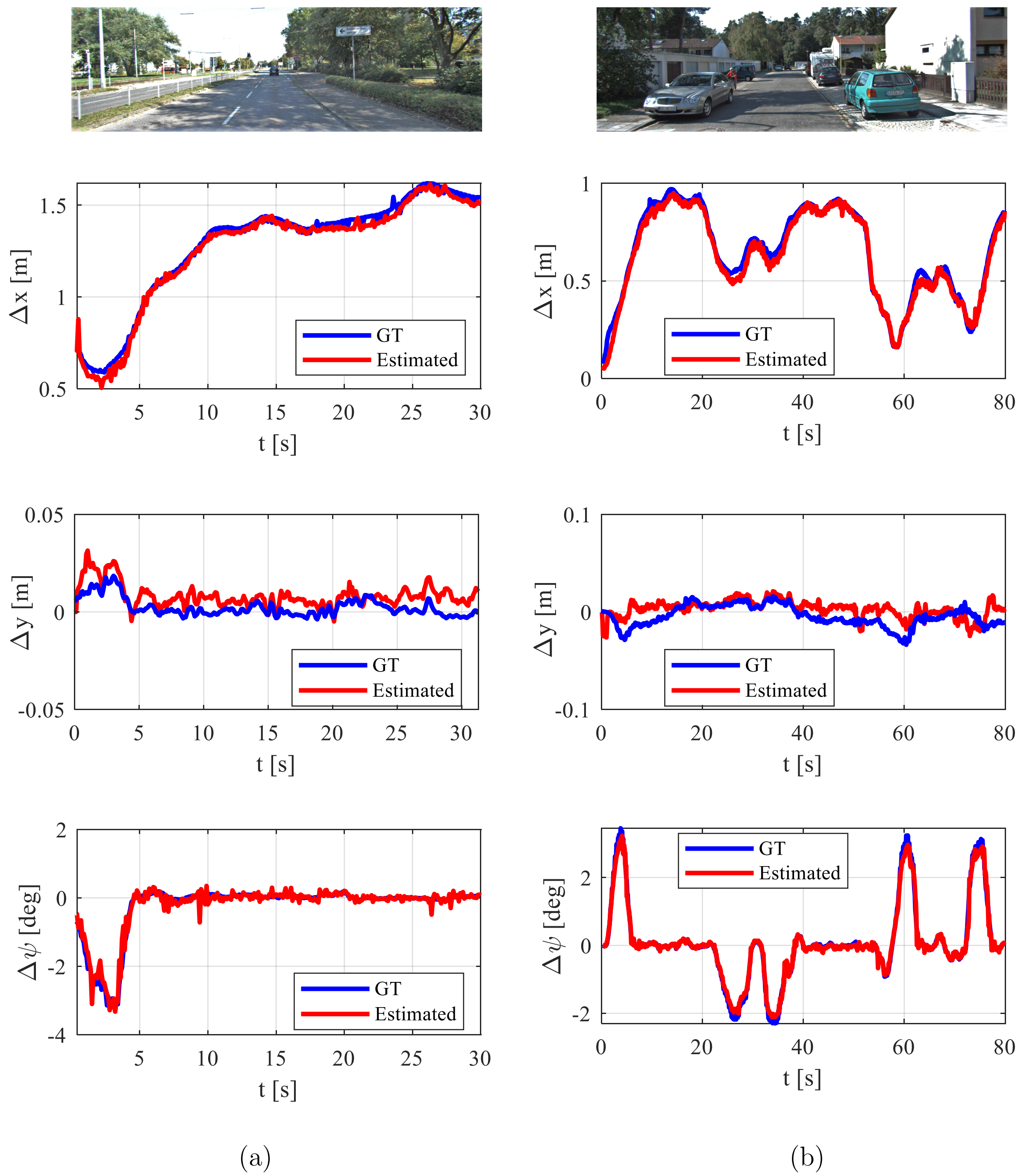


Figure 3.7: The results of odometry estimation for two scenarios. The scenario in (a) contains a vehicle driving in a highway environment at higher speeds. The scenario in (b) contains driving in an urban environment with lower speeds.

Table 3.2: RMSE of Vehicle Odometry estimation for the real world experiments

Scenario	RMSE($\Delta\hat{x}$)	RMSE($\Delta\hat{y}$)	RMSE($\Delta\hat{\psi}$)
(a)	0.0033m	0.0070m	0.1578°
(b)	0.0033m	0.0121m	0.1123°

The ground truth for lateral velocity is reported to have an accuracy of around 0.05m/s (see Figure 3.8). Given the fact that the time step is 0.1s, the accuracy of the lateral displacement is about 0.005m. Therefore, the signal-to-noise ratio of the ground-truth lateral displacement is small and hence it is not appropriate for the evaluation of lateral displacement estimation.

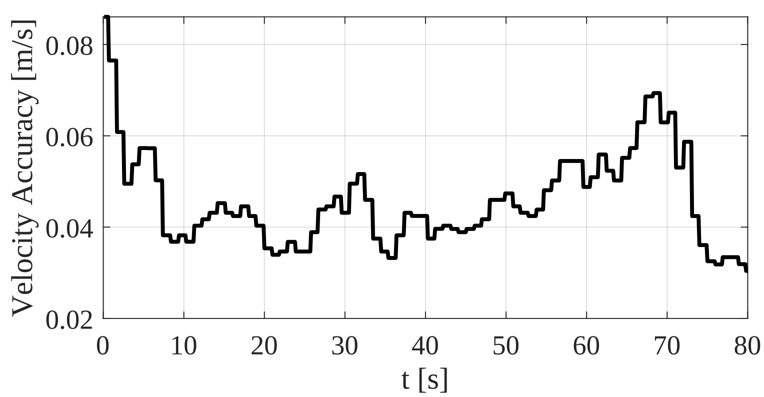
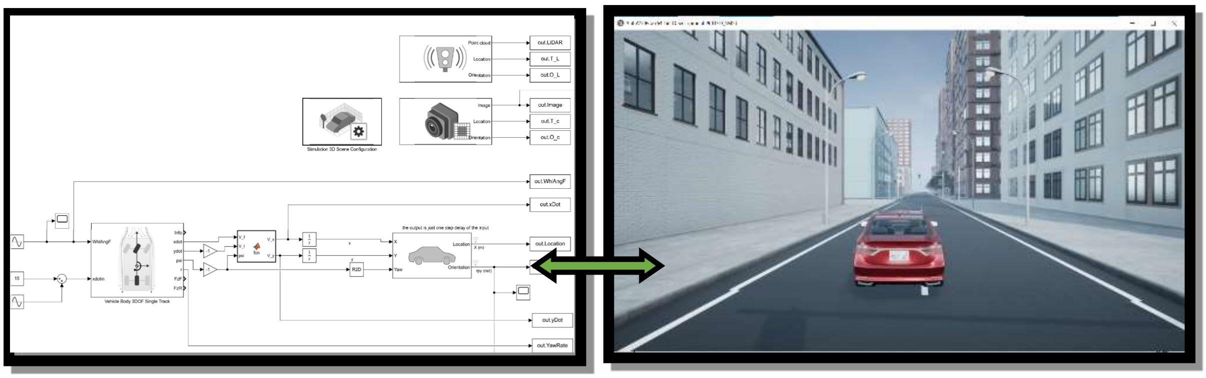


Figure 3.8: The velocity measurement of scenario b reported from the RTK-GNSS-INS system that is used for ground truth. The accuracy is around 0.05m/s accuracy which corresponds to 0.005m accuracy of displacement given that the time step is 0.1s.

To be able to evaluate the model-based odometry estimation system in estimating the lateral displacement, there is a need to have some dataset that contains maneuvers with accurate ground-truth information. Accordingly, a driving scenario was designed in MATLAB Simulink along with Unreal Engine that contains higher-speed driving with large steering inputs (Figure 3.9). The sensor configuration (camera, Lidar, GNSS, IMU extrinsic, and intrinsic calibrations) and the vehicle's specifications were chosen to be as close as possible to the configurations in real-world experiments. Given the fact that the experiment is in the simulation environment, a precise ground truth signal is available for evaluating the estimation.



MATLAB/Simulink Automated Driving Toolbox   Unreal Engine

Figure 3.9: MATLAB/Simulink provides the functionality of generating and running a driving scenario in Unreal Engine.

Figure 3.10 shows the odometry results based on the simulation. After 4s, the estimated signals converge to the ground truth signals. Accordingly, in addition to the longitudinal displacement and yaw angle displacement, the lateral displacement is estimated accurately.

3.2.9 GNSS Sensitivity Analysis

In this section, the effect of GNSS specification on estimation performance is studied. The goal is to see the effect of GNSS measurements frequency and noise level on the longitudinal odometry estimation accuracy. Therefore, different levels of noise in GNSS measurements ($\sigma_{\text{GNSS},x}, \sigma_{\text{GNSS},y} = 1,\ 2,\ 5,\ 10$m) and different GNSS measurement frequencies ($f = 1, 2, 5, 10$ Hz) are considered. Figure 3.11 illustrates GNSS measurement at different rates while the other sensors are at the same rate and synchronized together.

Table 3.3 and Table 3.4 summarize the result of sensitivity analysis of using GNSS to different levels of sampling frequency and measurement noise, respectively. Accordingly, the sampling frequency and GNSS noise level have a small effect on the estimation accuracy meaning that the system is reliable to be used with low-cost low-frequency GNSS sensors.

The results show that it can benefit from low-grade low-rate GNSS sensors and remain robust to the GNSS intermittent signal loss. The system is designed to be reconfigurable (modular) and to support multi-rate synchronized sensors so it can fit the needs and availability of different onboard sensors of a vehicle.

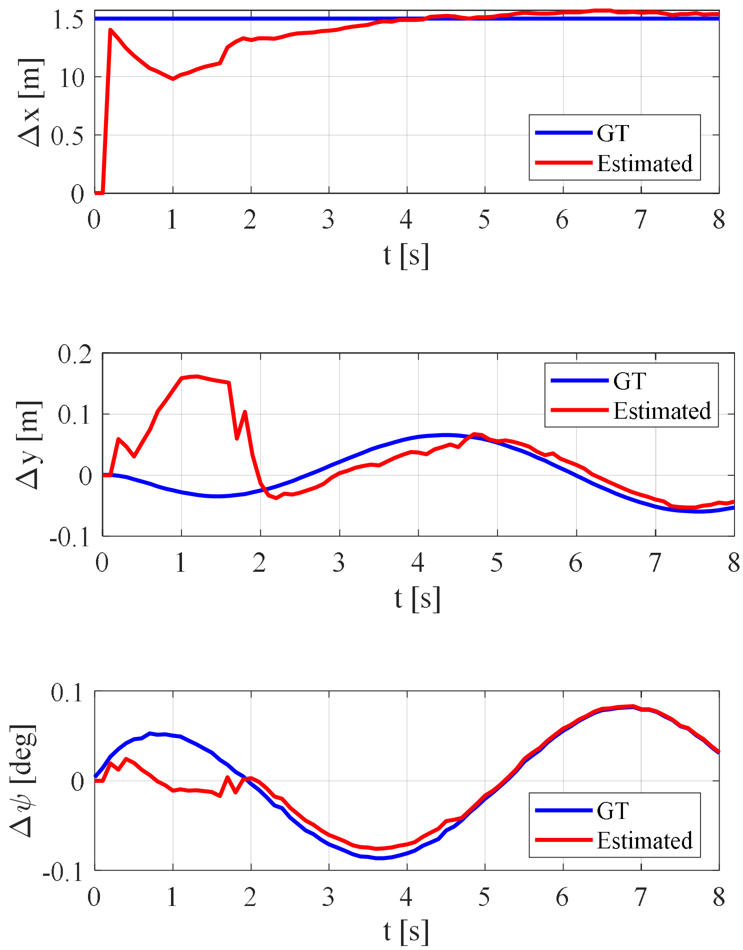


Figure 3.10: The odometry results based on simulation experiments. Throughout the first 4 seconds of the maneuver, the estimated states converge close to the ground truth values.

Table 3.3: The result for analyzing the sensitivity of odometry estimation accuracy to the sampling frequency of GNSS. The sampling frequency of 1Hz is considered the baseline for the comparison.

GNSS Sampling Rate(Hz)	1	2	5	10
% Change in RMSE($\Delta\hat{x}$)	0%	-0.337%	-0.270%	2.02%

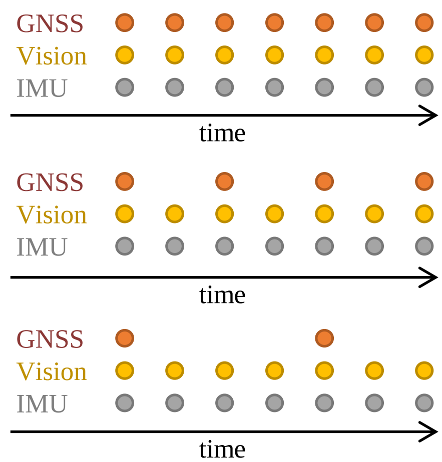


Figure 3.11: A visual representation of different GNSS measurement signals with different sampling frequencies ($f = 1,\ 2,\ 5$ Hz).

Table 3.4: The result for analyzing the sensitivity of odometry estimation accuracy to the standard deviation of GNSS noise. The noise level of 1m is considered the baseline for the comparison.

$\sigma(n_{\text{GNSS}})(m)$	1	2	5	10
% Change in RMSE($\Delta\hat{x}$)	0%	+0.0677%	+11.38%	+8.47%

3.3 Machine-Learning-Based Vehicle Odometry

In the previous section, exteroceptive sensors were mostly used to estimate the odometry of the vehicle by observing the environment's motion induced by the vehicle's ego-motion. In this section, as an alternative approach to estimating vehicle odometry, a machine-learning model is designed and tested that uses the vehicle's proprioceptive sensor, including IMU and wheel encoders.

A Nadaraya-Watson Kernel Regression (NWKR) model is used which is an estimation method that interpolates the output value for a given point, based on the observations around that area. This algorithm works based on the weighting average. The weights used in the NW method are calculated using a kernel function. In this kernel function, based on the distance between the point of interest $\mathbf{x} \in \mathbb{R}^p$ and observations around that point (reference data) $\mathbf{x}_i$, a value will be assigned [78, 79]. NWKR uses the following equation to estimate $\widehat{y} = \widehat{f}_n(\mathbf{x})$ for the point of interest $\mathbf{x}$, based on nearest n points:

$$\widehat{f}_n(\mathbf{x}) = \frac{\sum_{i=1}^{n} K\left(\mathbf{x}, \mathbf{x}_i\right) y_i}{\sum_{i=1}^{n} K\left(\mathbf{x}, \mathbf{x}_i\right)} , \tag{3.29}$$

where K which is used as weights in this equation is a multivariate Gaussian kernel:

$$K\left(\mathbf{x}, \mathbf{x}_i\right) = \frac{1}{(2\pi)^{d/2} \, |\mathbf{H}|^{0.5}} \exp\left(-\frac{\left(\mathbf{x} - \mathbf{x}_i\right)^{\top} \mathbf{H}^{-1} \left(\mathbf{x} - \mathbf{x}_i\right)}{2}\right) , \tag{3.30}$$

where $\mathbf{H}$ is a positive definite covariance matrix that is used to reflect the correlation between different features and can be used as a controlling parameter that tunes the smoothness of the regression. In this work, three NWKR models are developed for estimating longitudinal and lateral displacement as well as yaw angle change.

3.3.1 Input Feature Selection

The measurements coming from the vehicle's proprioceptive onboard sensors, including IMU and encoders, provide information about different physical quantities that can be used as input features to the NWKR model. However, a model that uses a minimal subset of effective input features that are highly correlated to the output is desired to reduce overall computational cost and improve the accuracy of the estimation algorithm. In this regard, to evaluate the quality of different subsets of input features, Akaike's Information Criterion (AIC) is used [80]:

$$AIC = n\log(RSS/n) + 2p \ , \tag{3.31}$$

where n is the number of points in the training dataset, p is the number of input features and RSS (Residuals Sum of Squares) is

$$RSS = \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 \ . \tag{3.32}$$

Akaike's information criterion (AIC) balances the goodness of fit by reducing RSS and penalizing the model complexity by reducing the size of the feature set p. AIC is defined such that the smaller the value of AIC is, the better the model will be.

The equations of the governing vehicle dynamics are

$$\begin{aligned} a_x &= \dot{u} - vr + b_{ax} + n_{ax} \\ a_y &= \dot{v} + ur + b_{ay} + n_{ay} \ , \end{aligned} \tag{3.33}$$

where a_x and a_y are x and y components of acceleration, measured by IMU, b_{ax} and b_{ay} are acceleration biases, and n_{ax} and n_{ax} are the measurement noises, respectively. Additionally, the steering angle δ and wheel encoder ω_w are considered in the feature set due to the correlation to the yaw rate and speed, respectively. Accordingly, the set of effective signals is $\{\omega_w, r, a_x, a_y, \delta\}$. The feature set is used to select a subset of features based on the goodness of fit using AIC. A k-fold cross-validation approach with $k = 10$ is used to test the model's performance in generalizing over unseen data. The result of AIC is presented in Figure 3.12.

According to AIC results in Figure 3.12, to estimate the longitudinal displacement, wheel encoders provide enough information for longitudinal displacement estimation. This is because the environment that the odometry is designed for is about a low-speed shuttle operating in minimal slips. On the other hand, although including a_x and other feature into the input features set may seem reasonable, the AIC results show that it can inject some additional uncertainty (bias and noise) into the model that degrades longitudinal displacement estimation eventually which is reflected in the AIC results.

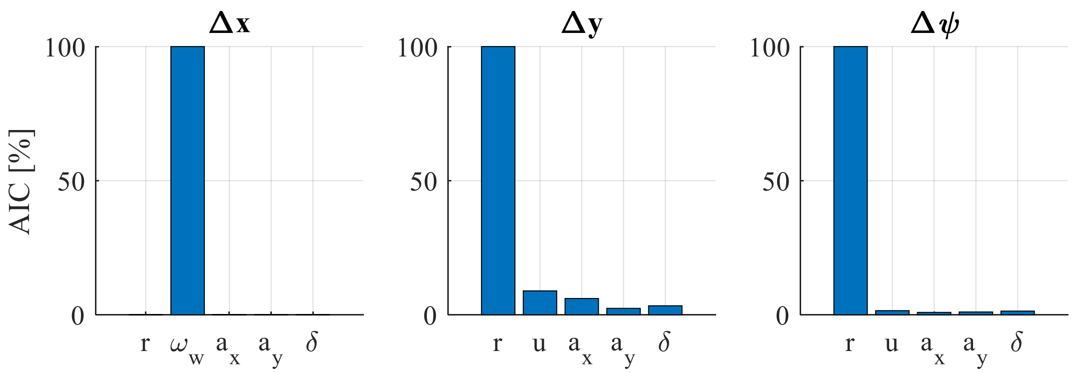


Figure 3.12: The AIC results for the input feature selection for odometry estimation.

For the heading change estimation in (3.35), one-step integration of the yaw rate can be suggested due to the large AIC of the yaw rate in Figure 3.12. However, the yaw rate measurements from IMU can be contaminated with time-varying bias, which results to drift in the odometry estimation. To account for yaw rate bias, the steady-state dynamics of yaw rate in the steady-state form are considered.

$$\begin{aligned} r &= \frac{1}{1-Ku^2}\frac{u}{l}\delta \\ K &= \frac{m\left(l_f C_f - l_r C_r\right)}{l^2 C_f C_r} \end{aligned} \quad , \tag{3.34}$$

where $l = l_r + l_f$ is the wheelbase, l_f is the vehicle's Center of Gravity (CG) to the front axle, l_r is the vehicle's CG to rear axle, C_f and C_r are the front and rear tire cornering stiffness of the vehicle, and m is the mass of the vehicle. According to (3.34), the yaw rate is determined by speed and steering angle. Hence, the steering wheel angle and speed are used among the input features set in $\widehat{f}_\psi$ in (3.35).

Finally, based on the AIC results in Figure 3.12 and the vehicle dynamics in (3.34) the following regression models are designed for odometry prediction:

$$\begin{aligned} \Delta\widehat{x} &= \widehat{f}_x\left(\omega_w\right) + n_x \\ \Delta\widehat{y} &= \widehat{f}_y\left(r, \omega_w, a_x, a_y, \delta\right) + n_y \\ \Delta\widehat{\psi} &= \widehat{f}_\psi(r, \omega_w, \delta) + n_\psi \end{aligned} \quad . \tag{3.35}$$

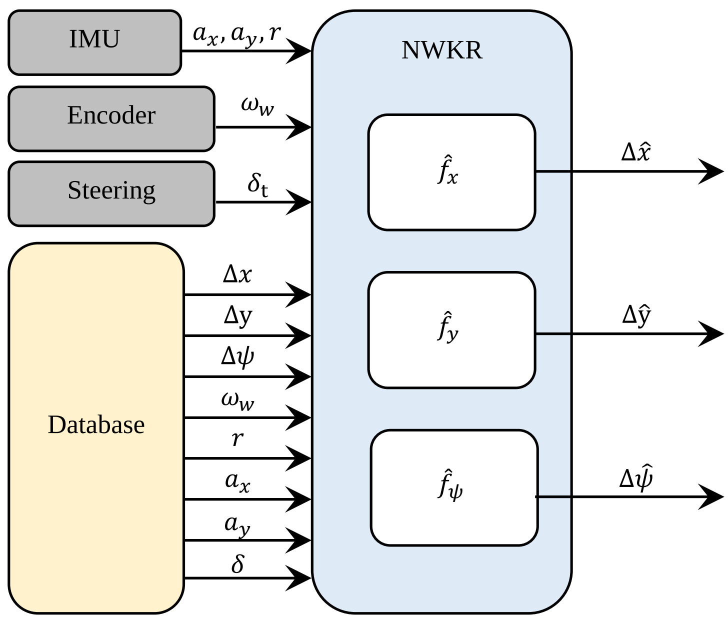


Figure 3.13: The schematic diagram of the machine-learning-based odometry estimator. Given the historical measurement from a database, the odometry is estimated based on the IMU measurements, wheel encoder, and steering angle.

Figure 3.13 illustrates the structure of the machine-learning-based odometry estimator.

3.3.2 Results and Discussion

The designed regression models were tested over unseen data collected by WATonoBus while operating on the Ring Road. The odometry output is provided by RTK-GNSS-INS with centimeter-level accuracy. Wheel speed is obtained from the encoder mounted on the electric motor. As seen in Figure 3.14, the vehicle operates at low speeds and small steering angles. The training dataset contains the recent historical data of the input features over the last 470 s of driving. All the input features to the model are normalized to have zero mean and unit standard deviation for better performance of the model.

Figure 3.15 illustrates the estimation results of vehicle odometry which shows very good estimation performance. Table 3.5 summarizes odometry prediction performance based on Root Mean Squared Error (RMSE). In the context of visual odometry and SLAM, the RMSE of displacements is usually referred to as Relative Trajectory Error (RTE).

One factor that can potentially degrade any observer's performance is the presence of bias in the measurement signals. The designed regression model is quite robust to the presence of bias in the input measurements. For the yaw odometry regression model, in the case of having $0.24°/s$ added bias in the yaw rate measurements, RMSE($\Delta\psi$) increases by $0.0111°$. This is due to the presence of redundancy in input features by having longitudinal velocity, lateral acceleration, and steering angle which is 12% less than a regression model that uses yaw rate as the only input feature. Obviously, including rich training data covering different biases can improve the performance of the regression model.

On the other hand, when RTK-INS solutions are available during the vehicle operation, the regression model uses the recent window of historical measurements for training data and hence it relies more on the more recent data and forgets the older ones. Since the bias in IMU has slow dynamics over time, the prediction will be still valid since it is using reference data with equal biases (Figure 3.16).

Finally, The obtained RMSE values in Table 3.5 are used as the level of uncertainty of the machine learning-based model in the fusion with other localizers in Chapter 4.

3.4 Summary

In this chapter, a model-based vehicle odometry system was developed to estimate the vehicle's odometry using IMU, camera, Lidar, and GNSS. Based on the experiments that

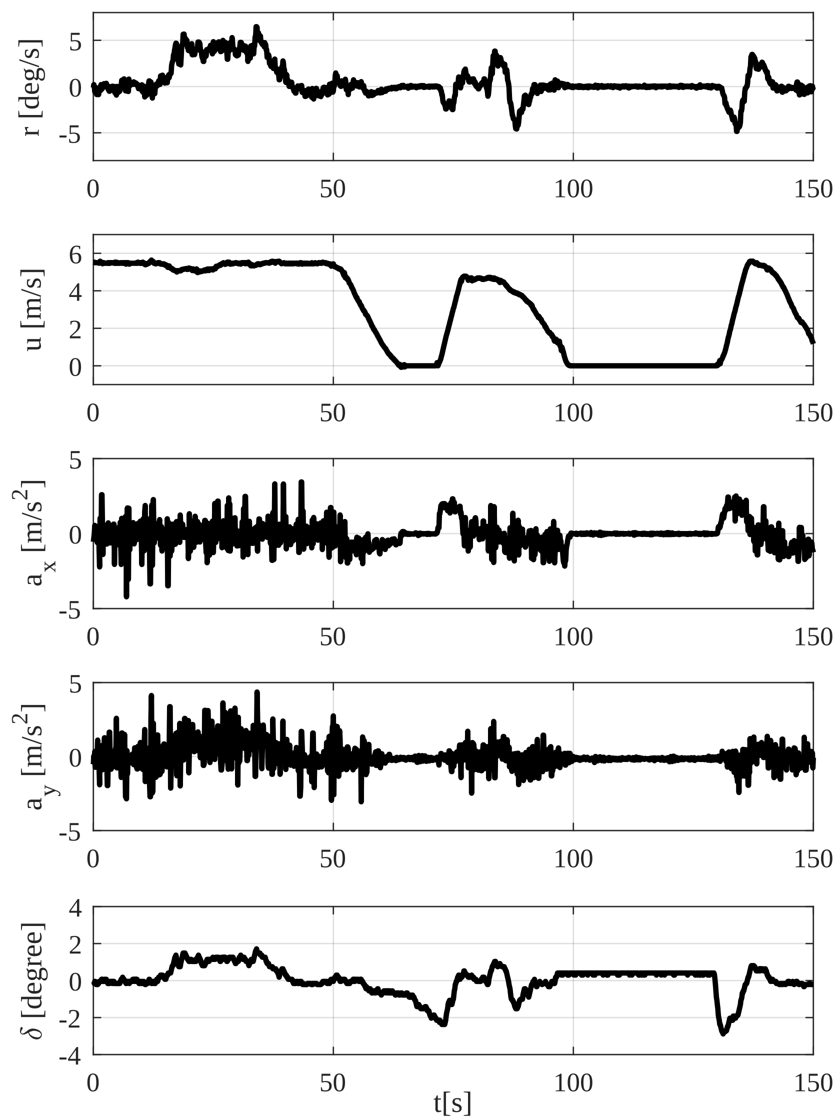


Figure 3.14: To evaluate the performance of the developed ML-based odometry system, a test dataset is collected from WATonoBus while operating around the ring road at the University of Waterloo campus. The WATonoBus is designed as a low-speed shuttle that operates at low speeds and small steering angles that result in relatively low accelerations.

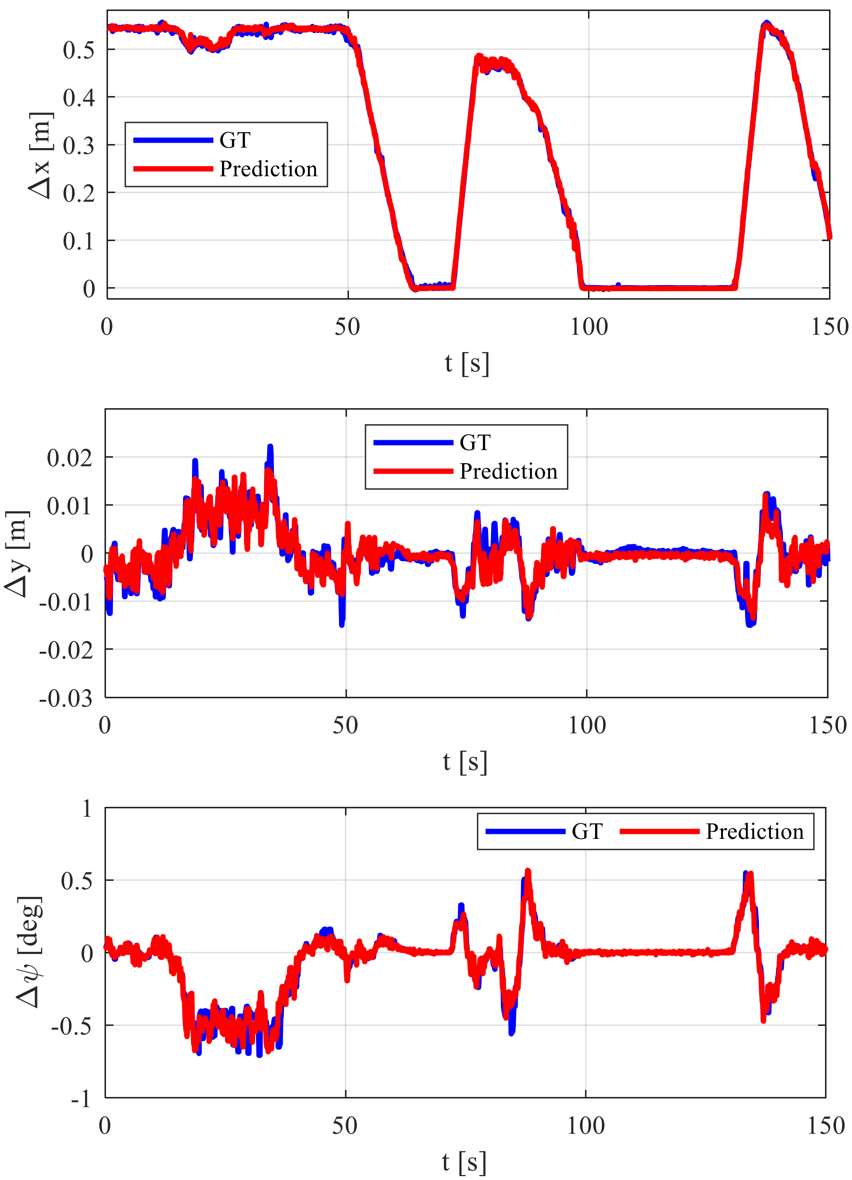


Figure 3.15: The results of estimating the vehicle odometry using the developed ML regression model. Accordingly, the developed regression model can estimate the vehicle lateral displacement (Δx), longitudinal displacement (Δy), and yaw angle change ($\Delta\psi$) accurately

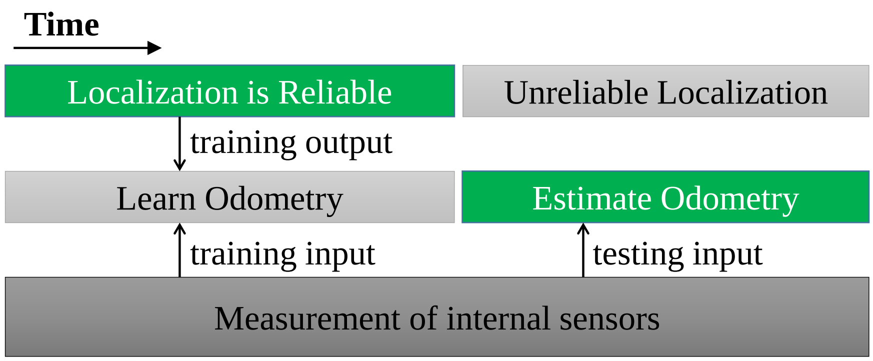


Figure 3.16: The procedure of data collection and model implementation. When the localization is reliable, the odometry system receives localization output along with measurements of internal sensors, to learn odometry. Then at a time that the localization becomes unreliable, the odometry system would use sensor measurements in addition to the collected reference data to relatively self-localize until the self-localization system becomes reliable once again.

Table 3.5: RMSE of Vehicle Odometry Prediction

RMSE($\Delta\hat{x}$)	RMSE($\Delta\hat{y}$)	RMSE($\Delta\hat{\psi}$)
0.0061m	0.0022m	0.076°

were performed in real-world and simulated environments, it has been shown that the model-based system has promising performance in estimating the odometry of a vehicle in different conditions. Additionally, the results of the GNSS sensitivity analysis showed that the developed odometry system can operate reliably with a GNSS with low sampling rates and accuracy.

On the other hand, a machine-learning-based odometry system was developed to compensate for the unavailability of accurate self-localization. Throughout the feature selection analysis, the best sets of input features were determined. The experimental results showed superior performance in odometry estimation by considering structural and measurement uncertainties through reference data.

Chapter 4

Vehicle Self-localization

4.1 Introduction

The objective of vehicle localization is to obtain the location and the heading of the vehicle with respect to a stationary frame. In landmark-based localization, self-localization is estimated based on comparing the observed landmarks with the map information. There is often some redundancy in the availability of landmarks around the vehicle, which gives the system the flexibility to attend to only a subset of observable landmarks in real-time while the localization problem remains observable and efficient. To do so, there is a need to quantify the uncertainty of different landmarks.

In this chapter, a new self-localization system is developed that uses geometric information of landmarks that are observable and invariant to seasonal changes. Based on the developed uncertainty models, situation- and uncertainty-aware attention mechanisms are developed to fuse various sources of information according to the uncertainty level.

The rest of this chapter is organized as follows. Section 4.2 formulates the self-localization task as an optimization problem. The landmark-based extrinsic calibration of Lidar is discussed in Section 4.3. Section 4.4 presents the uncertainty quantification of measurements and the developed attention mechanisms. Finally, Section 4.5 provides a summary.

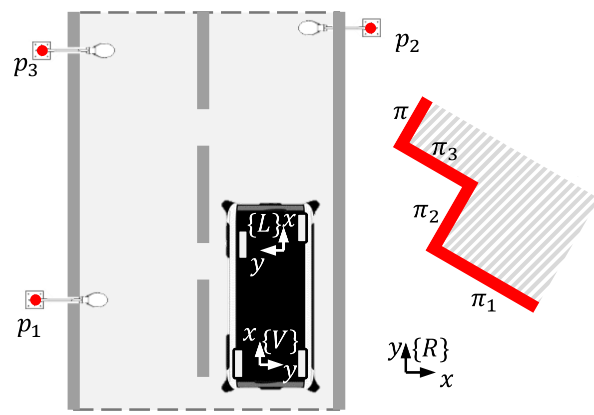


Figure 4.1: A bird-eye view of the ego-vehicle navigating within a known environment. The objective is self-localizing the frame {V} with respect to frame {R}. The self-localization is performed through relatively localizing nearby stationary landmarks, including light poles $\{p_1, p_2, p_3\}$, and building planes $\{\pi_1, \pi_2, \pi_3, \pi_4\}$.

4.2 Landmark-Based Self-Localization

As the vehicle drives in any drivable location within a known environment, the system uses the observation of nearby landmarks to self-localize. It selects the least yet enough suitable landmarks to self-localize efficiently and reliably while considering the motion constraints based on vehicle odometry (Figure 4.1).

4.2.1 Parametrization of Drivable Space

For a vehicle moving in a known environment along a known road, a one-dimensional path coordination system is defined, called the s-coordinate system, which describes the vehicle's location with a single scalar. The whole drivable space is then discretized into a finite number of equally spaced intervals along an s-curve (Figure 4.2). The scalar value that is used for parametrization is a member of a set called S-set:

$$s \in S = \{s_1, s_2, \ldots, s_n\} \ ,$$

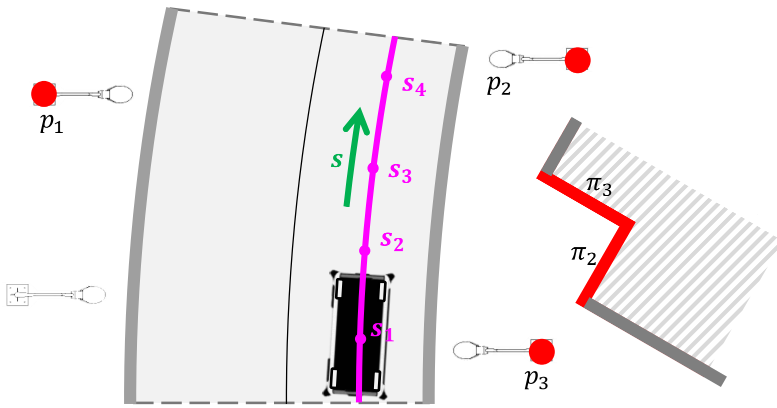


Figure 4.2: Parametrization of drivable space with a path coordinate system by defining equally distributed discrete s-coordinates along the s-curve. At each s-coordinate, only a subset of landmarks is observable.

which is the discretized arc length of the curve used for discrete parametrization. Since the s-curve geometry is known, the s-coordinate that is closest to the vehicle can determine the approximate location of the vehicle with some bounded error. Approximating the position by the closest s-coordinate results in a lateral error that cannot exceed more than the road's width (unless the vehicle becomes unstable and goes off the road) and a longitudinal error that cannot exceed the resolution of the s-coordinates.

4.2.2 Geometrical Model of Landmarks

Different planimetric landmarks, including building facades (outline of building footprint), light poles, and road geometry (curbs and lane markings) are represented by simple parametric models such as points, lines, and curves, independent of elevation.

Building facades are typically built as vertical planes locally normal to the ground. A vertical plane can be modeled as a 2D line on the horizontal ground plane with no elevation (xy-plane is the horizontal plane and z is toward the normal to this plane). The following equation represents a constraint for a point $\mathbf{p} = (x, y)$, $\pi_1, \pi_2, x, y \in R$ that is on a line:

$$\pi_1 x + \pi_2 y + 1 = 0 \rightarrow \bar{\pi}^\top \bar{p} = 0 \ , \tag{4.1}$$

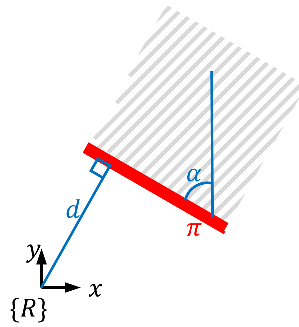


Figure 4.3: A vertical plane which is modeled as a line in xy-plane is parametrized by its distance to the origin and its angle

where $\bar{\pi} = (\pi_1, \pi_2, 1)$ is the homogeneous line coefficient and $\bar{p} = (x, y, 1)$ is the homogeneous location of the point. The following equation defines a 2D rigid transformation $\mathbf{T} \in SE(2)$ that maps a point $\bar{\mathbf{p}}$ to a new point $\bar{\mathbf{p}}'$:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & t_x \\ \sin\psi & \cos\psi & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} . \tag{4.2}$$

Through the same transformation, a line $\bar{\pi}$ will be mapped to a new line $\bar{\pi}'$ based on the following equation [19]:

$$\bar{\pi}' = \mathbf{T}^{-\top} \bar{\pi} \ . \tag{4.3}$$

According to Figure 4.3, an alternative representation for the vertical plane is to parametrize its distance to the origin and its angle:

$$\alpha = \tan^{-1}\left(-\frac{\pi_1}{\pi_2}\right), d = \frac{1}{\sqrt{\pi_1^2 + \pi_2^2}} \ . \tag{4.4}$$

The distance-angle geometric modeling in (4.4) is used for uncertainty quantification in Section 4.4.3.

The structure of pole-like objects such as light poles contains a vertical long pole with typically round or polygon cross-section. A pole is represented by $\mathbf{p} = (x, y)$ on the horizontal plane which is the center point of the cross-section.

To model curb and lane markings, a continuous parametric 2-D curve c is used that is represented as:

$$\mathbf{c}(s) = [c_x(s), c_y(s)]^\top , \tag{4.5}$$

where $c_x(s)$ and $c_y(s)$ are the x and y coordinates of the curve, respectively, which are parameterized by the arc length parameter $s \in \mathbb{R}$. Accordingly, a road boundary, whether a curb or a lane marking, is represented as a single parametric curve. For simplicity, a curve $\mathbf{c}$ is modeled by a piece-wise linear curve over the discrete domain S that results in n control points:

$$\mathbf{c} = \begin{bmatrix} c_{1,x}, c_{1,y} \\ c_{2,x}, c_{2,y} \\ \vdots \\ c_{n,x}, c_{n,y} \end{bmatrix}, \quad \mathbf{c} : S \to \mathbb{R}^2 \ . \tag{4.6}$$

The control points are stored to form the HD vector map of the curb.

4.2.3 Definition of HD Vector Map

For the map-based self-localization, a mapping of the landmarks is considered to be available with the following definition:

$$M = \{\pi_1, \pi_2, \ldots, \pi_{N_\pi}\} \cup \left\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{n_p}\right\} \cup \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n\} \ . \tag{4.7}$$

The set of landmarks can be constructed using mobile surveying. As the surveying vehicle goes along the road, nearby landmarks are observed and stored in the HD Vector Map. The procedure and the results regarding the mapping of the Univerity of Waterloo campus are presented in Section 5.3.

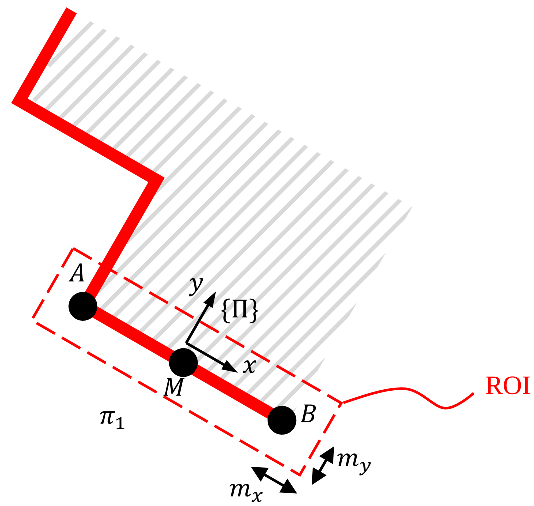


Figure 4.4: Definition of the Region-of-interest around a façade plane. A cuboid is constructed that is aligned with the plane and covers the points that fall onto the plane with added margin.

4.2.4 Landmark Extraction by Region-of-Interest

Lidar points that fall onto building façades and poles are used to generate their geometrical model. Accordingly, a Region of Interest (ROI) approach is used to select the point cloud around a landmark. The ROI for a building facade is defined as a cuboid with an infinite height that surrounds the façade plane plus some margins (Figure 4.4). To construct the ROI, the following procedure is done; given the two endpoints of a building wall AB, named ${}^{R}A$ and ${}^{R}B$, a frame $\{\Pi\}$ is constructed whose x-axis is aligned to the wall and y-axis whose s normal to the wall. To select the points in the ROI, the point cloud is transformed from the reference frame $\{R\}$ to the plane frame $\{\Pi\}$ via the following transformation:

$$
{}^{R}\bar{\mathbf{p}} = ({}^{R}\mathbf{T}_{\Pi})\,{}^{\Pi}\bar{\mathbf{p}} \quad \text{where} \quad {}^{R}\mathbf{T}_{\Pi} = \begin{bmatrix} \hat{n}_y & -\hat{n}_x & 0 & m_x \\ -\hat{n}_x & \hat{n}_y & 0 & m_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.8}
$$

where $M = (m_x, m_y, 0)$ is the middle point of the plane,

$$M = 0.5({}^R A +^R B) \ , \tag{4.9}$$

and $n = [n_x, n_y, 0]^\top$is the unit normal vector of the plane,

$$\hat{\mathbf{n}} = \mathbf{n} / \|\mathbf{n}\| \ , \quad \mathbf{n} = [0, 0, 1] \times (A - B) \ . \tag{4.10}$$

The set of points inside the ROI is defined by the following set:

$$ROI = \left\{ {}^{\Pi}\mathbf{p} = (x, y) \Big| |x| < \frac{|AB|}{2} + m_x \ \ \& \ \ |y| < m_y \right\} \ , \tag{4.11}$$

where m_x and m_y are real positive scalars that represent margins along x and y axis, respectively.

For a light pole, an ROI is formed by a cuboid around the expected location of the pole with a large height and limited width and height.

4.2.5 Robust Model Fitting of Landmarks

Since the ROI is constructed based on the expected location of a landmark plus some margin, it may contain some outlier points that belong to the surrounding of the landmark which should be removed. To reject the outliers, a Random Sample Consensus (RANSAC) approach is used during landmark model fitting [81]. RANSAC iteratively selects a minimum number of random samples (points) from all the points in ROI to create a geometrical model hypothesis in each iteration. A score for the rest of the points in ROI is calculated, showing how much they support the geometrical model hypothesis. The criteria for classifying a point as an outlier is to compare the distance of the point to the landmark with some threshold. If the number of supportive points goes beyond some predefined threshold, then the hypothesis would be selected, and the points that do not support the hypothesis are considered outliers. Otherwise, a new random sample set of points is selected, and the process is repeated. For light poles, the median of the points' location is selected as the geometrical model that can further increase the robustness to the presence of outliers.

4.2.6 Measurement and Motion Models

To estimate self-localization, the information coming from different sensor measurements along with dynamical models is fused together. A low-grade GNSS sensor is considered to be used in the sensory suite that measures the global position of the sensor. The main reason for considering a GNSS is to provide a rough initial solution to the optimization problem in case the odometry is not available. The global coordinate system is the Universal Transverse Mercator (UTM) coordinate system, which expresses the location as northing, easting, and height, in which the z-direction is aligned with the gravity direction:

$$\mathbf{m}_G = \begin{bmatrix} \text{northing} \\ \text{easting} \\ \text{heading} \end{bmatrix} . \tag{4.12}$$

The GNSS measures the position in $\{G\}$ frame which is assumed to be contaminated with a zero-mean Gaussian noise. Accordingly, the GNSS measurement model is defined as:

$$\mathbf{m}_G = h_G\left(\mathbf{X}_k\right) + \mathbf{n}_{\text{GNSS}}, \mathbf{n}_{\text{GNSS}} \sim N\left(0, \mathbf{\Sigma}_G\right) \tag{4.13}$$

On the other hand, an online observation of a landmark along with its geometrical model in the HD vector map imposes a constraint on the self-location of the vehicle (Figure 5.6). According to (4.3), the residue of observing a plane in the Lidar frame (${}^{L}\hat{\bar{\pi}}$) given its model in the reference frame (${}^{R}\hat{\bar{\pi}}$) is:

$$\mathbf{r}_\pi =^R \hat{\bar{\pi}} - \mathbf{T}\left(\psi, x_k, y_k\right)^{-\top} \left({}^{L}\hat{\bar{\pi}}\right) . \tag{4.14}$$

The residue of observing a light pole in the Lidar frame (${}^{L}\hat{\bar{\mathbf{p}}}$) given its model in the reference frame (${}^{R}\hat{\bar{\mathbf{p}}}$) is:

$$\mathbf{r}_p =^R \hat{\bar{\mathbf{p}}} - \mathbf{T}\left(\psi, x_k, y_k\right) \left({}^{L}\hat{\bar{\mathbf{p}}}\right) . \tag{4.15}$$

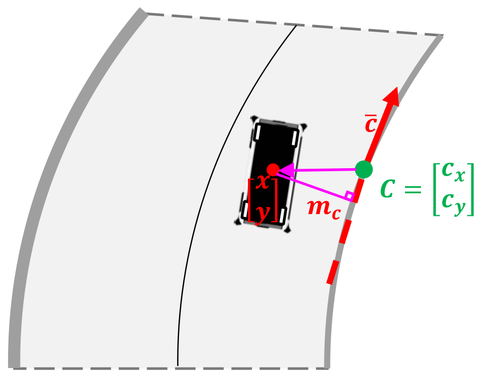


Figure 4.5: The normal distance to the curb is computed by using a curb unit tangent vector and a point on the curb

Additionally, during the online operation, a curb/lane detection module provides nearby curb points in the vehicle frame. Given the control points of the curb curve in the HD map, a constraint can be considered over the self-location of the vehicle. Obviously, the detected curb points do not necessarily correspond to the control points of the vector map. Therefore, the curb constraint is formed based on the distance to a line fitted over the detected curb points. To do so, a reweighted least squares approach with the bi-square weighting function is used to estimate the linear model over the control points that contain outliers [82]. The distance to the fitted line is used as the final measurement of the curb detection module. According to Figure 4.5, given the s-position at which the vehicle is located, the closest control point of the vector map is found and the corresponding curb unit tangent vector ($\bar{\mathbf{c}}$) is calculated based on the vector formed by the two closest control points. The constraint for the curb measurement is formed as:

$$m_c = \left\| \left(\begin{bmatrix} x \\ y \end{bmatrix} - \mathbf{c} \right) \times \bar{\mathbf{c}} \right\| \rightarrow r_c = m_c - \left| (x_k - x_c)\, \bar{c}_y - (y_k - y_c)\, \bar{c}_x \right| \;, \tag{4.16}$$

in which m_c is the normal distance to the online detected curb points, $\mathbf{c} = (x_c, y_c)$ is the nearest curb control point, and $\bar{\mathbf{c}} = (\bar{c}_x, \bar{c}_y)$ is the unit tangent to the curb curve. Following the same procedure, the residual for lane detection is formed as:

$$r_l = m_L - \left| (x_k - x_l)\, \bar{l}_y - (y_k - y_l)\, \bar{l}_x \right| \;, \tag{4.17}$$

in which m_l is the normal distance to the online detected lane points, $\mathbf{l} = (x_l, y_l)$ is the nearest lane control point, and $\bar{\mathbf{l}} = \left(\bar{l}_x, \bar{l}_y\right)$ is the unit tangent to the lane curve.

Additionally, the translation and rotation of the vehicle follow the vehicle kinematics that imposes constraints over two consecutive self-localization states. Based on the odometry estimation, the motion of the vehicle in the body frame is used to form the odometry residual:

$$\mathbf{r}_o = \mathbf{m}_o - \begin{bmatrix} \cos\psi_k & \sin\psi_k & 0 \\ -sin\psi_k & \cos\psi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k - x_{k-1} \\ y_k - y_{k-1} \\ \psi_k - \psi_{k-1} \end{bmatrix} , \tag{4.18}$$

where $\mathbf{m}_o = [\Delta x,\ \Delta y,\ \Delta\psi]^\top$ is the odometry estimation provided by the vehicle odometry module.

By assuming no-side-slip conditions (low-speed driving on dry roads), the following constraint can be formed based on the following kinematics-based two-dimensional vehicle odometry model:

$$\begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix} - \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \psi_{k-1} \end{bmatrix} - \begin{bmatrix} \cos\psi_k & -\sin\psi_k & 0 \\ \sin\psi_k & \cos\psi_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_k \\ 0 \\ r_k \end{bmatrix} \Delta t < \begin{bmatrix} \gamma_x \\ \gamma_y \\ \gamma_\psi \end{bmatrix}, \tag{4.19}$$

where u_k and r_k the vehicle's longitudinal and yaw rate, respectively, and $[\gamma_x, \gamma_y, \gamma_\psi]^\top$ is the upper bound threshold for the constraint.

4.2.7 Uncertainty-Aware Fusion

Based on the residual functions derived in Section 4.2, the localization problem is formulated as a nonlinear least-square optimization problem with the following form:

$$\begin{aligned} x_k^\star, y_k^\star, \psi_k^\star = \operatorname*{argmin}_{x_k, y_k, \psi_k} & \left(\|r_c\|_{\Omega_c} + \|r_l\|_{\Omega_l} + \|\mathbf{r}_o\|_{\boldsymbol{\Omega}_o} + \sum_{n=1}^{n_\pi} \|\mathbf{r}_{\pi,n}\|_{\boldsymbol{\Omega}_{\pi,n}} + \sum_{n=1}^{n_p} \|\mathbf{r}_{p,n}\|_{\boldsymbol{\Omega}_{p,n}} \right) \\ s.t. \quad & |\psi_k - \psi_{k-1} - r_k \Delta t| < \gamma_\psi \\ & |x_k - x_{k-1} - u_k \Delta t \cos\psi_k| < \gamma_x \\ & |y_k - y_{k-1} - u_k \Delta t \sin\psi_k| < \gamma_y \ , \end{aligned} \tag{4.20}$$

where γ_ψ, γ_x, and γ_y are the bounds that limit the maximum translation and rotation of the vehicle during a single time step based on the vehicle odometry model to prevent any unfeasible large jumps in self-localization. Since the vehicle self-location states in (4.20) are based on the vehicle frame $\{V\}$ while Lidar-based measurements are expressed in the $\{\mathrm{L}\}$ frame, there is a need to find the extrinsic calibration ${}^{L}\mathbf{T}_V$ to that transform the states into the Lidar frame. The procedure of the extrinsic calibration based on the vector HD map is presented in Section 4.3.

Each residue in (4.20) has a corresponding information matrix ($\mathbf{\Omega}$) that can be a function of the state. Therefore, the least-square problem contains heteroskedastic uncertainties. By transforming residuals based on the uncertainties, the problem becomes the following ordinary Least Square:

$$x_k^\star, y_k^\star, \psi_k^\star = \underset{x_k, y_k, \psi_k}{\operatorname{argmin}} \|\bar{\mathbf{r}}\| \ , \tag{4.21}$$

where $\bar{\mathbf{r}} = \bar{\Omega}^{0.5}\mathbf{r}$ is the transformed residual vector based on the information matrix:

$$\bar{\mathbf{\Omega}} = \operatorname{diag}(\Omega_c, \Omega_l, \mathbf{\Omega}_o, \mathbf{\Omega}_{\pi,1}, \mathbf{\Omega}_{\pi,2}, ..., , \mathbf{\Omega}_{\pi,n_\pi}, \mathbf{\Omega}_{p,1}, \mathbf{\Omega}_{p,2}, ..., \mathbf{\Omega}_{p,n_p}) \ , \tag{4.22}$$

where $\mathbf{\Omega}_\pi$ is the associated uncertainty of the plane residue and $\mathbf{\Omega}_p$ is the associated uncertainty of the pole residue. Finally, the residual vector $\bar{\mathbf{r}}$ is defined as:

$$\bar{\mathbf{r}} \triangleq \begin{bmatrix} r_c^\top & r_l^\top & \mathbf{r}_o^\top & \mathbf{r}_{\pi,1}^\top & \mathbf{r}_{\pi,2}^\top & \cdots & \mathbf{r}_{\pi,n_\pi}^\top & \mathbf{r}_{p,1}^\top & \mathbf{r}_{p,2}^\top & \cdots & \mathbf{r}_{p,n_p}^\top \end{bmatrix}^\top \ . \tag{4.23}$$

4.3 Map-Based Lidar Extrinsic Calibration

Accurate fusion of landmark measurements in the Lidar frame requires an accurate Lidar-to-vehicle extrinsic calibration, the transformation between the Lidar frame and the vehicle frame (${}^{R}\mathbf{T}_V$). It is often possible to obtain a rough initial Lidar-to-vehicle transformation by doing manual measurements. Therefore, the whole Lidar-to-vehicle transformation can be expressed as:

$${}^{V}\mathbf{T}_L = (\Delta\mathbf{T})\left({}^{V}\mathbf{T}_L'\right) \ , \tag{4.24}$$

where ${}^{V}\mathbf{T}'_{L}$ is the rough initial transformation that is known at prior and $\Delta\mathbf{T}$ is the transformation deviation that is unknown and hence to be estimated. The transformation deviation $\Delta\mathbf{T}$ can be expressed as:

$$\Delta\mathbf{T} = \begin{bmatrix} \cos\Delta\psi & -\sin\Delta\psi & 0 & \Delta t_x \\ \sin\Delta\psi & \cos\Delta\psi & 0 & \Delta t_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.25}$$

where $\Delta\psi$ is the deviation in the yaw angle, and Δt_x and Δt_y are the deviation in the lateral and longitudinal lever arms, respectively.

The approach here is to utilize the information provided by the HD vector map to estimate the deviations. The deviations are estimated by minimizing the error between the location of light poles in the Lidar frame (${}^{L}\mathbf{p}$) and the expected location according to the HD vector map (${}^{R}\mathbf{p} = [p_x, p_y, 0, 1]^{\top}$) (Fig. 4.1). The expected location of a light pole expressed in $\{R\}$ given by the HD vector map is as follows:

$${}^{R}\mathbf{p} = ({}^{R}\mathbf{T}_{V})({}^{V}\mathbf{T}_{L})({}^{L}\mathbf{p}) \ , \tag{4.26}$$

where ${}^{R}\mathbf{T}_{V}$ is the vehicle's pose in $\{R\}$ given by GNSS. Accordingly, the residual function of the minimization problem can be formed as follows:

$$\mathbf{r}\left(\Delta t_x, \Delta t_y, \Delta\psi\right) = \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}^{\top} \left({}^{R}\mathbf{T}_{V}^{-1}\left({}^{R}\mathbf{p}\right) - \begin{bmatrix} cos(\Delta\psi) & -sin(\Delta\psi) & 0 & \Delta t_x \\ sin(\Delta\psi) & cos(\Delta\psi) & 0 & \Delta t_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^{V}\mathbf{T}_{L} \begin{bmatrix} p_x \\ p_y \\ 0 \\ 1 \end{bmatrix} \right) . \tag{4.27}$$

Then, the following nonlinear optimization problem can be defined for n_p light poles:

$$\Delta t_x^{\star}, \Delta t_y^{\star}, \Delta\psi^{\star} = \underset{\Delta t_x, \Delta t_y, \Delta\psi}{\operatorname{argmin}} \frac{1}{2} \sum_{n=1}^{n_p} \|\mathbf{r}_n\|_{\mathbf{\Omega}_{p,n}} \ . \tag{4.28}$$

By solving the above nonlinear least square problem, the calibration between the Lidar and GNSS frames can be obtained.

4.4 Uncertainty Quantification

According to the optimization problem in (4.20) the self-localization problem is formed as a least-squared problem that contains the residuals with the information matrices defined in (4.22). It is crucial to quantify the information matrices associated with each observation of landmarks so that the solution to the localization problem becomes optimal.

To account for the uncertainty involved in the measurements, the residual analysis is performed. The uncertainty associated with pole detection, plane detection, and GNSS measurements is explained in the following. An outlier rejection rule is also developed for pole detection by analyzing the number of falling points.

4.4.1 Pole Detection Uncertainty

Landmark detection involves fitting a geometrical model over Lidar clustered points that may produce uncertain outputs. It is important to consider the effect of landmark uncertainty on the localization solution by identifying the uncertainty characteristics of landmark detection. For that reason, the dependence of pole residuals on the detected range of the pole is analyzed. Figure 4.6 represents bearing (ϕ) and range (r) measurement of a pole. As the vehicle goes along a nearby pole, it observes the pole at various bearing angles and ranges. Given the accurate pose of the vehicle by a precise localization module, the residual of pole detection can be calculated based on the position of the pole in the HD vector map. Accordingly, the residue of range measurement of a pole in the Lidar frame (${}^{L}\hat{\hat{\mathbf{p}}}$) given its ground truth model in the reference frame (${}^{R}\hat{\hat{\mathbf{p}}}$) is:

$$\widetilde{r}_r = \left\| \mathbf{T}\left(\psi_{GT}, x_{kGT}, y_{kGT}\right)^{-1}\left({}^{R}\hat{\hat{\mathbf{p}}}\right) \right\| - \left\| {}^{L}\hat{\hat{\mathbf{p}}} \right\| \ , \tag{4.29}$$

where ψ_{GT}, x_{kGT}, and y_{kGT} are the ground truth location of the vehicle.

For residual analysis, all the observations of poles that exist in the environment are collected at different ranges r. Then, a k-nearest neighbor (k-NN) approach is used to estimate the dependence of mean and variance of residuals to the range using Euclidean distance:

$$\begin{aligned} \hat{\widetilde{r}}_r = \mu\left(\widetilde{r}_r\right) = average\left(\left\{\widetilde{r}_{r'} | r' \epsilon knn(r)\right\}\right) \\ \hat{\sigma}_r^2 = \sigma^2\left(\widetilde{r}_r\right) = \sum_{r' \epsilon knn(r)} \frac{\left(\widetilde{r}_{r'} - \mu_r\right)^2}{k-1} \ . \end{aligned} \tag{4.30}$$

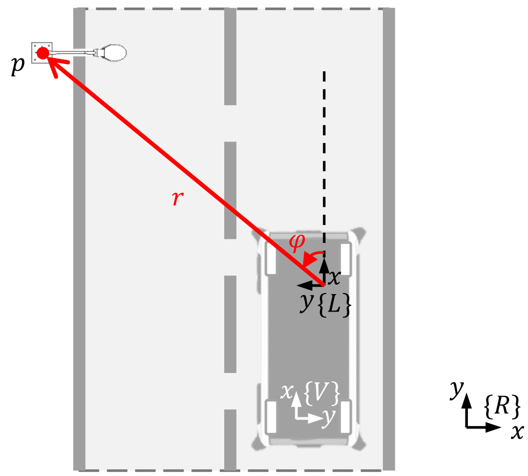


Figure 4.6: Range and bearing measurements for pole detection. Different levels of uncertainty is associated with pole detection as the vehicle observes the pole at various bearing angles and ranges

To fix heteroscedasticity in pole detection uncertainty, the Generalized Least Square (GLS) is used to perform weighted regression. This type of regression assigns weight to each data point based on the variance of its fitted value. To remove the effect of heteroskedasticity, the following additive uncertainty model is used:

$$\widehat{r}_r = r_r + \epsilon_r, \quad \epsilon_r \sim \mathcal{N}\left(\hat{\tilde{r}}_r, \hat{\sigma}_r^2\right) \ . \tag{4.31}$$

Using the same approach as (4.30), the k-NN is used to model the characteristics of the uncertainty in bearing measurements:

$$\begin{aligned} \hat{\tilde{r}}_\varphi = \mu\left(\widetilde{\varphi}_r\right) = average\left(\left\{\widetilde{\varphi}_{r'} | r' \epsilon knn(r)\right\}\right) \\ \hat{\sigma}_\varphi^2 = \sigma^2\left(\widetilde{\varphi}_r\right) = \sum_{r' \epsilon knn(r)} \frac{\left(\widetilde{\varphi}_{r'} - \mu_\varphi\right)^2}{k-1} \ . \end{aligned} \tag{4.32}$$

Accordingly, the additive uncertainty in the bearing measurements is as follows:

$$\widehat{r}_\varphi = r_\varphi + \epsilon_\varphi, \quad \epsilon_\varphi \sim \mathcal{N}\left(\hat{\widetilde{r}}_\varphi, \hat{\sigma}^2_\varphi\right) \ . \tag{4.33}$$

4.4.2 Pole Detection Outlier Rejection by Number of Falling Points Test

The analysis of the number of falling points onto a light pole allows filtering outliers from pole detection. For instance, pole detection can be erroneous due to partial occlusion of a pole with another object near the pole. Accordingly, the number of falling points onto poles as a function of distance to the pole is considered:

$$n_{fp} = f(r) \ , \tag{4.34}$$

where n_{fp} is the number of falling points onto a pole. A k-NN is used to estimate the function in (4.34) as a regression over the history of all observed poles:

$$\begin{aligned} \hat{n}_{fp} &= f(r) = average\left(\left\{f(r')|r^{'}\epsilon knn(r)\right\}\right) \\ \hat{\sigma}_{fp} &= Se\left(r\right) = \sqrt{\sum_{r^{'}\epsilon knn(r)} \frac{\left(f(r') - f(r)\right)^2}{k-1}} \quad . \end{aligned} \tag{4.35}$$

For outlier rejection, a confidence band around is defined that takes into account the uncertainty in pole detection due to possible variation in the geometry of the poles (e.g. height and cross-section). Any detected pole which lies outside of the following confidence band is considered an outlier:

$$\hat{n}_{fp} \pm t_{p,\nu}\hat{\sigma}_{fp} \ , \tag{4.36}$$

where $t_{n,p}$ is the t value at the probability values in p using the corresponding degrees of freedom in ν.

4.4.3 Plane Detection Uncertainty

To account for the effect of plane detection uncertainty on the localization problem, the plane residuals' dependence on the number of falling points is analyzed.

Based on distance-angle parametrization in (4.4), the uncertainty of plane distance detection is quantified through k-NN approach:

$$\begin{aligned} \hat{r}_d = f\left(n_{fp}\right) = average\left(\left\{f\left(n'\right)|n' \epsilon knn(n_{fp})\right\}\right) \\ \hat{\sigma}_d^2 = \sigma^2\left(n_{fp}\right) = \sum_{n' \epsilon knn(n_{fp})} \frac{\left(f\left(n'\right) - f\left(n_{fp}\right)\right)^2}{k-1}\ . \end{aligned} \tag{4.37}$$

Accordingly, the additive uncertainty in the plane distance measurement is as follows:

$$\widehat{r}_d = r_d + \epsilon_d, \quad \epsilon_d \sim \mathcal{N}\left(\hat{r}_d, \hat{\sigma}_d^2\right)\ . \tag{4.38}$$

4.4.4 GNSS Noise Characteristics

The GNSS measures the global position with some uncertainties due to multiple factors. Different phenomena such as the receiver noise (Antenna design, Analog-to-digital conversion, etc.), multipath, atmospheric effects, clock errors, relativity, etc. can cause the GNSS measurements to not be perfect. However, it is important to study the characteristics of the uncertainty of the GNSS measurements to make a realistic measurement model.

According to [83], in short periods such as minutes or hours, it is reasonable to assume that the GNSS noise is a random Gaussian Noise with zero mean. Other noise types such as Flicker and random walk noise appear at longer periods. In this project, since the GNSS signals are used in short time intervals, the effect of noise on higher time intervals can be neglected. Furthermore, the GNSS alignment can be reinitialized multiple times to remove any misalignment due to the long-time noise characteristics of GNSS.

4.4.5 Situation- and Uncertainty-Aware Attention Mechanism

A situation- and uncertainty-aware attention mechanism is used to properly rely on "suitable" landmarks used for the self-localization task. At any drivable location within the known environment, a subset of all mapped landmarks is observable with different levels of uncertainty. A situation- and uncertainty-aware attention mechanism proactively select the landmarks that are more likely observable and less uncertain.

The system is situation-aware since it knows a prior approximate self-location so that it can blindly approximate where the landmarks are going to be located (where to attend?) before the actual observation of them. Therefore, processing the whole online measurement data is not necessary. On the other hand, it is uncertainty-aware since the level of uncertainty involved in the estimation of a landmark's location is predicted based on the observations of the same landmarks in the past.

The mechanism contains a hard and a soft attention mechanism. A hard attention mechanism is used to select a subset of observable landmarks. It is modeled as a binary mapping from the domain of s-coordinates, S, to the set of all mapped landmarks, M, that selects the set of observable landmarks, M_o,

$$\begin{aligned} A_h &: S \to M \\ A_h(s) &= M_o \ . \end{aligned} \tag{4.39}$$

Additionally, a soft attention mechanism is modeled as a function from the domain of observable landmarks M_o to $\mathbb{R}^+$ that provides the level of uncertainty associated with the observable landmarks:

$$\begin{aligned} A_s &: M_o \to \mathbb{R}^+ \\ A_s(l) &= \begin{cases} U_\pi(l) & \text{if } l \in \text{planes} \\ U_p(l) & \text{if } l \in \text{poles} \\ U_c(l) & \text{if } l \in \text{curbs} \end{cases} \ , \end{aligned} \tag{4.40}$$

where U_π, U_p, and U_c are the uncertainty models presented in Section 4.4.

4.5 Summary

In this chapter, a new situation- and uncertainty-aware self-localization system is developed that uses an HD vector map along with landmarks that are more reliable at every drivable location within the known environment. Some Uncertainty models are developed to quantify the uncertainty of landmarks that are detected under different conditions. Additionally, a new map-based calibration algorithm is developed that refines the extrinsic calibration of a Lidar sensor to the vehicle frame.

Chapter 5

Evaluation and Experimental Studies

5.1 Introduction

In this chapter, the evaluation and experimental studies are provided for the vehicle self-localization that is developed in Chapter 4. The system is deployed on WATonoBus while operating at the University of Waterloo Ring Road.

The rest of this chapter is organized as follows. Section 5.2 presents the experimental setup on which the algorithm is deployed. In Section 5.3, the procedure and the results of obtaining a light planimetric map of landmarks of the Ring Road are presented. Section 5.4 provides the results of Lidar extrinsic calibration using the developed algorithm in the previous chapter. In Section 5.5, the results obtained from the uncertainty quantification of poles and planes in the Ring Road are presented and discussed. Section 5.6 presents the metrics and the baseline algorithm used for performance evaluation of the developed self-localization system. In 5.7, the localization result of the developed landmark-based self-localization system is presented and discussed. Finally, Section 5.8 provides a summary.

5.2 Experimental Setup

The developed self-localization algorithm is implemented as a ROS application, named HD-LOC, that includes a set of nodes that communicate in the ROS environment (Figure 5.1). Accordingly, HD-LOC consists of two main ROS nodes, front-end, and back-end. The front-end node is responsible for processing the Lidar raw measurement data by clustering the points and extracting geometrical models of expected planes and poles. The output of the front-end is the detection of the light poles and planes over the Lidar point cloud. The back-end node finds the optimized self-location based on the front-end observations and the location of the landmarks in the reference frame which is provided by the attention mechanism and an HD vector map.

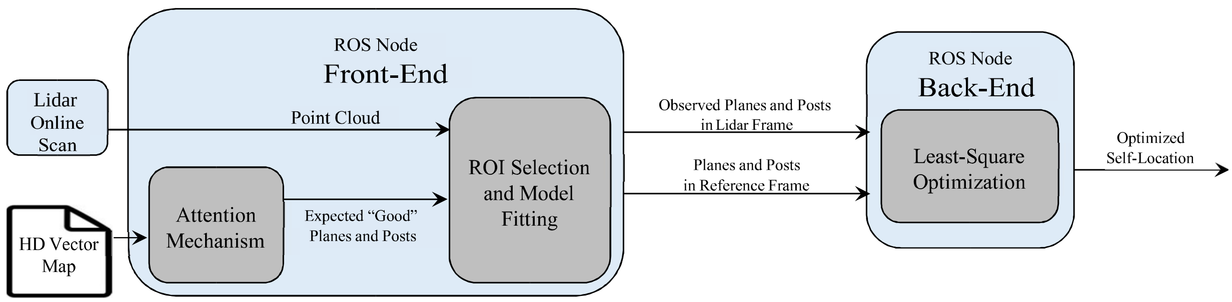


Figure 5.1: System diagram of HD-LOC. It contains two ROS nodes that communicate within the ROS environments. The ROIs are formed around the expected location of landmarks according to the HD Vector map and *a priori* self-localization of the ego-vehicle. Landmark model fitting is performed over ROI clusters. Finally, the landmark models in the Lidar frame and in the reference frame are used in the least-square problem to estimate the self-location of the vehicle.

To evaluate the performance of the proposed algorithm, the self-localization of WATonoBus is performed throughout the University of Waterloo Ring Road. WATonoBus is an Autonomous shuttle bus developed in the Mechatronic Vehicle Systems (MVS) Lab at the University of Waterloo [84] (Figure 5.2). It is equipped with multiple hardware-synchronized sensors, including a 32-channel RoboSense 360° Lidar at 10 Hz, Applanix POS LVX GNNS-INS at 50 Hz (used as the Ground Truth), and multiple cameras at 10 Hz. The processing unit includes a Simply NUC Ruby R8 PC. Figure 5.3 shows a sample of scans captured by the short-range and long-range Lidars.

5.3 Planimetric Landmark Mapping

The objective of mapping is to build a set of landmarks suitable for self-localization tasks. Planimetric maps of landmarks of interest are typically available through aerial images

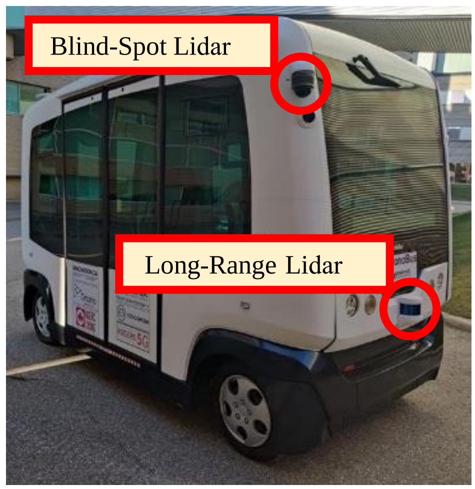


Figure 5.2: The WATonoBus platform is used for experimental studies of the developed self-localization system. The vehicle is equipped with a long-range 32-beam Lidar with a vertical Field of View (FOV) of 40° and horizontal FOV of 180° with up to 200 m range and ±3 cm accuracy. A short-range 32-beam blind spot Lidar is used with 360° horizontal FOV, 90° vertical FOV, and a range of 30m with up to ±3 cm typical range accuracy. The short-range Lidar is used for scanning nearby curbs. The long-range Lidar scans building planes and poles. The vehicle is equipped with an RTK-GNSS-INS with centimeter-level accuracy that provides the ground truth information of the self-localization.

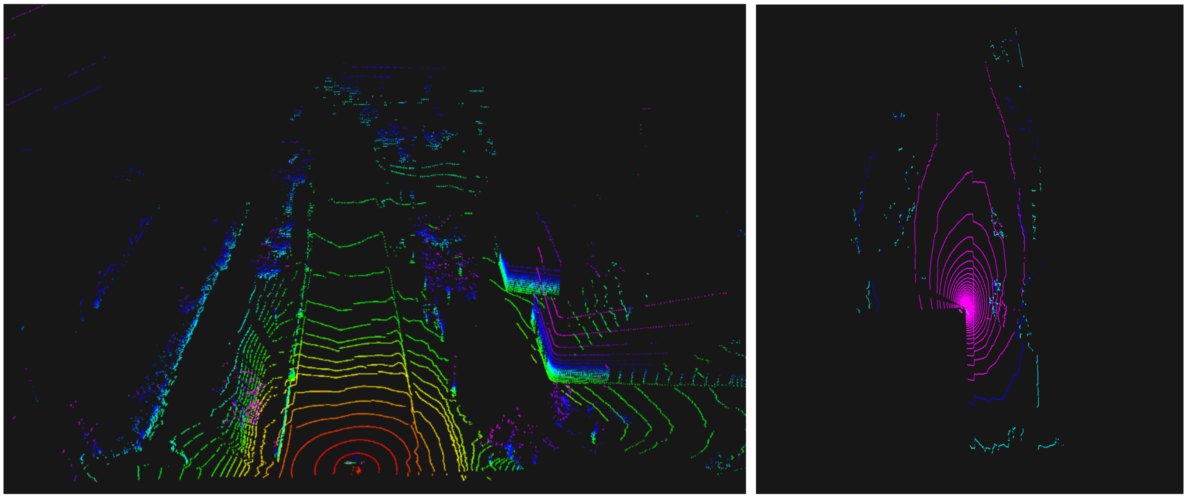


Figure 5.3: The scan of the long-range Lidar (left) and short-range blind spot Lidar (right). A Long-range Lidar is used for scanning bigger landmarks, including building planes and light poles, and a short-range Lidar is used to scan nearby curbs.

and open-sourced maps for urban areas. However, their accuracy is not sufficient for the self-localization purpose. Hence, surveying is usually performed to construct a precise HD vector map.

Given that a precise HD vector map is not available for the Ring Road, a mobile ground surveying approach is used to construct the map by obtaining geometric and georeferenced information about landmarks. This process involves Data Acquisition and Registration, Landmark Extraction, and Geometric Modelling.

The HD vector map of the Ring Road is constructed by WATonoBus acting as a surveying vehicle equipped with a Lidar sensor and precise RTK-GNSS-INS measurements.

5.3.1 Data Acquisition and Registration

The surveying vehicle equipped navigates in the environment and the point cloud generated from the scans is collected. All the point clouds are registered into a common local reference frame in the map that forms a single point cloud. The whole point cloud is then divided into high-elevation points and ground-level points based on thresholding over the Lidar height to the ground. Figure 5.4 illustrates the point cloud that is captured at the University of Waterloo's Ring Road with the ground removed. The point cloud was captured when the

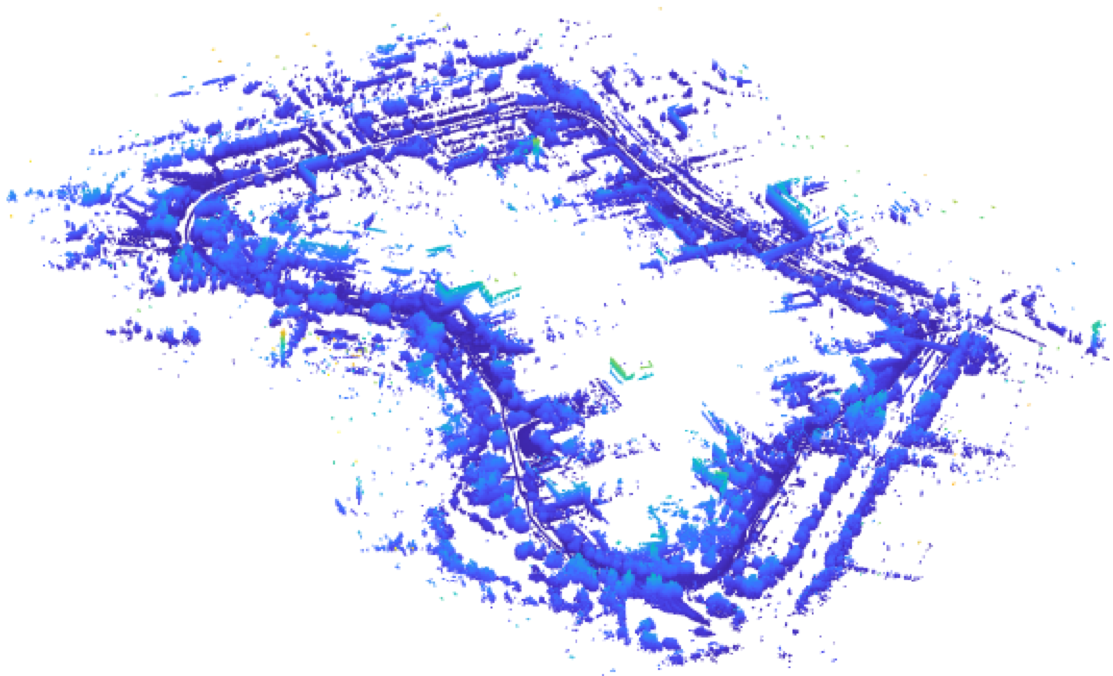


Figure 5.4: Result of Point cloud collection captured at the Ring Road at the University of Waterloo Campus. It contains scans of Lidar that are registered into a fixed local reference frame. The points that correspond to the ground are removed based on thresholding.

campus was not crowded to have limited occlusion due to dynamic objects such as cars and pedestrians.

As the initial step to extract the landmarks, the approximate location of light poles and building facades are extracted using available aerial imagery and manual inspection of the point cloud (Figure 5.5). The next step of HD vector mapping is to obtain the ground truth geometrical model of the landmarks that is described in the following sections.

5.3.2 Landmark Extraction and Geometrical Modelling

As the surveying vehicle goes along the road, nearby landmarks are observed from different angles of view and distances (Figure 5.6). The landmarks are extracted from the Lidar scan using the robust clustering approach developed in Sections 4.2.5 and 4.2.6.

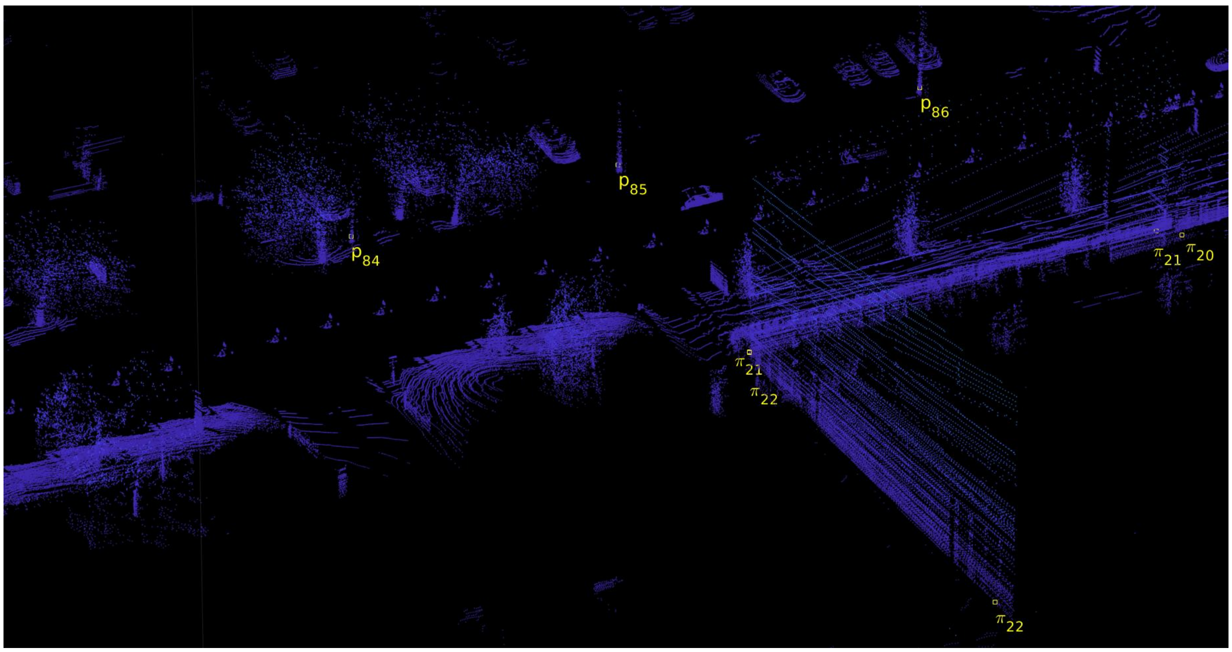


Figure 5.5: Manual selection and labeling of suitable landmarks. A landmark is considered suitable if it is observable from a portion of the road. Poles are represented as a single point and planes are represented by their two endpoints.

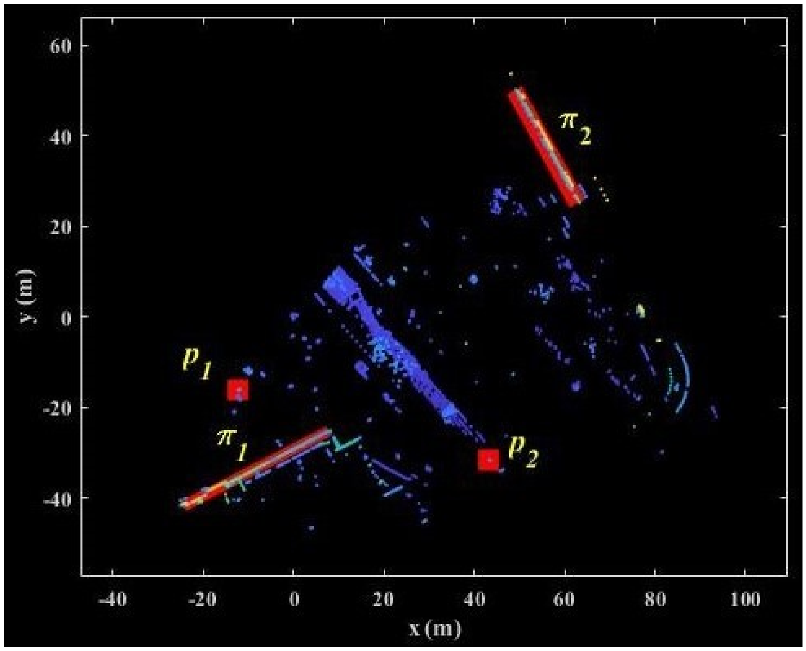


Figure 5.6: Bird-eye-view of a Lidar scan (denoted in blue) and detected light poles p_1 and p_2 and building planes π_1 and π_2 (denoted in red)

According to obtained results of uncertainty quantification of landmarks in Section 5.5, the most reliable observation of landmarks is stored in the HD vector map. For a pole, the observation that involves firstly the maximum number of falling points and secondly the minimum detection distance is treated as the most precise observation of the pole and hence stored in the HD Vector Map.

For extracting the location of building planes, the geometrical model is fitted over the aggregation of points extracted from ROI when observed at different times. Using the aggregation of point clouds has some benefits when compared to using a single observation of a landmark. First, using the aggregation of point clouds maximizes the extent of a mapped plane while a single observation of a plane may partially cover it. Additionally, the final fitted model over the aggregation of point clouds takes an average over individual observations from different views; therefore, it is more robust to the presence of outliers in the Lidar data or inaccuracies in some of the observation locations.

For extracting curb and lane markings, the control points measured by the curb and lane detection module are stored in a database. The curb detection module provides the curb location near the vehicle in the vehicle frame (Figure 5.7). A line robustly fits into

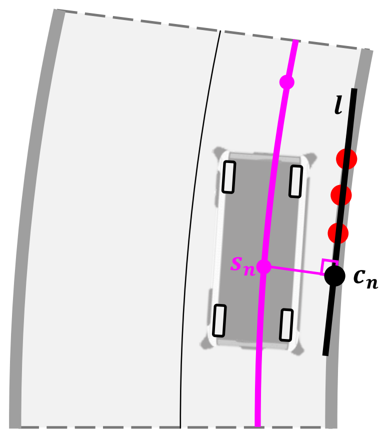


Figure 5.7: The process of collecting curb control points. A curb detection module provides the nearby curb points (denoted in red). A curb control point ($\mathbf{c}_n$) is obtained by projecting the nearest s-coordinate ($\mathbf{s}_n$) on the fitted curb line (l).

the curb points expressed in the vehicle frame:

$$l: \; y = b_1 x + b_2. \tag{5.1}$$

The projection of the nearest s-coordinate (s_x, s_y) onto the fitted line l is obtained:

$$\mathbf{c}_n = \begin{bmatrix} \frac{x_s \; - \; b_1 b_2 \; + \; b_2 y_s}{b_2^2 + \; 1} \\ \frac{b_1 + b_2 x_s \quad + \; b_2^2 y_s}{b_2^2 + \; 1} \end{bmatrix} \quad . \tag{5.2}$$

Given the position of the vehicle in the global reference frame, the location of the curb control point is transformed into the reference frame and stored. Upon multiple observations of a control point at different times, the mean location is used to account for any possible outlier in the observation of a curb/lane due to partial occlusion or algorithm misdetection.

5.3.3 Mapping Results

Figure 5.8 shows the result of HD Vector mapping around the University of Waterloo Ring Road. The developed HD vector map includes 107 light poles, 177 building planes, and curb control points along the Ring Road which is 2.63 km long.

5.4 Result of Map-Based Lidar Extrinsic Calibration

As described in Section 4.3, a map-based algorithm for Lidar extrinsic calibration is developed. The approach is to compare the poles in the HD vector map with the actual observation of the poles to estimate the calibration deviation and refine the calibration.

To validate the developed calibration algorithm, a single scan of Lidar is used in which 5 poles are observable. Additionally, an imprecise Lidar calibration is used to find the expected location of poles in the Lidar frame which is obtained by transforming the poles' location from the HD Vector map to the Lidar frame. On the other hand, the geometric model of the light poles is extracted from the Lidar scan.

Then, the developed Lidar calibration algorithm is used to find the precise calibration of Lidar. Figure 5.9 illustrates the expected location of light poles in the Lidar frame before and after refining the calibration. According to this figure, the result shows the effectiveness of the developed calibration algorithm qualitatively by illustrating the conformity of the expected location and actual location of poles after refinement.

5.5 Uncertainty Quantification Results

This section provides the results for applying the uncertainty quantification algorithms developed in Section 4.4 on the Ring Road dataset.

5.5.1 Pole Detection Uncertainty

Figure 5.11a illustrates the residuals poles' range detections in different ranges all over the Ring Road. Accordingly, a k-nearest neighbor (k-NN) approach is used to estimate the dependence of mean and variance of residuals to the range using $k = 2000$ and Euclidean distance. Figure 5.11b shows the dependence of residual standard deviation on the range

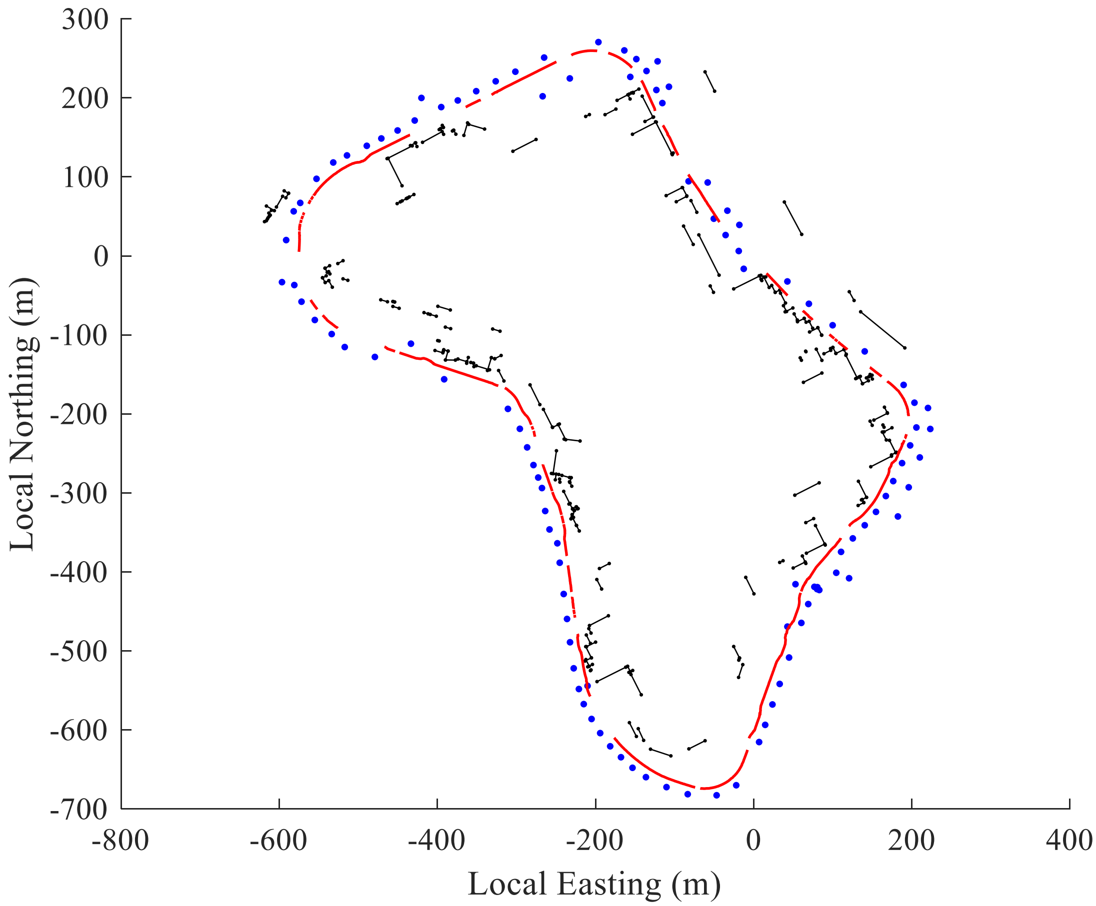
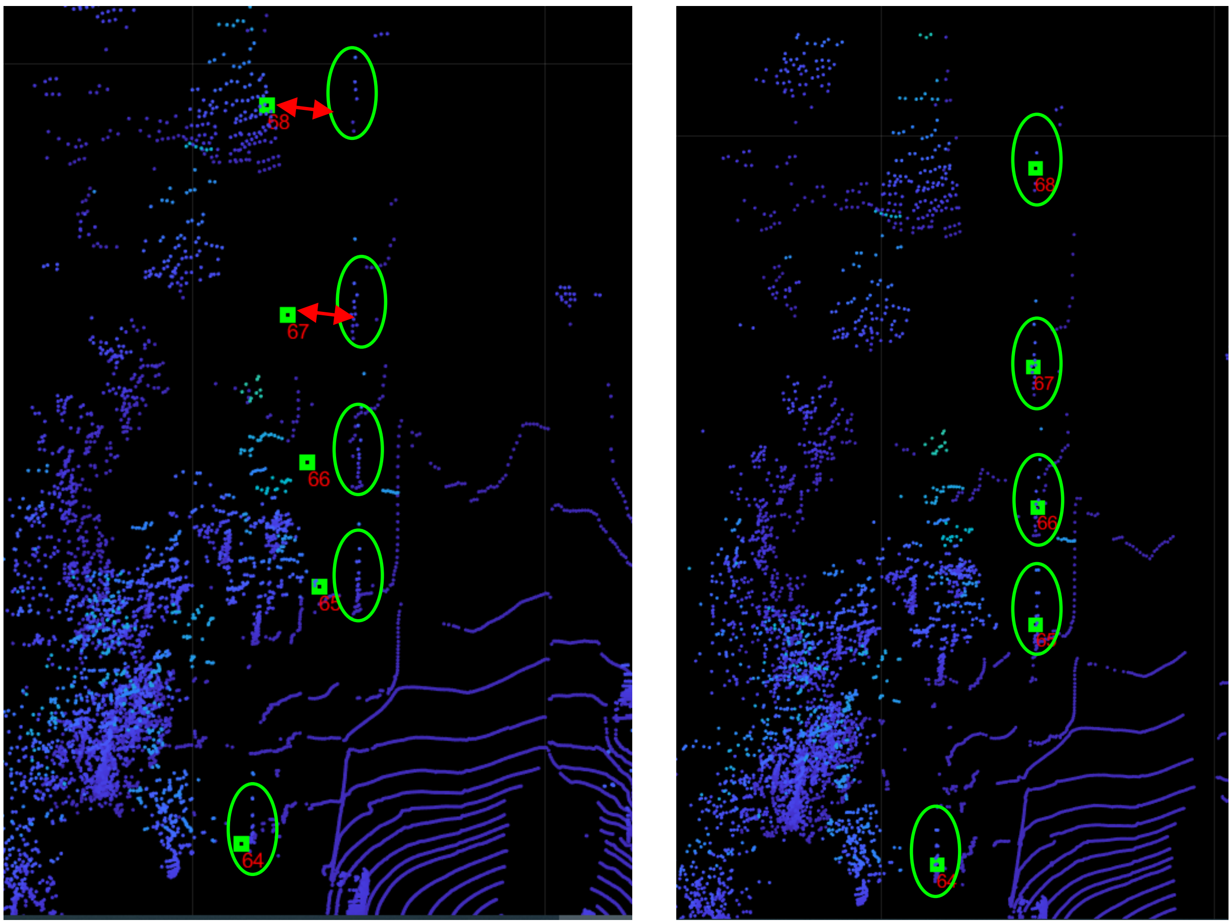


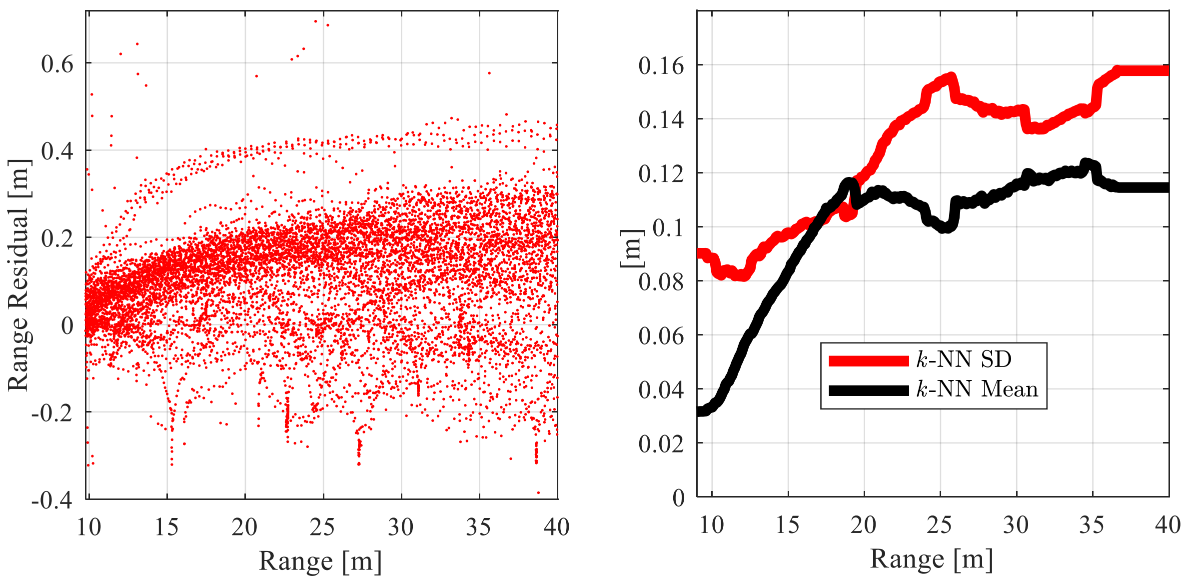
Figure 5.8: The result of mapping the landmarks at the University of Waterloo Ring Road, including 107 light poles (blue), 177 building planes (black), and curb points (red) with 2.63 km of length. The location of landmarks is expressed in the navigation frame (The local Easting axis (x) is perpendicular to gravity, perpendicular to the local Northing axis and is in the east direction. The local Northing axis (y) is perpendicular to the gravity vector and in the direction of the north pole along the earth's surface. The up axis (z) is co-axial with the gravity vector and positive in the up direction.)



b) Before Refinement of Extrinsic Calibration

b) After Refinement of Extrinsic Calibration

Figure 5.9: By minimizing the distance between the observed light poles (ovals) and their expected location acquired from the HD vector map (squares), the Lidar to Vehicle extrinsic calibration is refined. The conformity of the actual and expected location of the poles confirms the effectiveness of the developed Lidar calibration algorithm.



(a) (b)

Figure 5.10: Uncertainty quantification of pole detection k-NN regression of residuals (a) the bearing residuals for pole detection (b) the mean and standard deviation of bearing residuals for pole detection. The decreasing standard deviation shows the heteroskedasticity of uncertainty in estimating the pole bearing.

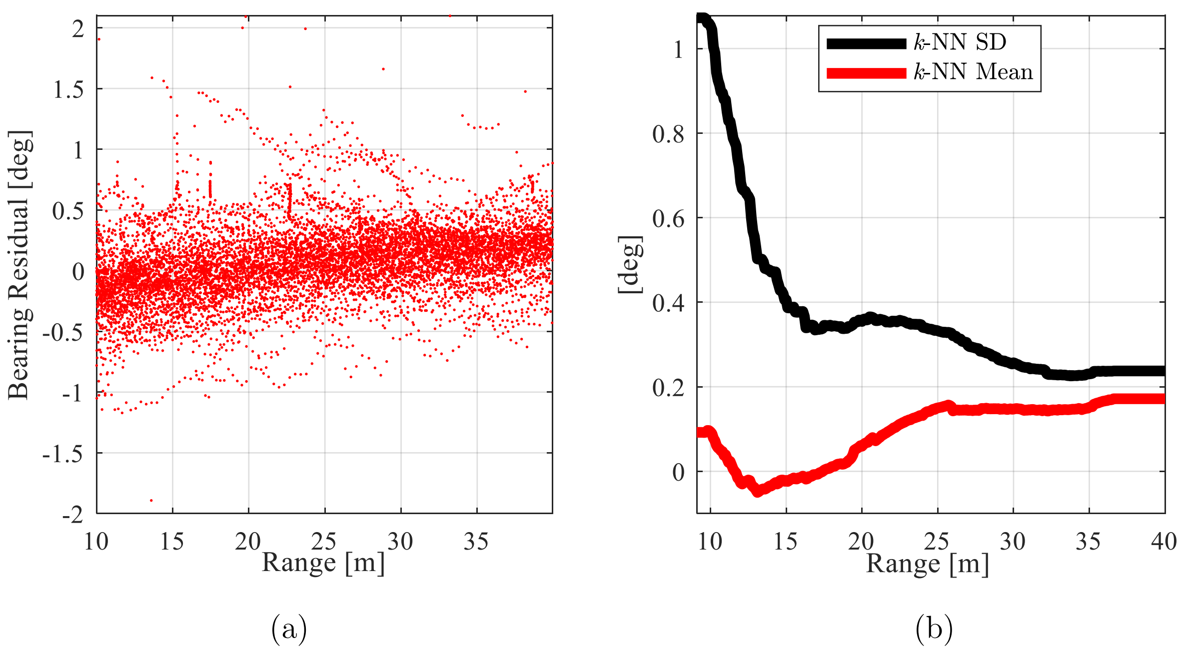


Figure 5.11: Uncertainty quantification of pole detection k-NN regression of residuals (a) the range residuals for pole detection (b) the mean and standard deviation of range residuals for pole detection. The increasing standard deviation shows the heteroskedasticity of uncertainty in estimating the pole range.

Non-constant variability of residuals shows heteroscedasticity in pole detection. It is because a pole's predicted location is modeled by selecting the median of all the points falling onto the pole from different angles as the center of the pole. However, when observed from farther ranges, Lidar beams that arrive at the location of the pole become sparser and hence may hit the pole not necessarily in the middle of its cylindrical body which may cause some error in the observed location of the pole. Additionally, the zero-radius assumption of the pole body results in additional errors in calculating the range. According to Figure 5.11b, the residuals mean is also increasing. It is because some small heading errors in the extrinsic calibration of Lidar to GNSS can be propagated proportionally to the range, which causes some epistemic uncertainty in pole detection.

Here, to account for the non-uniform uncertainty, one approach is to correct the geometrical model of poles to account for the pole's geometry. This approach seems reasonable by removing the root cause of varying uncertainty. However, it involves making the pole detection process complex and less efficient by complicating the geometrical model of landmarks.

Another approach for reducing the uncertainty is to obtain a very precise Lidar-GNSS calibration to remove the epistemic uncertainty and heteroskedasticity. However, obtaining and continuously maintaining a very accurate Lidar to GNSS calibration is a difficult task. This thesis provides a novel approach for Lidar-GNSS calibration. But, removing all the inaccuracies from the calibration is infeasible given the presence of measurement uncertainty due to the inherent inaccuracy of Lidar and GNSS measurements.

In this thesis, as described in (4.31), an alternative approach is used that considers the effect of total uncertainty in the regression problem to obtain a more accurate localization solution.

Figure 5.10a shows the dependence of bearing measurement residuals on the range. In contrast to range residuals, bearing detection of landmarks becomes more accurate in larger ranges. This is because small errors in detecting the pole's position result in larger bearing errors in shorter ranges.

Figure 5.12 shows the number of falling points onto a pole versus the distance from the pole. Accordingly, any detected pole that lies outside the 97.5% confidence band is considered an outlier.

5.5.2 Plane Detection Uncertainty

Figure 5.13 presents the results of residual analysis of plane distance detection. According to Figure 5.13a, the variation of the distance error over the number of falling points is

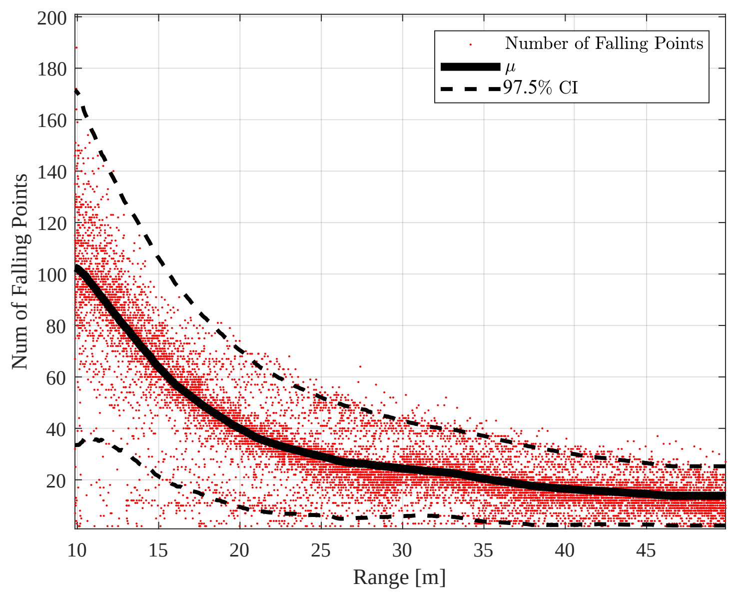


Figure 5.12: The result of pole outlier rejection based on regression over the number of falling points on the poles versus the range. The number of falling points is decreasing because when detecting a pole from farther away, a smaller subset of Lidar laser beams collides with the body of the pole. Any detected pole that is outside of a 97.5% confidence band is considered an outlier.

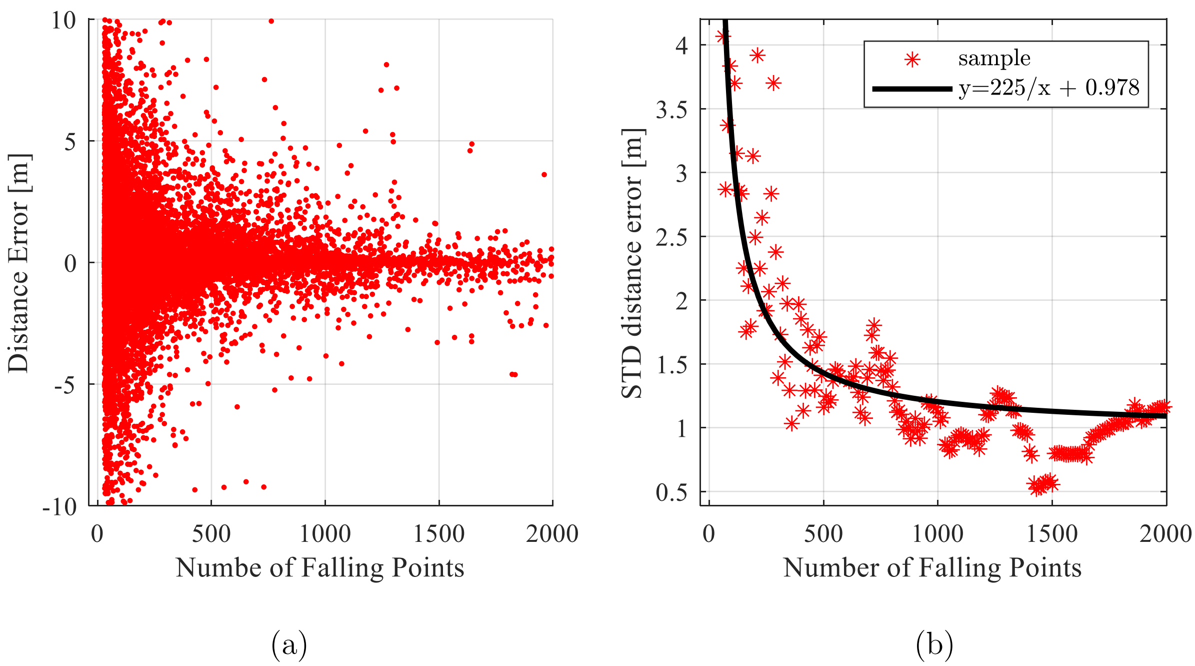


Figure 5.13: The results for (a) variation of plane distance error versus the number of points that fall on a plane and (b) the variation of the standard deviation of the distance error versus distance to the plane.

not uniform which shows the heteroskedasticity in the uncertainty of plane detection. An empirical relationship is derived using the output of variance estimation based on the k-NN method (Figure 5.13b):

$$\sigma_{\pi,d}(N) = 225/N \ + \ 0.978 \text{ [m]} \ , \tag{5.3}$$

where N is the number of points that fall on the plane. Accordingly, the error in plane detection exponentially increases as the number of falling points decreases. This is a reasonable conclusion since a plane that is fitted onto a fewer number of points tends to have higher uncertainty because the points are less likely to represent the overall geometry of the whole plane.

5.6 Real-time Performance

5.6.1 Metrics

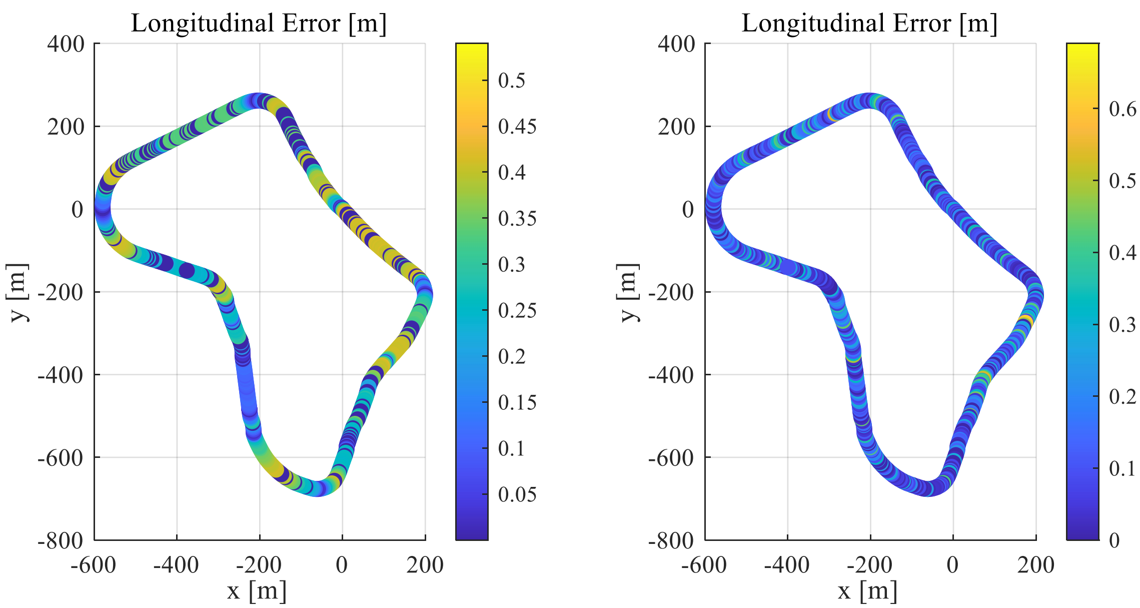
The self-localization error along the lateral direction and the longitudinal direction of the vehicle are the main metrics to evaluate the performance of the system. Dividing the whole localization error into the lateral and longitudinal components is very informative since an automated vehicle cannot afford a large lateral localization error, since it would result in going into the opposite lane whereas the large longitudinal error is less severe.

5.6.2 Baseline Algorithm

The self-localization performance of the system is compared with the results of NDT Map Matching in the same environment. The NDT Map Matching algorithm is a registration algorithm that uses standard optimization techniques applied to statistical models of 3D points to determine the most probable registration between two sets of point clouds. To increase the speed of the registration, both the map and the online scans of the Lidar are spatially down-sampled into 0.1m resolution.

5.7 Localization Results

Table 5.1 summarizes the self-localization results of NDT Map Matching and HD-LOC. Accordingly, HD-LOC has less localization accuracy compared to NDT Map Matching.



(a) NDT Map Matching (b) HD-LOC

Figure 5.14: Longitudinal error of self-localization over the Ring Road. Accordingly, HD-LOC estimates the self-localization of WATonoBus with good consistency over the entire Ring Road. On the other hand, the accuracy of NDT map matching deteriorates in some part of the Ring Road.

This is because NDT Map Matching preserves most of the geometrical details of landmarks features while HD-LOC approximates the geometry of the landmarks with simple geometrical models including planes and lines. Note that NDT Map Matching could lead to a weaker localization accuracy if tested over maps with large seasonal changes. On the other hand, since HD-LOC filters out unstructured objects that change may shape in time (e.g. vegetation), it can preserve its self-localization performance in different seasons.

Figure 5.15: The environment contains mostly temporarily parked vehicles while lacking enough longitudinal and lateral excitation. HD-LOC solely attends to the light poles (denoted in red) to localize the vehicle while filtering out the rest of the point cloud.

Figure 5.14(a) and (b) qualitatively represent the longitudinal localization error of NDT Map Matching and HD-LOC, respectively, over the entire University of Waterloo Ring Road. Accordingly, HD-LOC has a relatively more uniform performance all over the Ring Road compared to NDT Map Matching. In fact, some locations of the Ring Road contain a small number of good landmarks that makes it difficult to accurately self-localize. Figure 5.15 illustrates a location at the university ring road which contains mostly unreliable landmarks, including parked vehicles and vegetation, and lacks enough structured landmarks such as building walls. In this case, NDT Map Matching is outperformed by

Table 5.1: Quantitative comparison of NDT results with the HD-LOC

Methods	**Longitudinal Error (m)**			**Lateral Error (m)**			**Computational Time (ms)**
	Median	**95%**	**Max**	**Median**	**95%**	**Max**	**Range**
NDT Map Matching	0.24	0.41	0.54	0.15	0.39	0.61	100-1000
HD-LOC	0.26	0.64	0.70	0.26	0.63	0.70	<20

HD-LOC because of attending to all the landmarks regardless of their quality whereas HD-LOC solely attends to the light poles to localize the vehicle while filtering out the rest of the point cloud.

5.7.1 Runtime Analysis

The system is benchmarked on a CPU. A single step of inference using a CPU single-core takes less than 20 ms in total for HD-LOC. Therefore, it can run at 50 Hz, regardless of how much the initial guess of self-location is accurate. On the other hand, NDT Map Matching takes around 100 ms up to 1000 ms for a single inference, based on how the initial guess is accurate. Therefore, it can run at 1 Hz. Therefore, the developed HD-LOC is 50 times faster than NDT Map Matching. Since Map Matching algorithms deal with a trade-off between accuracy and speed according to the down-sampling resolution of the point cloud, they can be even slower once more accuracy is desirable.

Map Storage Analysis

The storage size of the HD vector map is compared with the point cloud map required for NDT Map Matching. The down-sampled point cloud map that is used for NDT Map Matching takes 11 MB per km of driving while the HD vector map takes 0.07 MB per km of driving, which is 0.6% of the Lidar point cloud map.

5.8 Summary

In this chapter, the evaluation and experimental results are provided for the developed self-localization system. The results showed that this technique improves processing efficiency by proactively attending to the useful portion of Lidar data inside an ROI around the landmark and filtering out the rest of the points. The developed localization system has been installed on WATonoBus at the campus of the University of Waterloo. The experimental results demonstrate comparable accuracy, superior computational efficiency, and exceptionally low storage needs compared to a map-matching-based self-localization algorithm. By using landmarks that are invariant to seasonal changes and knowing "where to look" proactively, robustness and computational efficiency are improved.

Chapter 6

Conclusion and Future Work

6.1 Conclusions

The main objective of this research was to design a real-time self-localization algorithm for vehicles. To make the estimation algorithm practical and implementable on autonomous vehicles, a design for estimating the self-localization was proposed using common sensors available in autonomous vehicles.

One of the main challenges for map-based vehicle self-localization systems is estimating short-term vehicle odometry while landmarks measurements are not reliable. To tackle this challenge, two algorithms were developed for estimating vehicle odometry: (a) a model-based vehicle odometry that fuses the measurement of a camera, a Lidar, an IMU, and a low-cost GNSS, and (b) an ML-based vehicle odometry system that fuses IMU and wheel encoders. The model-based odometry system was designed to improve the self-localization problem in general settings by imposing an extra constraint on the optimization problem whereas the ML-based odometry system was mainly designed to compensate for the intermittent self-localization losses.

Regarding the model-based odometry system, based on the experiments that were performed in real-world and simulated environments, the following conclusions can be reached; first, the model-based system has promising performance in estimating odometry, both in urban and highway driving scenarios. Additionally, the fact that it does not rely on wheel sensors makes it robust to weak observability conditions of slippery driving conditions. Additionally, through a tightly-coupled fusion of IMU and camera, it can properly account for IMU biases by incorporating them in the set of states. Moreover, according to the results, it was shown that the developed multi-modal odometry system is robust to the GNSS noise level and sampling frequency value. As a result, using a GNSS sensor in the odometry system does not contradict the main purpose of odometry which is to provide a short-term localization solution while the main localization system is not reliable.

On the other hand, for the developed ML-based odometry system the following conclusions can be made; first, the developed ML-based odometry system provides exceptional estimation accuracy, more than the developed multi-modal system. This is because it uses the recently collected historical data for providing the odometry solution while the localization is not reliable. On the other hand, the result of the input feature selection shows that yaw rate, wheel speed, lateral and longitudinal acceleration, and steering wheel angle are sufficient for accurate odometry estimation. Moreover, the result showed that the ML-based odometry system could compensate for unknown biases in the IMU measurements and produce reliable accurate odometry estimation.

One challenge for landmark-based localization systems has been the presence of varying uncertainty of the landmarks in different conditions. To address this challenge, a new situation- and uncertainty-aware efficient map-based self-localization technique was developed. It has been shown through experimental results that the developed system has comparable accuracy, superior computational efficiency, and exceptionally low storage needs compared to traditional map-matching-based self-localization algorithms.

Regarding the developed landmark-based self-localization system, the following conclusions can be made; first, based on the developed methodology for mapping, it was shown that a precise GNSS along with a Lidar could be used to construct an HD vector map for an unknown environment. It was shown that the developed landmark-based Lidar-GNSS extrinsic calibration could effectively refine the calibration by comparing the observation of landmarks with an HD map. On the other hand, some uncertainty models were developed that are used for relying on more reliable landmarks at every drivable location within the known environment. Based on the uncertainty quantification results, it was shown that as a pole is observed from farther distances, the uncertainty of the pole's range detection increases significantly while the uncertainty of the pole's bearing detection decreases significantly. It was also shown that the uncertainty of plane distance detection decreases significantly as the number of falling points onto a plane increases. Additionally, based on the results for the number of falling points onto poles, a meaningful pattern was observed and utilized for the development of a rejection mechanism for pole outliers. Finally, the localization results showed that the developed self-localization system improves processing efficiency by proactively attending to the useful portion of measurements and filtering out the rest.

However, the self-localization system developed in this study is not without limitations. One significant constraint is its suitability for use in environments where an adequate number of observable landmarks are available at all drivable locations. Therefore, the system does not apply to environments with a limited number of sparse landmarks. Additionally, certain landmarks may not be visible in certain instances, such as when occluded by nearby traffic objects, including large buses, which can reduce the number of observable landmarks. Consequently, it is necessary to have some redundancy in the number of landmarks in the environment to ensure the reliable operation of the system. Furthermore, the self-localization system's effectiveness is contingent on favorable weather conditions. In particular, it would not perform optimally in harsh weather conditions such as fog or rain. Lidar sensors, which are used in this system to detect landmarks, are not inherently capable of generating accurate point clouds in adverse weather conditions, leading to erroneous readings.

6.1.1 Future Works

Suggestions made in this section are for potential future works to enhance the accuracy of the developed self-localization system.

Uncertainty-aware multi-modal vehicle odometry: In Section 3.2, a model-based vehicle odometry algorithm was presented that uses multiple exteroceptive sensors, including cameras and Lidars. The performance of exteroceptive sensors depends on environmental conditions such as lighting, the level of detail of the features, and the presence of dynamic objects. A residual analysis approach can be utilized to quantify the uncertainty of the sensors' measurements. In the fusion problem, the uncertainty models would provide the reliability of each sensor modality that could improve the overall performance of the system. Deactivating unreliable sensor modalities can improve the efficiency of the system as well. Moreover, the design of the developed odometry systems can be modified such that it can fuse measurements from asynchronous sensors that are available on production vehicles. Finally, including other sensors such as radars can improve the flexibility of the model-based system to be used in different vehicles with different sensors and improve the observability of the system in harsh driving conditions.

Reliable learning-based vehicle odometry system: In Section 3.3, a learning-based vehicle odometry system was presented that uses proprioceptive sensors of the vehicle. To reduce the effect of varying bias in the IMU measurements, one potential approach is to augment the learning-based model with vehicle kinematics and dynamics models that can correct the input features to the regression model. On the other hand, in case of a failure in the operation of a sensor, one can study how it affects the estimation performance and how it is going to be detected, diagnosed, and handled.

Hybrid model-learning-based vehicle odometry system: Since learning-based estimators become unreliable over unseen data, the model-based vehicle odometry can be added in a hybrid approach in case of lacking enough training data at some working points. This can be beneficial especially on slippery roads when proprioceptive sensors such as wheel encoders become unreliable while exteroceptive sensors such as camera and Lidar remains unaffected.

Adding more environmental landmarks: In Chapter 5, building planes, light-poles, and road curbs were used for the landmark-based self-localization. Other types of landmarks from the road such as lane markings can be used to improve the self-localization performance in unstructured environments, especially with low lateral excitation. To improve longitudinal self-localization accuracy, road signs can also be used. Additionally, data-driven models can be used to quantify the uncertainty of the self-localization output in different conditions.

References

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[2] R. Hajiloo, M. Abroshan, A. Khajepour, A. Kasaiezadeh, and S.-K. Chen, "Integrated steering and differential braking for emergency collision avoidance in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3167–3178, 2021.

[3] E. Mohammadbagher, N. P. Bhatt, E. Hashemi, B. Fidan, and A. Khajepour, "Real-time pedestrian localization and state estimation using moving horizon estimation," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–7, 2020.

[4] T. Wen, Z. Xiao, B. Wijaya, K. Jiang, M. Yang, and D. Yang, "High precision vehicle localization based on tightly-coupled visual odometry and vector HD map," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 672–679, 2020.

[5] X. Wei, I. A. Bârsan, S. Wang, J. Martinez, and R. Urtasun, "Learning to localize through compressed binary maps," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10308–10316, 2019.

[6] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, pp. 2743–2748 vol.3, 2003.

[7] C. Guo, M. Lin, H. Guo, P. Liang, and E. Cheng, "Coarse-to-fine semantic localization with HD map for autonomous driving in structural scenes," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1146–1153, 2021.

[8] C. Zhang, H. Liu, Z. Xie, K. Yang, K. Guo, R. Cai, and Z. Li, "AVP-Loc: Surround view localization and relocalization based on HD vector map for automated valet parking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5552–5559, 2021.

[9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[10] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[11] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[12] S. Cho, C. Kim, M. Sunwoo, and K. Jo, "Robust localization in map changing environments based on hierarchical approach of sliding window optimization and filtering," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3783–3789, 2022.

[13] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 251–263, 1997.

[14] W.-C. Ma, I. Tartavull, I. A. Bârsan, S. Wang, M. Bai, G. Mattyus, N. Homayounfar, S. K. Lakshmikanth, A. Pokrovsky, and R. Urtasun, "Exploiting sparse semantic HD maps for self-driving vehicle localization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5304–5311, 2019.

[15] C. Zhang, L. Liu, Z. Xue, K. Guo, K. Yang, R. Cai, and Z. Li, "Robust lidar localization on an HD vector map without a separate localization layer," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5536–5543, 2021.

[16] Y. Gal, *Uncertainty in deep learning*. PhD thesis, University of Cambridge, 2016.

[17] Q. V. Le, A. J. Smola, and S. Canu, "Heteroscedastic gaussian process regression," in *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, (New York, NY, USA), p. 489–496, Association for Computing Machinery, 2005.

[18] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185–207, 2013.

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

[20] R. Sim and G. Dudek, "Learning and evaluating visual features for pose estimation," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1217–1222 vol.2, 1999.

[21] V. Peretroukhin, L. Clement, M. Giamou, and J. Kelly, "PROBE: Predictive robust estimation for visual-inertial navigation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3668–3675, 2015.

[22] N. Ouerhani, A. Bur, and H. Hügli, "Visual attention-based robot self-localization," in *In Proceeding of European Conference on Mobile Robotics*, pp. 8–13, 2005.

[23] P. Newman and K. Ho, "SLAM-loop closing with visually salient features," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 635–642, 2005.

[24] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson, "Landmark selection for vision-based navigation," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 334–349, 2006.

[25] S. Frintrop and P. Jensfelt, "Attentional landmarks and active gaze control for visual SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1054–1065, 2008.

[26] S. Hochdorfer and C. Schlegel, "Landmark rating and selection according to localization coverage: Addressing the challenge of lifelong operation of SLAM in service robots," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 382–387, 2009.

[27] H. Strasdat, C. Stachniss, and W. Burgard, "Which landmark is useful? learning selection policies for navigation in unknown environments," in *2009 IEEE International Conference on Robotics and Automation*, pp. 1410–1415, 2009.

[28] M. Chli and A. J. Davison, "Active matching for visual tracking," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1173–1187, 2009. Inside Data Association.

[29] A. Handa, M. Chli, H. Strasdat, and A. J. Davison, "Scalable active matching," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1546–1553, 2010.

[30] F. Crete, T. Dolmiere, P. Ladret, and M. Nicolas, "The blur effect: perception and estimation with a new no-reference perceptual blur metric," in *Human Vision and Electronic Imaging XII* (B. E. Rogowitz, T. N. Pappas, and S. J. Daly, eds.), vol. 6492, p. 64920I, International Society for Optics and Photonics, SPIE, 2007.

[31] V. Peretroukhin, W. Vega-Brown, N. Roy, and J. Kelly, "PROBE-GK: Predictive robust estimation using generalized kernels," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 817–824, 2016.

[32] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4762–4769, 2016.

[33] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[34] Y. Zhou, G. Wan, S. Hou, L. Yu, G. Wang, X. Rui, and S. Song, "DA4AD: End-to-end deep attention-based visual localization for autonomous driving," in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 271–289, Springer International Publishing, 2020.

[35] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[36] H. Liu, H. Ma, and L. Zhang, "Visual odometry based on semantic supervision," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 2566–2570, 2019.

[37] L. Wang, J. Zang, Q. Zhang, Z. Niu, G. Hua, and N. Zheng, "Action recognition by an attention-aware temporal weighted convolutional neural network," *Sensors*, vol. 18, no. 7, 2018.

[38] W. Ci, Y. Huang, and X. Hu, "Stereo visual odometry based on motion decoupling and special feature screening for navigation of autonomous vehicles," *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8047–8056, 2019.

[39] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 1–20, 2019.

[40] I. Cvišić and I. Petrović, "Stereo odometry based on careful feature selection and tracking," in *2015 European Conference on Mobile Robots (ECMR)*, pp. 1–6, 2015.

[41] R. Kottath, S. Poddar, R. Sardana, A. P. Bhondekar, and V. Karar, "Mutual information based feature selection for stereo visual odometry," *Journal of Intelligent Robotic Systems*, 2020.

[42] G. Zhang and P. A. Vela, "Optimally observable and minimal cardinality monocular SLAM," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5211–5218, 2015.

[43] H. H. Nguyen and S. Lee, "Orthogonality index based optimal feature selection for visual odometry," *IEEE Access*, vol. 7, pp. 62284–62299, 2019.

[44] P. Ganti and S. L. Waslander, "Network uncertainty informed semantic feature selection for visual SLAM," in *2019 16th Conference on Computer and Robot Vision (CRV)*, pp. 121–128, 2019.

[45] P. Ganti and S. L. Waslander, "Network uncertainty informed semantic feature selection for visual SLAM," in *2019 16th Conference on Computer and Robot Vision (CRV)*, pp. 121–128, 2019.

[46] M. Buczko and V. Willert, "How to distinguish inliers from outliers in visual odometry for high-speed automotive applications," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 478–483, 2016.

[47] A. Antonini, "Pre-integrated dynamics factors and a dynamical agile visual-inertial dataset for UAV perception," tech. rep., 2018.

[48] Y. Zhao and P. A. Vela, "Good feature matching: Toward accurate, robust VO/VSLAM with low latency," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 657–675, 2020.

[49] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[50] S. Heo, J. Cha, and C. G. Park, "EKF-based visual inertial navigation using sliding window nonlinear optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2470–2479, 2019.

[51] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3565–3572, 2007.

[52] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[53] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15–22, 2014.

[54] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, 2007.

[55] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 834–849, Springer International Publishing, 2014.

[56] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[57] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[58] J.-S. Hu and M.-Y. Chen, "A sliding-window visual-IMU odometer based on tri-focal tensor geometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3963–3968, 2014.

[59] W. Lee, K. Eckenhoff, P. Geneva, and G. Huang, "Intermittent GPS-aided VIO: Online initialization and calibration," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5724–5731, 2020.

[60] G. Cioffi and D. Scaramuzza, "Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5089–5095, 2020.

[61] Y. Liang, S. Müller, D. Schwendner, D. Rolle, D. Ganesch, and I. Schaffer, "A scalable framework for robust vehicle state estimation with a fusion of a low-cost IMU, the GNSS, radar, a camera and Lidar," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1661–1668, 2020.

[62] K. J. Wu and S. I. Roumeliotis, "Unobservable directions of vins under special motions," *Department of Computer Science Engineering, University of of Minnesota: Minneapolis, MN, USA*, 2016.

[63] K. Zindler, N. Geiß, K. Doll, and S. Heinlein, "Real-time ego-motion estimation using lidar and a vehicle model based extended Kalman filter," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 431–438, 2014.

[64] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," tech. rep., 2015.

[65] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, 2012.

[66] B. Nisar, P. Foehn, D. Falanga, and D. Scaramuzza, "VIMO: Simultaneous visual inertial model-based odometry and force estimation," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2785–2792, 2019.

[67] B. Liang and N. Pears, "Visual navigation using planar homographies," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 1, pp. 205–210 vol.1, 2002.

[68] H. Wang, K. Yuan, W. Zou, and Q. Zhou, "Visual odometry based on locally planar ground assumption," in *2005 IEEE International Conference on Information Acquisition*, pp. 6 pp.–, 2005.

[69] J. J. Guerrero, R. Martinez-Cantin, and C. Sagüés, "Visual map-less navigation based on homographies," *Journal of Robotic Systems*, vol. 22, no. 10, pp. 569–581, 2005.

[70] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC," in *2009 IEEE International Conference on Robotics and Automation*, pp. 4293–4299, 2009.

[71] D. Scaramuzza, "1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints," *International Journal of Computer Vision*, vol. 95, no. 1, pp. 74–85, 2011.

[72] W. Zong, L. Chen, C. Zhang, Z. Wang, and Q. Chen, “Vehicle model based visual-tag monocular ORB-SLAM,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1441–1446, 2017.

[73] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, “VINS on wheels,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5155–5162, 2017.

[74] R. Kang, L. Xiong, M. Xu, J. Zhao, and P. Zhang, “VINS-Vehicle: A tightly-coupled vehicle dynamics extension to visual-inertial state estimator,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3593–3600, 2019.

[75] J. Sola, “Quaternion kinematics for the error-state Kalman filter,” *arXiv preprint arXiv:1711.02508*, 2017.

[76] A. Martinelli, “Visual-inertial structure from motion: Observability vs minimum number of sensors,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1020–1027, 2014.

[77] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[78] E. A. Nadaraya, “On estimating regression,” *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[79] G. S. Watson, “Smooth regression analysis,” *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, vol. 26, no. 4, pp. 359–372, 1964.

[80] S. Sheather, *A modern approach to regression with R.* Springer Science & Business Media, 2009.

[81] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, p. 381–395, jun 1981.

[82] P. J. Huber, “Robust statistics,” in *International encyclopedia of statistical science*, pp. 1248–1251, Springer, 2011.

[83] A. R. Amiri-Simkooei, C. C. J. M. Tiberius, and P. J. G. Teunissen, “Assessment of noise in GPS coordinate time series: Methodology and results,” *Journal of Geophysical Research: Solid Earth*, vol. 112, no. B7, 2007.

[84] "WATonoBus: Waterloo all-weather autonomous shuttle bus." https://uwaterloo.ca/watonobus/ [Online; accessed 2023-03-25].

[85] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.