TUM School of Computation, Information and Technology
Technical University of Munich

Master's Thesis in Robotics, Cognition, Intelligence

Point Clouds Localization Using Vehicle and Infrastructure LiDARs

Lokalisierung in Punktwolken mittels LiDAR Sensoren aus dem Fahrzeug und der Infrastruktur

Supervisor Prof. Dr.-Ing. habil. Alois C. Knoll

Advisor Bare Luka Žagar, M.Sc. Walter Zimmer, M.Sc.

Author Omar Elsobky

Date November 15, 2023 in Munich

Disclaimer

I confirm that this Master's Thesis is my own work and I have documented all sources and material used.

Munich, November 15, 2023

(Omar Elsobky)

Acknowledgement

I want to start the Acknowledgement chapter by saying it was a really long and hard year, yet I learned a lot! I could not do it without the support and help of the following people. My first supervisor, M.Sc. Bare Luka, whose insight and knowledge steered me through this research and drove us around to collect the dataset. My second supervisor, M.Sc. Walter Zimmer who always pushed me to achieve things as perfectly as I could. Moreover, Walter helped me collect dataset in the following aspects: data extraction, data synchronization, first steps of data registration, and visualization. Dr. Ekim Yurtsever, who always helped and guided me on my next steps and enabled me to finally understand the whole picture. My shadow supervisor, Mingyu, always gave me constructive feedback. My colleague, Alex, with whom I had long calls since our topics overlapped, also provided a docker image for MULLS.

Finally, I would like to thank the people closest to my heart whom I could not finish my master's without them. My number one supporter, my fiancee, Roaa. My whole family, especially mom and dad. My friends everyone gave me support in their own way.

Abstract

Advancements in autonomous vehicles and robotic mapping have underscored the importance of reliable LiDAR pose estimation, often referred to as LiDAR Odometry, which is essential for precise localization and environment perception in diverse and challenging scenarios. Traditional LiDAR odometry methods typically rely on vehicle-mounted LiDAR, providing localization data relative to the vehicle's starting position and incrementally constructing a map based on the sensor's field of view (FoV). However, this map is initially confined to the sensor's FoV at any given moment and expands as the vehicle traverses the environment. For this, we propose a different approach than all modern and state-of-the-art methods: vehicle-infrastructure localization. We supplement the pose estimate method with two LiDARs: a dynamic, vehicle-mounted LiDAR and a static, infrastructure-mounted LiDAR. The proposed approach provides more environmental information and ensures localization within a global frame, leveraging the predefined pose of the infrastructure. We detail the simple dataset generation process for vehicle-infrastructure localization, from data collection to ground truth generation; the creation and modification of classical and state-of-the-art (SOTA) LiDAR odometry; the identification of challenges encountered; and the comparison of performance across different versions of LiDAR odometry methods, including Classical Point-to-Point ICP and KISS-ICP. Finally, we present extensive experiments and comprehensive results for both the online and offline versions of Classical P2P ICP and KISS-ICP.

Zusammenfassung

Fortschritte bei autonomen Fahrzeugen und robotergestützter Kartierung haben die Bedeutung einer zuverlässigen LiDAR-Positionsbestimmung hervorgehoben, die oft als LiDAR-Odometrie bezeichnet wird und für eine präzise Lokalisierung und Wahrnehmung der Umgebung in vielfältigen und herausfordernden Szenarien unerlässlich ist. Traditionelle Methoden der LiDAR-Odometrie verlassen sich in der Regel auf fahrzeugmontierte LiDARs, die Lokalisierungsdaten relativ zur Startposition des Fahrzeugs liefern und schrittweise eine Karte basierend auf dem Sichtfeld des Sensors (FoV) erstellen. Diese Karte ist jedoch zunächst auf das FoV des Sensors in jedem gegebenen Moment begrenzt und erweitert sich, während das Fahrzeug die Umgebung durchquert. Hierfür schlagen wir einen anderen Ansatz vor als alle modernen und neuesten Methoden: vehicle-infrastructure localization. Wir ergänzen die Schätzmethode der Position mit zwei LiDARs: einem dynamischen, fahrzeugmontierten LiDAR und einem statischen, infrastrukturmontierten LiDAR. Der vorgeschlagene Ansatz bietet mehr Umgebungsinformationen und gewährleistet eine Lokalisierung innerhalb eines globalen Rahmens, indem er die vordefinierte Position der Infrastruktur nutzt. Wir erläutern den einfachen Prozess der Datensatzgenerierung für vehicle-infrastructure localization, von der Datenerfassung bis zur Erstellung von Ground-Truth-Daten; die Erstellung und Modifika-

tion klassischer und neuester (SOTA) LiDAR-Odometrie-Methoden; die Identifizierung von Herausforderungen; und den Vergleich der Leistung verschiedener Versionen von LiDAR-Odometrie-Methoden, einschließlich der klassischen Punkt-zu-Punkt-ICP und KISS-ICP. Abschließend präsentieren wir umfangreiche Experimente und umfassende Ergebnisse für die Online- und Offline-Versionen des klassischen P2P ICP und KISS-ICP.

Contents

Chapter 1

Introduction

"*Not all those who wander are lost.*" - J.R.R. Tolkien.

I have to disagree with one of my favorite quotes, which shows wandering does not always mean you are lost. However, in the autonomous driving field, it's critical to remember that any deviation or 'wandering' does imply that the vehicle is lost.
Autonomous driving represents a transformative intersection of technology, society, and transport, promising significant shifts in our everyday lives. The vision of fully autonomous vehicles navigating seamlessly through traffic without human intervention is progressively becoming a reality. This intriguing innovation, however, is hinged on five critical components that collectively form the backbone of self-driving systems.

The first component is perception, where the vehicle uses many sensors such as cameras, LiDAR, and radar to perceive the surrounding environment. Second, localization, in which the vehicle accurately identifies its location and orientation relative to its environment, often via GPS, sensors information, and high-definition maps. The third component is planning, where the system strategies a safe and efficient path based on the perceived environment and localization. Fourth, control involves executing the planned route by controlling the vehicle's movements. Finally, the human-machine interface represents the fifth component, providing communication between the autonomous system and the human occupants or users, ensuring that the system's intentions and current status are understandable. Each of these components is interdependent, necessitating harmonious operation to actualize the complex task of autonomous driving.

With the advancement of the autonomous vehicle field, localization has become a critical area of research. Traditional GPS-based localization methods are not always sufficient, particularly in urban environments where GPS signals can be unreliable or unavailable. Autonomous vehicles require decimeter-level positioning for highway operation and near-centimeter level on local and residential streets [Rei+19].

Recently, Light Detection and Ranging (LiDAR) sensors have shown promise in improving the localization of autonomous vehicles. However, the use of LiDAR for localization presents unique challenges and opportunities. Most research to date has focused on vehicle-mounted LiDAR, but infrastructure-based LiDAR presents an untapped resource that could potentially enhance localization capabilities.

1.1 Motivation

Researching in autonomous driving in general and in localization specifically has a great motivation behind it, starting with decreasing the number of transportation accidents. According to recent statistics by the Federal Statistical Office (Destatis), the number of people injured in road traffic accidents in Germany in February 2023 increased by approximately 5% compared to the same month of the previous year [Fed23]. Another growing problem is traffic congestion, the INRIX 2023 scorecard reveals the extent of traffic congestion in Germany, with 19.7% Travel Time Tax1 (T3), showing that a journey on roads in urban areas during peak weekday driving hours takes nearly 20 percent longer on average compared to the same journey in uncongested conditions [INR]. Moreover, [INR] shows drivers in Munich waste 35 hours per year in traffic due to congestion. One of the most critical parts of autonomous driving is localizing the vehicle within its surroundings. The research in vehicle or mobile robot localization has been highly intensive, specifically in simultaneous localization and mapping (SLAM). SLAM is when the vehicle simultaneously maps its surroundings and localizes itself within the constructed map. Furthermore, variants of different types of SLAM have emerged because of extensive research during previous decades. On the one hand, there are SLAM variants for visual sensors like cameras and ToF cameras. These are called Visual-SLAMs [Kan+]. On the other hand, LiDAR-SLAM variants use LiDAR sensors, where the most focus in these variants was the SLAM front part, LiDAR Odometry.

The LiDAR Odometry research has done a great job of localizing the vehicle using point clouds generated from vehicle-mounted LiDARs. Lasers, with their higher precision relative to cameras, ToF, and various other sensors, provide point clouds that offer highly accurate distance measurements. This makes them exceptionally effective for creating maps through SLAM. However, these point clouds often lack the fine detail and density found in images, which can result in a shortage of features necessary for accurate matching.

Numerous methods have emerged during the last decades. For instance, [Viz+23], [Pan+21], [Del+22], [ZS14],[Zha+23], [BS18], and [Des18]. Each of these methods does LiDAR pose estimates (localization) using different approaches to the same problem. Some rely on hand-tuned feature extraction, which requires tedious parameter tuning that depends on sensor resolution, environment structure, etc. [Viz+23]. Other methods are sensor ambiguous, thus only requiring a few parameters to be tuned. However, all LiDAR odometry methods only rely on one source of input, the vehicle-mounted LiDAR. This observation begs the question: What if we could enhance the localization process by incorporating an additional layer of data?

1.2 Objective

The observation discussed earlier opens several research directions for exploration, such as investigating the impact of integrating a secondary input on the localization accuracy of autonomous vehicles and developing novel LiDAR odometry methods that leverage dual inputs from two different LiDARs, one stationary and one dynamic, or both are dynamic. Before embarking on these potential research directions, a fundamental question arises: how do LiDAR Odometry methods perform when a secondary input is introduced?

The primary objective of this thesis is to evaluate the performance of state-of-the-art and classical LiDAR odometry methods when they are tasked with localizing a vehicle using dual inputs. It involves analyzing the necessary assumptions, identifying the challenges encountered, and comparing the performance across different versions of LiDAR odometry methods. The two inputs are from a dynamic, vehicle-mounted LiDAR and a static, infrastructure-mounted LiDAR. Throughout this thesis, this setup will be referred to as vehicle-infrastructure localization. A diagrammatic representation of this concept is presented in Figure 1.1, illustrating the basic principle of adding the secondary input through a static infrastructure-mounted LiDAR. At the same time, the vehicle is equipped with a dynamic LiDAR. This exploration aims to elucidate not only the advantages but also the potential challenges of enhancing vehicle localization systems with this sophisticated sensing technology.

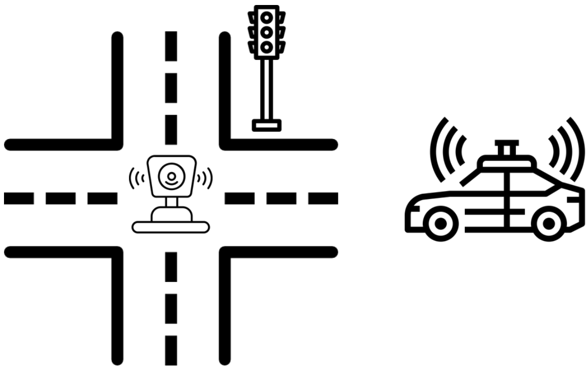


Figure 1.1: Vehicle-infrastructure localization Diagram

1.3 Structure of the Thesis

The thesis unfolds systematically, beginning with Chapter 2, Background Chapter, which lays the foundational knowledge necessary to comprehend the specialized field of study. Chapter 3, Related Work Chapter, delves into the contributions of the scientific community in LiDAR localization, highlighting state-of-the-art LiDAR SLAM and odometry. The collection of needed data, our contributions to state-of-the-art methods, and the specific methodologies we've developed are thoroughly articulated in Chapter 4, Methodology Chapter. The subsequent analysis of our methods, encompassing both quantitative and qualitative assessments, and engaging in a critical discussion of our results is presented in Chapter 5, Experiments Chapter. In the final chapter, Conclusion & Future Work Chapter, we discuss the key findings that encapsulate our conclusions, challenges, and limitations and show future work.

Chapter 2

Background

In this chapter, the foundational knowledge required for the research is laid out. The discussion is segmented into three distinct sections:

1. **Sensors**: This section delves into the diverse sensors integral to autonomous driving and vehicle localization. Each sensor type, the data it yields, and the methodologies to interpret and handle the sensor's data are explored.

2. **Localization Algorithms Basics**: Fundamental algorithms and concepts, such as SLAM and Odometry, that are essential for vehicle localization are discussed here.

3. **Datasets**: The chapter concludes by examining publicly available vehicle datasets, elucidating their structure, and pinpointing the potential application areas.

2.1 Sensors

2.1.1 GPS

The Global Positioning System (GPS) is a satellite-based navigation system that provides accurate positioning information to receivers anywhere on Earth. GPS satellites transmit signals that are received by GPS receivers. The receivers then use these signals to calculate their position, speed, and time.

GPS Format

There are multiple ways to represent the GPS data. These ways include the following:

- **Latitude/Longitude**

- **Universal Transverse Mercator (UTM):** a map projection system for assigning coordinates to locations on the surface of the Earth.

GPS Disadvantages

There are a number of limitations to using GPS alone for vehicle positioning. These limitations include:

- **Accuracy:** The accuracy of GPS positioning can be affected by a number of factors, including the number of satellites in view, the quality of the receiver, and the environment. According to [Alo+11], in urban areas, the accuracy of GPS positioning can be as low as $10-15$ meters.

- **Obstructions:** GPS signals can be blocked by buildings, trees, and other objects. This can make it difficult to get a GPS signal in some areas [Alo+11].

- **Signal loss:** GPS signals can be lost in areas with high levels of electromagnetic interference, such as near power lines or cell towers.

2.1.2 LiDAR

LiDAR, an acronym for "light detection and ranging" serves as a pivotal technology that employs laser beams to generate a 3D representation of the surveyed environment. This method determines ranges by targeting surfaces with a laser and gauging the time taken for the reflected light to return to its receiver. LiDAR units emit lasers and measure distances based on the laser returns they receive from reflecting surfaces [LI20]. Moreover, Figure 2.1 presents a conceptual representation of the LiDAR operating principle, whereas, in most commercial LiDARS, the data is generated in a point cloud format. LiDAR is essential in different sectors, including automotive, infrastructure, and robotics, due to its ability to produce high-resolution maps and facilitate precise data interpretation in diverse conditions [Lid23]. In particular, autonomous vehicles leverage LiDAR to create a meticulous map of ever-evolving surroundings, ensuring safe navigation. Its high distance accuracy further enables the vehicle's system to discern and evade obstacles, even in diverse weather and lighting conditions.

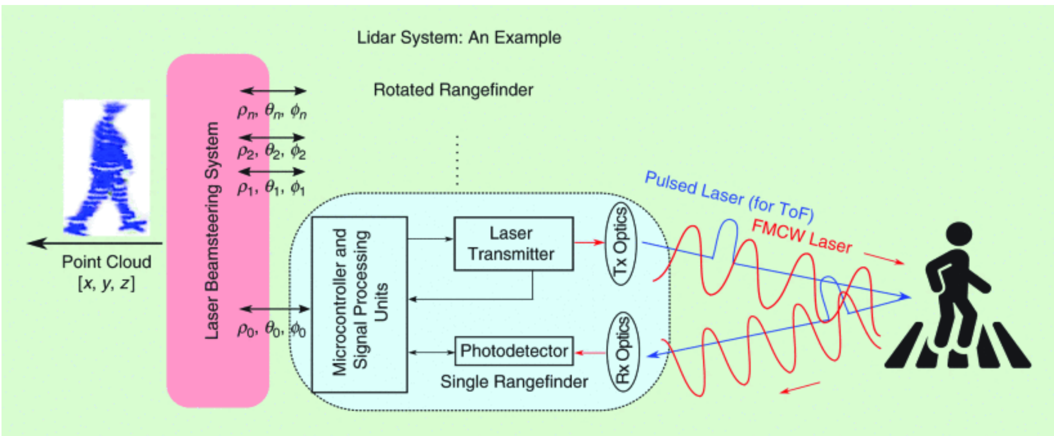


Figure 2.1: This illustrates a ToF laser rangefinder. The device uses either a direct or coherent method to measure the distance in a specific direction, which is directed by the scanning system. Tx represents the transmitter, while Rx denotes the receiver. [LI20]

Additionally, according to [LI20], LiDAR outputs are used for tasks such as object detection, classification, tracking, and predicting intentions, as they consist of various layers of information. Since LiDARs have a robust and wide-ranging accuracy, the provided physical information is highly reliable. Furthermore, applications of LiDAR extend to areas such as geodesy, archaeology, geology, and more, underscoring its multifaceted utility [Wik23a].

LiDAR Point Cloud

A point cloud consists of individual data points located in space. These points can depict a three-dimensional form or entity, with each point defined by its Cartesian coordinates (X, Y, Z). The organization of these points isn't critical to the object's representation. Using the point clouds comes with the following pros and cons:

- **Pros**:
  - Efficient regarding storage needs relative to other 3D data methods.
  - Enhanced sensors facilitate access to high-quality point cloud data.
- **Cons**:
  - Point clouds lack inherent structure, presenting challenges for learning-based models.
  - Retrieving specific localized information is challenging due to the lack of order among points.

2.1.3 IMU

An Inertial Measurement Unit (IMU) is a sophisticated electronic device that quantifies and conveys specific force, angular rate, and occasionally the orientation of a particular body [23]. This is achieved by merging data from accelerometers and gyroscopes, with some IMUs incorporating magnetometers. An IMU's intrinsic functionality hinges on principles of gravity and fundamental physics, rendering it capable of relaying data unaffected by external conditions [Fen23]. Thus, even when other perception sensors are compromised due to adverse weather, the IMU provides a consistent feed, assisting vehicles in maintaining their trajectory securely. This reliability is paramount for the viability of autonomous vehicles in urban and highway contexts.

The accelerometers within the IMU measure linear acceleration across three orthogonal axes. Through temporal integration of this acceleration, velocity is derived, and further integration offers positional shifts. On the other hand, gyroscope sensors assess the angular rate along three orthogonal axes. Time-integrated angular rates produce variations in roll, pitch, and yaw, shown in Figure 2.2, essentially capturing the attitude adjustment of an object. This combination of gyroscopic and accelerometer data can provide measurements encompassing 6 degrees of freedom, commonly referred to as 6-DOF [Fen23].

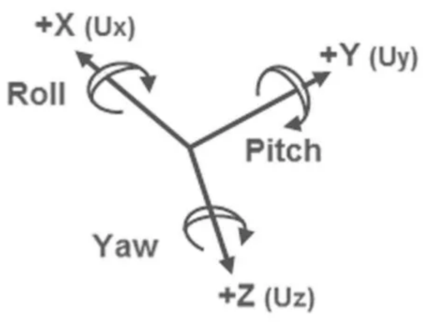


Figure 2.2: The IMU leverages its accelerometer and gyroscope to record measurements across six degrees of freedom: three translational movements (forward-backward, side-to-side, and vertical) and three rotational movements (roll, pitch, and yaw) [Fen23].

However, despite its sophisticated measurement capabilities, the accuracy of IMU-based information can diminish rapidly due to inherent drift, contingent on the quality of the IMU itself [IT20]. To counteract this, it's imperative to complement the IMU data with positioning information from an alternative source.

2.2 Localization Fundamentals

Vehicle localization, an essential task in autonomous driving and advanced driver assistance systems, can be achieved using various methods and sensors. In this section, Localization fundamentals will be presented. Firstly, the Basic Simultaneous Localization and Mapping (SLAM) and Odometry are explained briefly. Then, the focus is shifted to the LiDAR domain, LiDAR SLAM, and LiDAR Odometry, as the main focus of the research is within the LiDAR domain.

2.2.1 Simultaneous Localization and Mapping (SLAM) & Odometry

In the fundamental section on Simultaneous Localization and Mapping (SLAM), a comprehensive understanding of this technology is essential. SLAM is a critical process through which a mobile robot constructs a map of an unknown environment while concurrently locating itself within that map [DB06]. It negates the requirement for any prior knowledge of the mobile location, simultaneously estimating the trajectory and landmark positions in real-time. SLAM not only establishes a robot's autonomous navigation capabilities by creating a self-reliant map but also fundamentally challenges the robot by placing it in an unfamiliar location to build and navigate the map concurrently [DB06].

Odometry, a foundational element of SLAM, involves estimating a mobile robot's path using observed readings. It serves as the groundwork upon which a robot can track its position and maintain a local map [Yan+22]. SLAM builds upon odometry, expanding it to create a globally consistent map and trajectory. This is achieved primarily through loop closures, which are critical for global coherence—without them, SLAM would be reduced to basic odometry [Cad+16]. The advancement of SLAM over the last decade has seen its application extend from terrestrial robots to those in underwater and aerial environments [Yan+22]. While traditional odometry methods, such as wheel encoders, were susceptible to drift, contemporary approaches leveraging visual and inertial cues have significantly minimized drift, making them more reliable for accurate navigation [Cad+16].

In SLAM systems, the front end and back end have distinct but interconnected roles, creating a dynamic between raw data processing and abstract data inference. The front end is responsible for distilling raw sensor information into a form that's easier to manage. For example, vision-based SLAM extracts key points from images, simplifying complex data into identifiable features [Cad+16]. On the other side, the back end takes these simplified models and infers the relationships between them, applying theories from graph and probabilistic estimation to optimize the map and trajectory of the robot [Cad+16]. This dual structure is essential for SLAM, allowing the system to efficiently interpret sensor data and adjust its understanding of the environment, mainly through a process known as loop closure [Cad+16]. This critical interaction between the front end and back end in ensuring SLAM's accuracy is depicted in Figure 2.3.

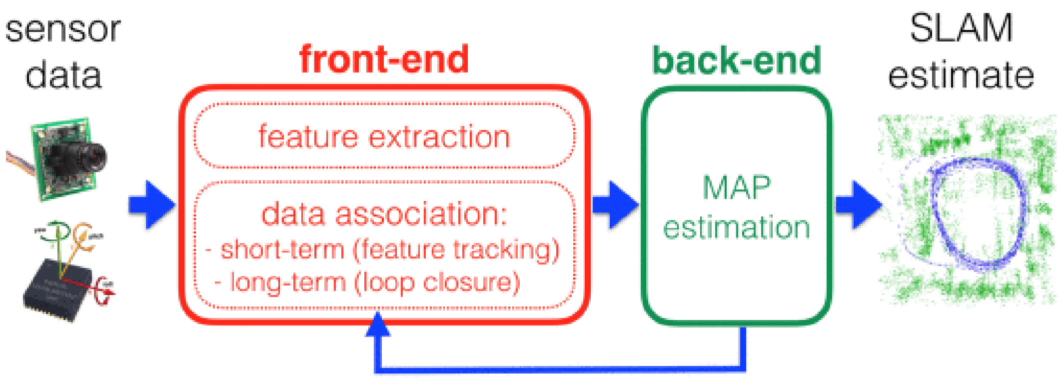


Figure 2.3: Front end and back end processes in a standard SLAM setup. The back end is designed to offer insights to the front end to assist in detecting and confirming loop closures [Cad+16].

With the evolution of 3D SLAM, the objective is to determine the robot's six degrees of freedom (DoF) pose and simultaneously map the environment in three dimensions using sensors that provide image and point cloud data [GSG22]. The transition to LiDAR SLAM and LiDAR Odometry is a focal point due to its precision and relevance to current research domains. The following section delves into LiDAR-based SLAM and Odometry, pivotal for advanced autonomous vehicle navigation, where accurate 3D environmental mapping is crucial.

2.2.2 LiDAR Odometry & Iterative Closest Point (ICP)

In the field of autonomous navigation, LiDAR SLAM and Odometry stand out for their ability to provide high-fidelity environmental mapping and precise localization. Unlike visual inputs from standard or Time-of-Flight (ToF) cameras, The essential task of LiDAR SLAM is matching scans. Thus, In the LiDAR sensing domain, the main focus of research is the SLAM's front end part, LiDAR Odometry. As [Viz+23] mentioned, nearly all modern SLAM systems build on top of odometry algorithms.

LiDAR Odometry generates vehicle pose estimates by matching the current scan point cloud with previous scans point cloud. The key algorithm in point cloud matching, or as known point cloud registration, is Iterative Closest Point (ICP).

Iterative Closest Point (ICP) is a foundational algorithm used in LiDAR SLAM for aligning two point clouds to a common reference frame. As mentioned in one of the state-of-the-art LiDAR odometry methods [Viz+23], the ICP algorithm is based on a fundamental two-step iterative process. Initially, it identifies the corresponding points between the two data sets. Subsequently, it calculates the optimal transformation—typically involving rotation and translation—that minimizes a predefined error metric based on these correspondences. This iterative cycle is repeated until the algorithm converges to a stable solution according to a set convergence criterion. A significant limitation of ICP is its dependency on an accurate initial pose estimate; a poor estimate may result in convergence to a local minimum. Authors of [ZPK18] describe the above-mentioned Classical ICP algorithm using the steps shown in Algorithm 1 :

In Figure 2.4, we see a visual representation of the Iterative Closest Point (ICP) process, depicted in four stages using the three point clouds of the Stanford Bunny dataset to illustrate ICP steps [Sta93] [Lam19]. Initially, in Figure 2.4a, the point clouds are in their original

Algorithm 1 Iterative Closest Point Algorithm

Input: Target point cloud $\mathbf{P}$, source point cloud $\mathbf{Q}$
Output: Transformation matrix $\mathbf{T}$
Initialize $\mathbf{T}$ using initial guess
repeat
    Find the correspondence set $\mathcal{C} = \{(p, q)\}$ from $\mathbf{P}$ and $\mathbf{Q}$ transformed by $\mathbf{T}$
    Update $\mathbf{T}$ by minimizing the objective function $E(\mathbf{T})$ defined over $\mathcal{C}$
until convergence

state, oriented differently, which sets the stage for the ICP algorithm to begin its work. The next step, Figure 2.4b, demonstrates the initiation of the ICP algorithm where it identifies correspondences between the point clouds, bringing them into an initial rough alignment yet still spatially separated. As the ICP iterates, shown in Figure 2.4c, the point clouds incrementally adjust their orientations and positions, progressively reducing the distance between them. The final state, Figure 2.4d, showcases the convergence of the ICP algorithm, where the point clouds have been fully aligned, indicating the successful registration of the datasets into a common frame of reference. This visual progression underscores the efficacy of the ICP algorithm in aligning disparate 3D data into a coherent structure.

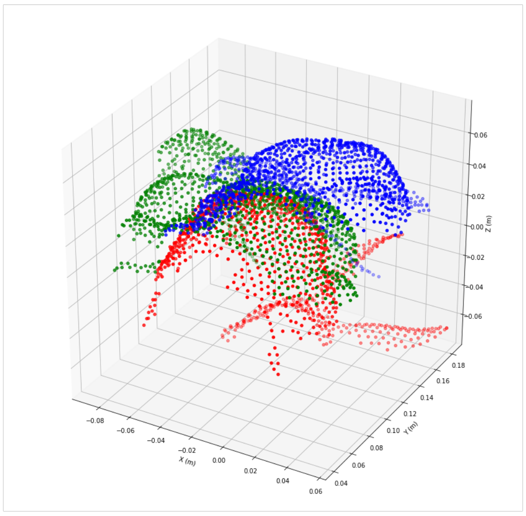
The classical approach in the Iterative Closest Point (ICP) algorithms is the point-to-point method, which seeks to minimize the objective function $E(T) = \sum_{(p,q)\in\mathcal{C}} \|p - Tq\|^2$. This function computes the sum of squared distances between corresponding points from the target point cloud $\mathcal{P}$ and the source point cloud $\mathcal{Q}$ transformed by the current transformation matrix T, essentially aligning the two point clouds by finding the best-fit transformation [BM92]. While effective, several variants have emerged. For instance, the point-to-plane ICP algorithm enhances the basic approach by minimizing a different objective function, [CM92], $E(T) = \sum_{(p,q)\in\mathcal{C}}((p - Tq) \cdot n_p)^2$, where n_p is the normal of the point p, thus often achieving faster convergence rates [RL01].

2.3 Datasets

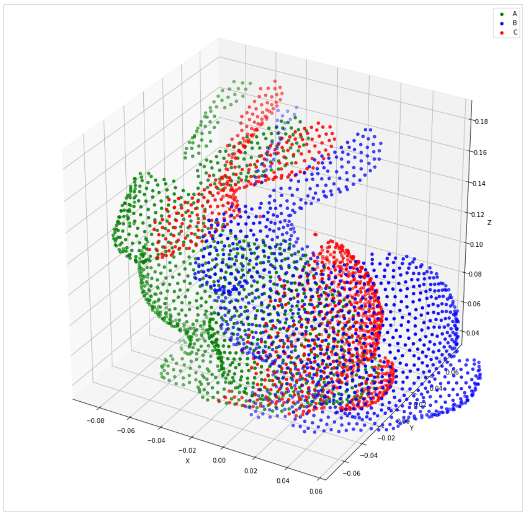
Datasets play a crucial role in the field of vehicle localization, as they provide the means for testing and verifying localization methods. LiDAR datasets for vehicle localization vary in structure, content, and format. In this section, we provide a brief overview of some of the most popular vehicle LiDAR datasets, highlighting their key features and structure.

2.3.1 KITTI

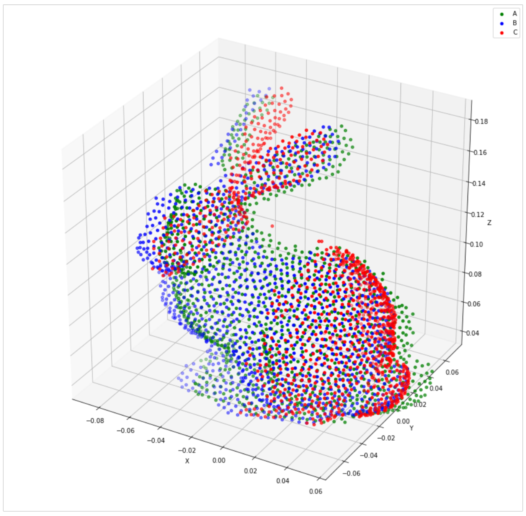
Content: The KITTI dataset includes raw sensor data from stereo cameras, LiDAR, GPS, and IMU sensors. It provides ground truth for various computer vision tasks, such as object detection, tracking, and depth estimation [GLU12].
Structure: The dataset is organized into several sequences subfolders for different types of data (e.g., images, LiDAR point clouds, calibration data, etc.) Each subfolder contains timestamped files corresponding to different data modalities [GLU12].



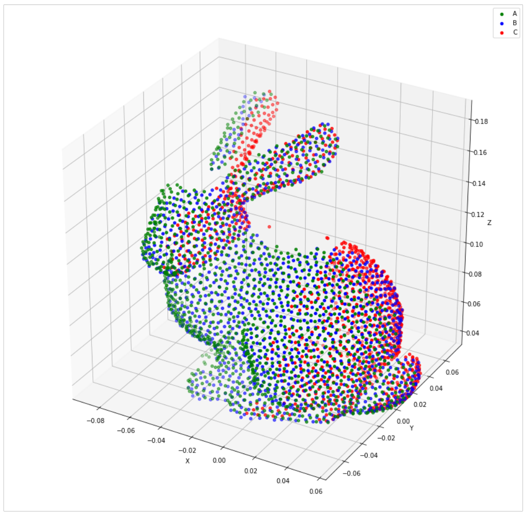
(a) Original state of point clouds [Lam19].



(b) Start of ICP - point clouds aligned but displaced [Lam19].



(c) Point clouds getting closer [Lam19].



(d) Fully registered point clouds [Lam19].

Figure 2.4: Visualization of the ICP process with three point clouds representing bunnies [Lam19].

2.3.2 TUM RGB-D

Content: The TUM dataset comprises RGB-D data (color images with depth information) collected with various camera setups. It is primarily used for visual SLAM benchmarks [Stu+12].
Structure: The dataset is divided into several sequences, each containing color images, depth images, and camera pose ground truth. Each sequence is stored in its own folder, and files are named according to their timestamps. The high-precision ground-truth trajectories are obtained using a sophisticated motion-capture system equipped with eight high-speed cameras operating at 100 Hz, thereby providing an accurate evaluation measure for the performance of the visual SLAM system [Stu+12].

2.3.3 NCLT

Content: The NCLT dataset contains data from various sensors, including LiDAR, GPS, IMU, and cameras. It is primarily used for long-term SLAM research [CUE16].
Structure: The dataset is organized into subfolders for different days of data collection. Each day's subfolder contains separate directories for different sensor modalities, with timestamped files for each data type [CUE16].

2.3.4 MulRan

Content: The MulRan dataset includes a single LiDAR and a single radar, data collected in various environments as it was collected in different cities. Each city corresponds to a specific sequence. The following sequences are available: DDC, KAIST, Riverside, and Sejong City. [Kim+20].
Structure: The dataset is divided into several sequences, with each sequence containing LiDAR data in binary files for different sensors and a pose file for the scans identical to KITTI format. For the radar data, MulRan provides polar images and ray data, respectively. The data is stored in a hierarchical folder structure, with each sequence having its own sub-directories [Kim+20].

Chapter 3

Related Work

This chapter starts by covering the literature review about prior work in Section 3.1. Then, give an overview of the chosen registration methods for the methodology chapter and why these methods were chosen in Section 3.2.

3.1 Current State of Research

Vehicle localization can be achieved using different types of methods. Each method requires different inputs to be able to do vehicle localization. In the following subsections, an overview of different state-of-the-art methods is given. For this thesis scope, the focus will be on LiDAR-only methods for vehicle localization, as LiDAR sensors are known for their high accuracy and reliability in various environmental conditions.

3.1.1 Simultaneous Localization and Mapping (SLAM)

SLAM uses a combination of different sensor readings to simultaneously create a map of the environment and determine the vehicle's position within it. Following are the LiDAR SLAM state-of-the-art methods.

KISS-ICP

KISS-ICP (Keep It Small and Simple Iterative Closest Point) stands out as a notably effective and simple method for vehicle localization. Contrary to the trend of adding complexity to sensor odometry methods to improve their performance [Viz+23], KISS-ICP takes a different approach by eliminating unnecessary components and concentrating on the core elements of the algorithm [Viz+23]. This focus on simplicity yields efficient results and enables KISS-ICP to perform robustly in diverse environments [Viz+23]. Furthermore, KISS-ICP employs point-to-point ICP along with adaptive thresholding for correspondence matching, enhancing its stability and accuracy. Notably, KISS-ICP is a LiDAR-agnostic method, offering flexibility and ease of configuration with only a few required parameters, as shown in Figure3.1.

Adaptive Thresholding plays a crucial role in KISS-ICP success since it is common to have a maximum value of 1 m or 2 m, which is often seen as an outlier filtering scheme. However, the threshold depends on the expected initial pose error, the number and type of moving objects within the environment, and sensor noise [Viz+23].

KISS-ICP also employs the constant velocity model for two primary reasons [Viz+23]:

- The constant velocity model offers broad applicability without necessitating additional sensors or the complexities of time synchronization between various sensors
- The constant velocity model is sufficient to supply a reliable initial guess in the search for data associations and deskewing of 3D scans.

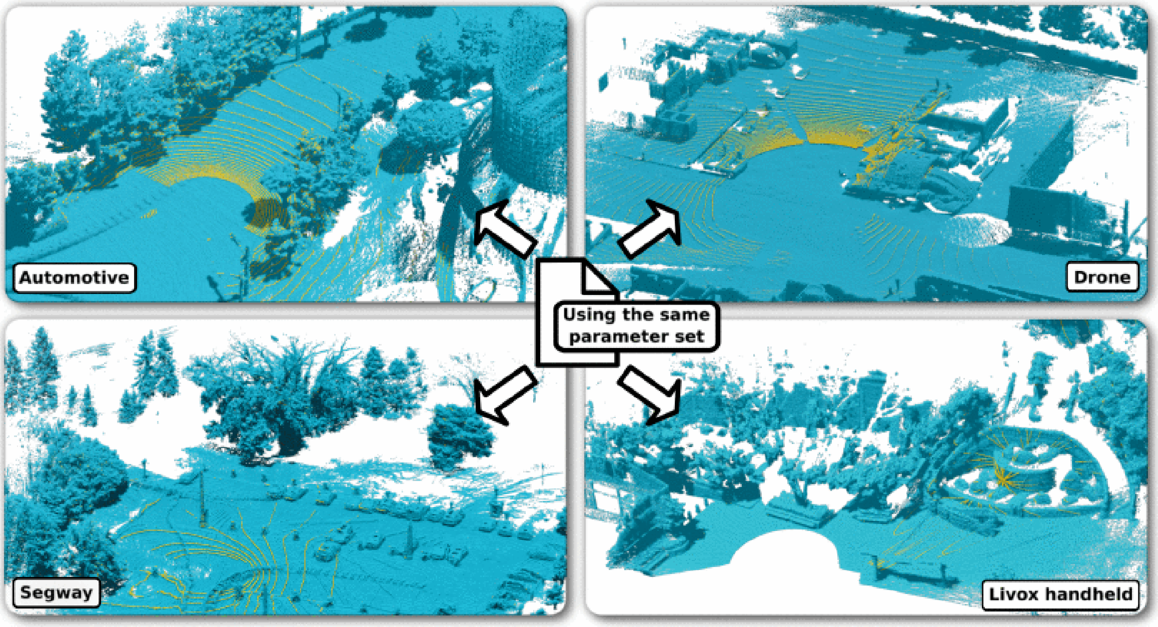


Figure 3.1: Local maps in blue generated by applying KISS-ICP to different datasets with the same parameters. The yellow scan is the latest scan. These scans were recorded using different sensors with different point densities, orientations, and shooting patterns [Viz+23].

KISS-ICP computes the trajectory of the LiDAR by registering the point clouds sequentially. For each scan $\mathcal{P} = \{p_i \mid p_i \in \mathbb{R}^3\}$ Five steps are preformed to obtain the global pose estimate $T_t \in \mathrm{SE}(3)$ at time t [Viz+23]. These steps are

1. **Sensor Motion Prediction and Motion Compensation (Deskewing)**: The first step involves applying sensor motion prediction and motion compensation, often referred to as deskewing, to undo the distortions of the 3D data caused by the sensor's motion during scanning.

2. **Subsampling the Current Scan**: The second step involves subsampling the current scan to reduce the data processed in the subsequent steps.

3. **Estimating Correspondences**: The third step involves estimating correspondences between the input point cloud and a reference point cloud, called the local map. An adaptive thresholding scheme is used for correspondence estimation, restricting possible data associations and filtering out potential outliers.

4. **Registering the Input Point Cloud to the Local Map**: The fourth step involves registering the input point cloud to the local map using a robust point-to-point ICP algorithm.

5. **Updating the Local Map**: The final step involves updating the local map with a downsampled version of the registered scan.

As discussed, KISS-ICP is very simple yet efficient due to focusing on the core elements of localization. In the following subsection, a brief overview of CT-ICP will be given.

CT-ICP

CT-ICP (Continuous-Time Iterative Closest Point) is a unique method for vehicle localization that seeks to leverage both continuity in scan matching and discontinuity between consecutive scans. Enabling elastic distortion of scan during registration for increased precision, and discontinuity for increased robustness to high frequency motions [Del+22].

CT-ICP odometry has two poses for each scan, As shown in Figure 3.2:

- a pose for the beginning of the scan $\Omega_b^n = (R_b^n, t_b^n)$ (where R denotes rotation, t denotes translation, and b denotes beginning)

- a pose for the end of the scan $\Omega_e^n = (R_e^n, t_e^n)$ (where e denotes end).

For each new scan S_n the following steps are executed:

1. Extract a sample of keypoints indexed by I_n using grid sampling.

2. Register the keypoints into the local map, which is a dense point cloud M_n built from all previously registered scans and stored in a sparse voxel grid.

3. Estimate the two optimal poses Ω_b^* and Ω_e^* to handle the scan distortion during optimization.

4. Transform the points to the world frame and add them to the local map.

5. Express each sample point p_i in the world frame, compute the normal of the point's neighborhood in the local map, and the transformation from the LiDAR frame at time τ_i to the world frame.

6. Estimate the transformation by interpolating between Ω_b and Ω_e with $\alpha_i = (\tau_i - \tau_b)/(\tau_e - \tau_b)$ and use the spherical linear interpolation (SLERP), [con23] for rotation interpolation.

7. Introduce weights to favor planar neighborhoods, defined by the planarity of the neighborhood of p_i^W, which is calculated based on the eigenvalues of the neighborhood's covariance.

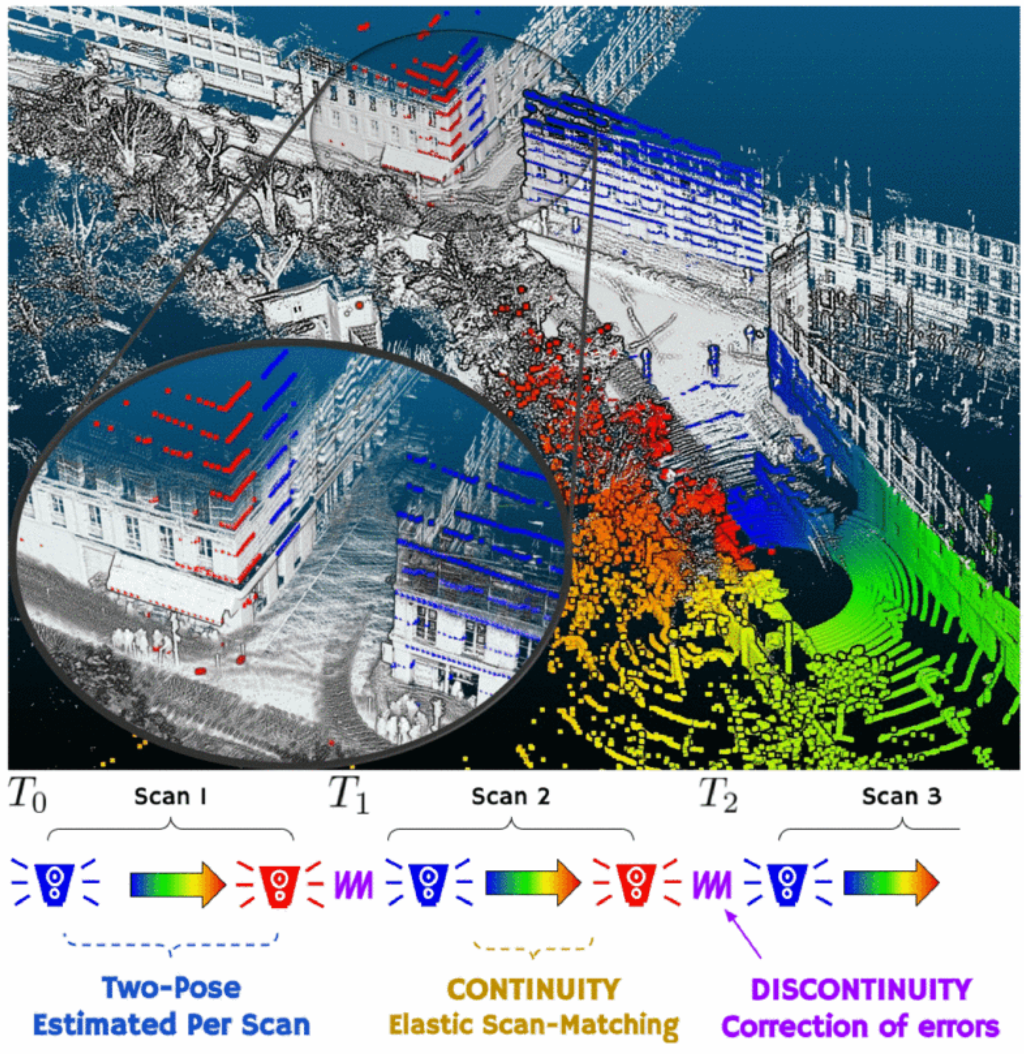


Figure 3.2: On the top figure, there are point clouds from red to blue colors, depending on the timestamp. The blue represents the oldest scan, and the red represents the newest scan. Moreover, the local map, colored white, is the target map for new scans. Where all new scans are deformed elastically to align with the local map by joint optimization of two poses at the beginning and the end of each scan, then interpolating based on the timestamp. Resulting in continuous-time scan-to-map odometry. The bottom figure CT-ICP predicted trajectory with continuity of poses intra-scan and discontinuity between scans [Del+22].

As discussed, CT-ICP is an efficient Real-time Elastic Odometry due to leveraging the benefits of both continuity and discontinuity of the scans. In the following subsection, a brief overview of MULLS-ICP will be given.

MULLS-ICP

MULLS, which has been extensively experimented with using three datasets with more than 100,000 frames collected by different types of LiDARS on various environmental types, is a versatile LiDAR SLAM via Multi-metric Linear Least Square (MULLS). MULLS consists of two components, front end, and back end [Pan+21].

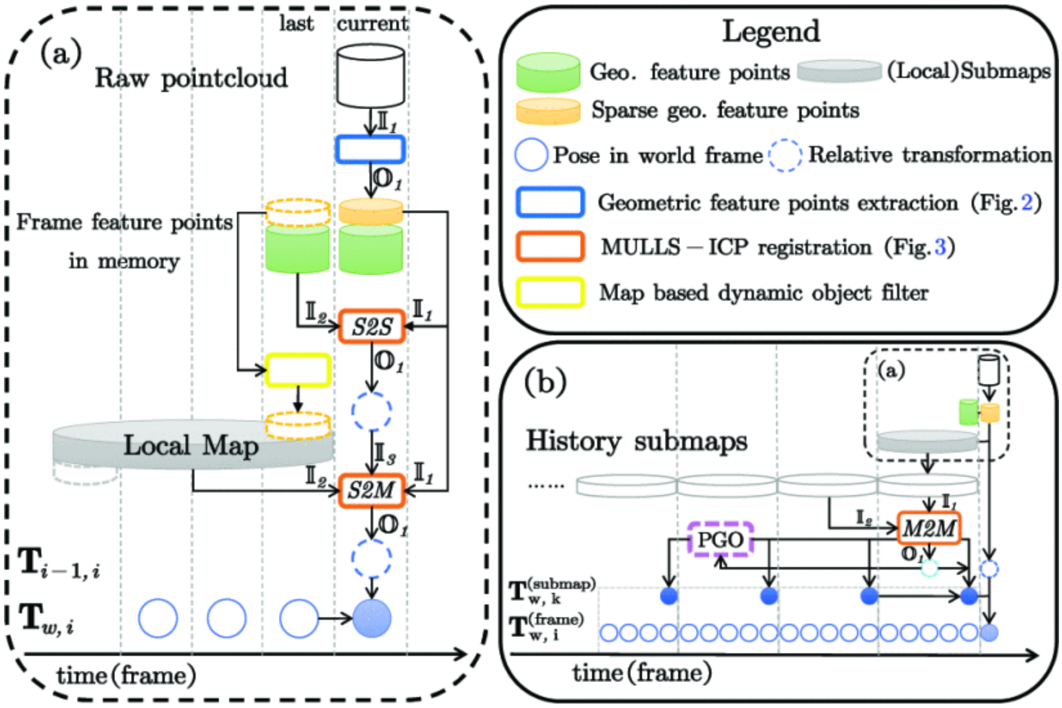


Figure 3.3: Front end and back end of the MULLS-SLAM workflow [Pan+21].

MULLS operates through a front end component. As shown in Figure 3.3, on receiving an incoming scan, it employs dual-threshold ground filtering and principal component analysis to initially classify and extract feature points, such as ground, facade, pillar, and beam. These points are further downsampled for computational efficiency. MULLS maintains a local map populated with static feature points from prior frames, set to the reference pose of the preceding frame. Utilizing the motion of the last frame as an initial approximation, a quick scan-to-scan MULLS-ICP is performed between the current frame's sparse feature points and the previous frame's dense features. This estimated transformation provides an improved starting point for the subsequent scan-to-map MULLS-ICP, which continues until reaching a defined convergence. Furthermore, dynamic object filtering is applied, and then, sparser features are added to the local map. Any non-ground feature points significantly distant from similar category points in the local map are excluded. The map is then confined to a set radius, preserving only the dense features from the current frame for future processing.

MULLS back end component reduces the drift resulting from dead reckoning by applying hierarchical pose graph optimization to stored history submaps. As depicted in Figure 3.3, these submaps are the primary processing units, where the TEASER global registration [YSC20], is used to construct and validate adjacent and loop closure edges among these submaps. TEASER is a certified and efficient system [YSC20]. Initial correspondences are selected based on the cosine similarity of NCC features among vertex keypoints. Leveraging the estimations from TEASER as a starting point, the map-to-map MULLS-ICP fine-tunes inter-submap edges, ensuring precise transformation and information matrices. Edges that do not meet the criteria of set thresholds are removed. Including a loop closure edge facilitates pose correction for free submap nodes via inter-submap pose graph optimization (PGO). On a hierarchical scale, the inner-submap PGO solidifies the reference frame for each submap while concurrently adjusting poses for other frames.

3.1.2 V2X Datasets

V2X stands for Vehicle-to-everything, which is communication between a vehicle and any entity that affects or is affected by the vehicle [Wik23b]. Lately, V2X-Seq: A Large-Scale Sequential Dataset for Vehicle-Infrastructure Cooperative Perception and Forecasting was released[Yu+23]. V2X-Seq is based on another V2X dataset, DAIR-V2X: A Large-Scale Dataset for Vehicle-Infrastructure Cooperative 3D Object Detection [Yu+22]. Both datasets were collected using the same setup presented in Figure 3.4

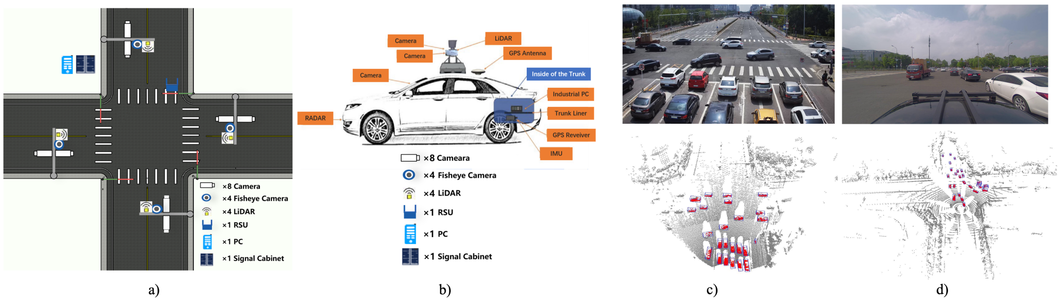


Figure 3.4: a) Infrastructure with sensors. b) vehicle sensors. c) Infrastructure image and point cloud with 3d annotation d) Vehicle image and point cloud with 3d annotation [Yu+22].

Furthermore, the V2X-Seq dataset has a diverse range of elements, including data frames, trajectories, vector maps, and traffic signal information. V2X-This dataset contains two parts: The V2X-Seq-SPD (Sequential Perception Dataset), featuring over 15,000 frames from 95 distinct scenarios, and the V2X-Seq-TFD (Trajectory Forecasting Dataset) offers around 80,000 scenarios each from infrastructure and vehicle viewpoints, along with an additional 50,000 cooperative-view scenarios [Yu+23].

3.2 Choice of Registration Methods

The registration methods selected for our research were chosen based on the availability of their code bases and their adaptability for incorporating a secondary input. These factors were crucial for integrating and modifying the algorithms to include infrastructure-mounted LiDAR data. The following registration methods are selected:

- **Classical Point-to-Point ICP:** This foundational ICP variant is included for its simplicity and the ease with which it can be implemented using the Open3D library [ZPK18]. Its classical status provides a benchmark for assessing more advanced methods.
- **KISS-ICP:** This state-of-the-art method boasts a publicly available code base that is both lightweight and user-friendly, requiring minimal parameter tuning. Its well-structured packaging facilitates straightforward integration and deployment.
- **MULLS-ICP:** Selected for its public availability and specialized design, this method focuses on aligning similar features, making it an excellent choice given the typically low overlap between vehicle and infrastructure point clouds.

Chapter 4

Methodology

This chapter provides a description of the methodology. Section 4.1, presents an overview dataset utilized in this research. Following the dataset discussion, the chapter will introduce the theoretical framework that underpins our approach. The subsequent sections will elucidate the different versions (online & offline) of two different ICP methods and modifications made to them and the reasons behind these design choices in the Algorithm Design section, 4.2.

4.1 Dataset

As vehicle-infrastructure localization is a new topic, a new data set is needed. This section will shed light on the specifics of the dataset, its significance, and the considerations taken during its selection. The process through which this dataset was gathered is interpreted in Section 4.1.1. The subsequent section, Section 4.1.2, lists the essential preprocessing steps that were necessary to prepare the data for further analysis. Additionally, details of data annotation, initial transformation matrix, and ground truth generation are illustrated.

Moreover, a description of how the dataset was collected is stated in Section 4.1.1 and the preprocessing needed for the dataset in Section 4.1.2. Additionally, details of data annotation, initial transformation matrix, and ground truth generation are illustrated.

4.1.1 Dataset Collection

Environment

For data collection, the car was driven on A9 highway in Munich, as shown in Figure 4.1. This road was chosen because it contains a highly crucial element, static infrastructure LiDAR, to this research, Providentia++ [Krä+21].

Figure 4.1: The path the vehicle took during the data collection. The first part is a road before the intersection, then the intersection, and at last is the A9 highway.

Providentia++[1], also previously known as Providentia, is a large-scale infrastructure consisting of multimodal sensors, multiple edge computing units, a complex software architecture, and a broad range of state-of-the-art algorithms. Providentia is built along the A9 highway close to Munich, and Providetina++ is the infrastructure that was extended to urban areas in Garching-Hochbrück as shown in Figure 4.2. Its primary purpose is to provide a real-time and reliable digital twin of the current road traffic at any given time or day of the year for use in a variety of applications [Krä+21].

Figure 4.2: Providentia Coverage Top view [IM23] [Krä+21].

[1]https://innovation-mobility.com/en/project-providentia/

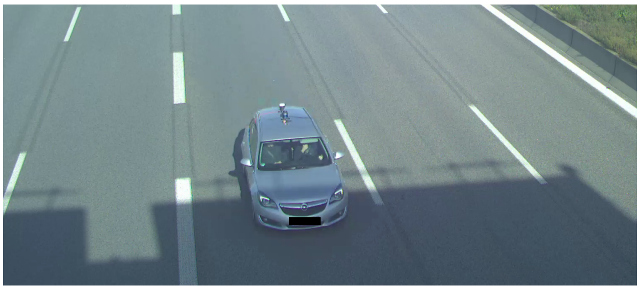


Figure 4.3: Dynamic RoboSense LiDAR, GPS, IMU, and Camera mounted on top of the vehicle.

Figure 4.4: Providentia Infrastructure. The infrastructure is basically an overhead Sign Structure that has multiple sensors mounted on top of it. One of these sensors is the static Ouster LiDAR.

Data Collection process

The most important aspect when collecting data from multiple sensors is Time Synchronization. To overcome this challenge, the dynamic LiDAR, IMU, and Camera were connected to the same notebook that was connected remotely to the static Ouster LiDAR. All mentioned sensors were synchronized to the Network Time Protocol (NTP) server at LRZ[2]. However, during the data collection, the GPS was connected using the WiFi to a phone, which resulted in 18 seconds difference between the GPS timestamps and other sensors timestamps. The time difference issue is solved in data prepossessing.

4.1.2 Data Preprocessing

After the dataset collection, data preprocessing is needed to make the data readable and usable. Data preprocessing is divided into three blocks:

- **Scene Extraction**: In this step, the extraction of 500 frames is done. There are 4 data sources for each timestamp: vehicle point clouds, infrastructure point clouds, vehicle GPS poses, and vehicle IMU data.
- **Point Cloud Cleaning**: Although point cloud provides accurate measurements, it also contains an enormous amount of INF & NaN readings. Thus, the point cloud cleaning is a needed step. Furthermore, ground removal is also needed in conducting different versions of experiments.
- **GPS and IMU JSON Conversion**: As the GPS and IMU data come in LLM and CSV format, respectively, a conversion to JSON format is needed to facilitate the processing of the data.

4.1.3 Initial Transformation Matrix Calculation

The determination of an initial transformation matrix is a pivotal step in aligning the point clouds accurately. This initial transformation serves as a starting point for the registration process, providing a rough alignment that can be refined through subsequent steps. The matrix is generated by leveraging the spatial information available from both GPS locations and IMU data associated with the LiDAR sensor.

[2] https://www.lrz.de/services/netzdienste/ntp/

The process begins with extracting GPS coordinates corresponding to the LiDAR positions at different time instances. The positional information from GPS gives a broad understanding of the LiDAR's movement across the environment. Subsequently, the IMU data, which provides the orientation and acceleration information, is integrated to refine the initial estimate of the LiDAR's pose. The fusion of GPS and IMU data helps in computing an initial guess of the transformation matrix, which encapsulates the translation and rotation parameters that define the LiDAR's pose in the global coordinate system.

The mathematical formulation involves the calculation of the distance between consecutive LiDAR GPS locations to estimate the translation vector. Concurrently, the IMU data estimates the rotation matrix by analyzing the orientation changes. Both these estimates are then combined to form the initial transformation matrix. This matrix plays a crucial role in minimizing the search space during the manual registration phase and, by extension, reduces the computational burden, paving the way for a more accurate and efficient registration process.

The calculated initial transformation matrix is then utilized as the baseline for the manual registration of every 10th frame, significantly streamlining the registration process by providing a reasonably accurate initial pose estimate. This systematic approach ensures a coherent transition into the manual registration phase, laying a solid foundation for the subsequent steps in the registration pipeline.

4.1.4 Ground Truth Generation

To generate the vehicle pose ground truth for the collected and cleaned dataset, registration between the source point clouds (Vehicle) and the target point clouds (Infrastructure) needs to be computed and the transformation matrix will be the ground truth. Since the two point clouds do have two different Field of View (FoV) and low overlapping points, the traditional registration methods do not work as expected. Traditional registration methods such as RANSAC assume a huge overlapping between point clouds for the algorithm to work. To overcome this challenge, manual registration is utilized to produce highly accurate ground truth data. Furthermore, for manual registration, the software tool Blender[3] is employed due to its robust set of features for 3D model manipulation and precise alignment capabilities. The process begins by importing the point cloud data into Blender, where the transformation tools are used to align the vehicle and infrastructure point clouds accurately. This manual alignment serves as a reference or "ground truth" against which automated registration methods can be compared and evaluated.

In particular, every tenth frame is manually registered to tackle the issue of low overlap between the point clouds, which traditional registration methods struggle with. By selecting keyframes for manual registration, a foundation is laid for interpolating the transformations of the in-between frames. Two prominent interpolation methods, namely SLERP (Spherical Linear Interpolation) [con23] and Linear Interpolation, are employed to achieve smooth transitions between these keyframes. SLERP is utilized for interpolating rotations because it provides uniform rotational motion. At the same time, Linear Interpolation is employed for translating the position vectors due to its straightforward application and efficacy in representing linear motion. The SLERP equation is given by

$$\mathrm{Slerp}(q_0, q_1, t) = q_0(q_0^{-1} q_1)^t,$$

where q_0 and q_1 are the quaternions representing the rotations of the start and end frames, and t is a parameter varying between 0 and 1. On the other hand, Linear Interpolation for

[3]https://www.blender.org/

translation vectors is expressed as

$$\mathbf{T}(t) = \mathbf{T}_0 + t(\mathbf{T}_1 - \mathbf{T}_0),$$

where $\mathbf{T}_0$ and $\mathbf{T}_1$ are the translation vectors of the start and end frames respectively, and t is the same parameter as before.

This dual interpolation strategy ensures that the estimated transformations between frames are smooth and geometrically accurate, thus adhering closely to the vehicle's and infrastructure's actual movements over time. The careful process of manual registration in Blender, coupled with thoughtful interpolation, ensures high accuracy, providing a reliable ground truth dataset. This dataset is crucial for the subsequent evaluation and fine-tuning of registration algorithms.

4.2 Algorithm Design

This section delves into the various registration techniques employed to benchmark the Iterative Closest Point (ICP) methods in vehicle-infrastructure localization. The essence of the ICP process involves two primary sources of inputs at each timestamp: one derived from the vehicle and the other from the infrastructure. These inputs are fed into the ICP Registration algorithm, which iteratively refines the alignment between the point clouds to deduce the optimal pose corresponding to that specific timestamp. This process is delineated in Figure 4.5, showing the flow from data acquisition to the pose output through the ICP registration algorithm.

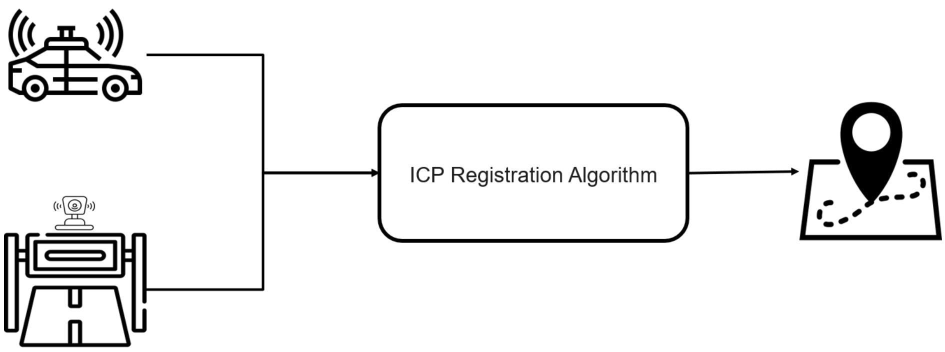


Figure 4.5: Process flow of ICP Registration from input data to pose output.

Each ICP method explored in this section is examined under two different setups: online registration and offline registration. In online registration, the vehicle point cloud at a specific timestamp is registered to the infrastructure point cloud at the same timestamp, thus mirroring a real-time operational scenario. Conversely, in offline registration, the vehicle point cloud at a specific timestamp is registered to the infrastructure local map, constant for all timestamps. The following subsections delve into the unique attributes and contributions of each ICP method within the context of vehicle-infrastructure localization.

4.2.1 Online Registration

Online registration involves registering online point cloud (pcd) frames with each other. This setup is essential for real-time or near-real-time applications where the data is being collected and processed concurrently. The source point clouds are the online scans of vehicle point clouds. Similarly, the target point clouds are the online scans of Infrastructure.

4.2.2 Offline Registration

Conversely, offline registration refers to the process of registering the vehicle frame with the infrastructure local map. This setup is more suited for scenarios where data is pre-collected, and the registration can be performed post data collection. The source point clouds are the online scans of vehicle point clouds. On the other hand, the target point clouds are the local map created out of the infrastructure point clouds.

4.2.3 Infrastructure Local Map Creation

The infrastructure local map serves as a critical component in our offline registration process, providing a reference geometry to which the point clouds collected from the vehicle can be aligned. The creation of this local map is carried out by leveraging the KISS-ICP algorithm.

Extraction Using Modified KISS-ICP A variant of the KISS-ICP (Keep It Small and Simple Iterative Closest Point) algorithm is crafted to facilitate the extraction of the infrastructure local map. Unlike the conventional KISS-ICP method, the adapted version is tailored to extract the created local map by KISS-ICP. This modification enables the usage of the created local map in other processes.

SLAM Approach in KISS-ICP The KISS-ICP algorithm is LiDAR Odometry, which employs mapping for localization, as detailed in Section 3.1.1. This mapping strategy ingrained in KISS-ICP ensures the precise alignment and fusion of individual point clouds, resulting in a meticulously curated infrastructure Local Map, pivotal for the subsequent registration stages.

4.2.4 Classical Point-to-Point ICP

In this subsection, we discuss the operational mechanics of the Point-to-Point ICP, along with a clarification of the specific aspects in which it was employed in this study.

One of the crucial research papers in point cloud registration, [BM92] introduces Point-to-Point (P2P) ICP. The paper outlines a versatile, representation-agnostic technique for precisely and computationally efficiently registering 3D forms, encompassing free-form curves and surfaces. This method has been utilized later as a foundational approach in point cloud registration.

Classical Point-to-Point ICP Summary

1. **Initialization**:
   - Assume we have two point sets $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$.
   - Initially, a rough alignment may be given or some method to provide a rough alignment is employed.
2. **Closest Point Finding**:

- For each point p_i in set P, find the closest point q_j in set Q.

$$\text{For each } i, \text{ find } j \text{ such that } \|p_i - q_j\| \text{ is minimized}$$

3. **Optimal Transformation Computation**:

   - Compute the optimal transformation T (comprising rotation and translation) that minimizes the objective function $E(T)$ over all pairs of points (p,q) in the correspondence set C:

   $$E(T) = \sum_{(p,q)\in C} \|p - Tq\|^2$$

   This minimizes the squared distance between each point in P and the transformed version of its corresponding point in Q.

4. **Transformation Application**:

   - Apply the computed transformation T to all points in Q to bring them into alignment with P:

   $$Q' = TQ$$

   Q' is the transformed point set.

5. **Convergence Check**:

   - Check if the difference in error $E(T)$ (or the transformation) is below a certain threshold, or if a maximum number of iterations has been reached.
   - If not, repeat steps 2 to 5.

The algorithm iteratively performs steps 2 to 5, gradually refining the alignment between the point sets by minimizing $E(T)$ in each iteration, until a well-aligned configuration is achieved or the process reaches a termination condition such as a maximum number of iterations.

The implementation of the P2P ICP algorithm was carried out utilizing the Open3D library [ZPK18]. Open3D is an open-source library that provides a convenient set of tools for processing 3D data. The library includes an implementation of the ICP algorithm, which was leveraged in this work to perform the point cloud registration. The utilization of Open3D facilitated a streamlined development process while ensuring the accuracy and efficiency of the registration tasks performed.

Online Registration Implementation

This paragraph elaborates on the implementation details of online registration using the Classical Point-to-Point Iterative Closest Point (P2P ICP) method for vehicle-infrastructure localization. As outlined in Section 4.2.1, the source and target point clouds for online registration are live scans from each timestamp. The method for online registration using Classical P2P ICP is straightforward and repetitive. It ensures the vehicle's point cloud is accurately registered with the infrastructure's point cloud. Algorithm 2 outlines the step-by-step procedure for each timestamp in the online registration process. The process begins by reading the point clouds from the vehicle and infrastructure based on the same timestamp. Following this, the data is made more manageable through downsampling, and then the necessary GPS and IMU data are gathered. With this information, an initial transformation matrix is calculated. Finally, the Classical P2P ICP method is applied to precisely register the vehicle's point cloud to the infrastructure. This direct approach is essential to ensure successful registration, which improves the vehicle's localization in an environment that constantly changes.

Algorithm 2 Classical P2P ICP for Online Registration

1: **Input:** Timestamps T, Vehicle point clouds P_{veh}, Infrastructure point clouds P_{infra}, Vehicle GPS & IMU data $D_{veh\_gps\_imu}$, Known infrastructure GPS & IMU data $D_{infra\_gps\_imu}$
2: **Output:** Registered point clouds
3: **for** each timestamp $t \in T$ **do**
4: Read $P_{veh}(t)$ & $P_{infra}(t)$ with the same timestamp t
5: Downsample $P_{veh}(t)$ & $P_{infra}(t)$ using FPDH
6: Read Vehicle GPS & IMU data $D_{veh\_gps\_imu}(t)$ for timestamp t
7: Compute initial transformation matrix $\mathbf{T}_0$ using $D_{veh\_gps\_imu}(t)$ and known $D_{infra\_gps\_imu}$
8: Transform $P_{veh}(t)$ using $\mathbf{T}_0$ to get $P'_{veh}(t)$
9: Apply Classical P2P ICP to register $P'_{veh}(t)$ to $P_{infra}(t)$
10: **end for**

Offline Registration Implementation

This paragraph details the process of offline registration using Classical Point-to-Point Iterative Closest Point (P2P ICP) for vehicle-infrastructure localization. Unlike online registration, offline registration utilizes the infrastructure's local map as the target point cloud for registration, as discussed in Section 4.2.2.

The offline registration approach mirrors its online version's systematic and iterative nature. It ensures that the vehicle's point cloud is precisely registered to the static local map of the infrastructure. The offline Classical P2P ICP process corresponds to the flow diagram presented in Figure 4.6, which visualizes the registration process for a single timestamp. The registration begins by reading the infrastructure local map, which is constant for all timestamps. Then, it is streamlined via downsampling techniques such as FPDH to optimize the subsequent processing steps. The vehicle's point cloud is similarly refined and aligned for each discrete timestamp through a calculated initial transformation matrix, informed by precise GPS and IMU data. The Classical P2P ICP method is then applied, registering the vehicle point cloud to the infrastructure local map.

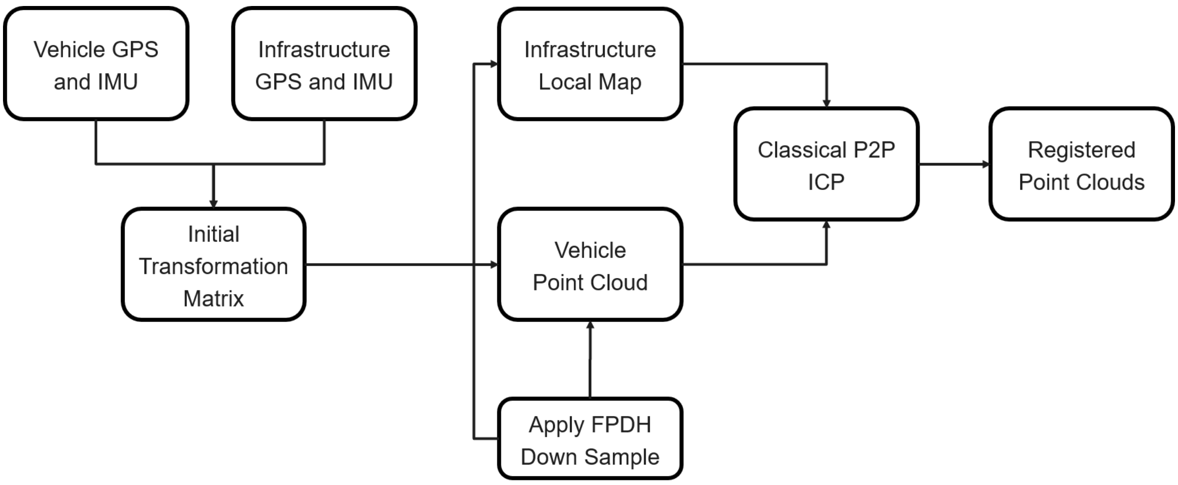


Figure 4.6: Process flow of offline Classical P2P ICP for a specific timestamp.

This completes explaining the Classical P2P ICP method for vehicle-infrastructure localization in online and offline contexts. The following sections will introduce the implementation of the KISS-ICP method for vehicle-infrastructure localization.

4.2.5 KISS ICP

As delineated in Section 3.1.1, KISS-ICP emerges as a simplistic yet highly effective method suited for vehicle localization tasks [Viz+23]. To adapt it for vehicle-infrastructure localization, a modified version of KISS-ICP was necessitated. This adapted version primarily accommodates an additional input for registration, which in this case is the infrastructure point clouds or infrastructure local map. The selection of KISS-ICP as a robust ICP registration technique for this investigation is underpinned by several reasons:

- The implementation of KISS-ICP is accessible for researchers, promoting open-source engagement[4].
- KISS-ICP exhibits operational resilience with minimal parameter tuning across a diverse range of LiDAR data [Viz+23].
- Encapsulated in Python, KISS-ICP provides a flexible environment for any necessary modifications.
- A state-of-the-art (SOTA) localization method, KISS-ICP is a robust benchmark for evaluating current LiDAR Odometry for vehicle-infrastructure localization performance.

Online Registration Implementation

Using KISS-ICP for online registration in vehicle-infrastructure localization needs adaptation of the original KISS-ICP, as stated before. While closely mirroring the Classical P2P ICP process outlined in Section 4.2.4, KISS-ICP introduces specific modifications essential for precise localization. The procedure is encapsulated in Algorithm 3

Algorithm 3 KISS-ICP for Online Registration

Input: Timestamps T, Vehicle point clouds P_{veh}, Infrastructure point clouds P_{infra}, Vehicle GPS & IMU data $D_{veh\_gps\_imu}$, Known infrastructure GPS & IMU data $D_{infra\_gps\_imu}$
Output: Registered point clouds
for each timestamp $t \in T$ **do**
  Read $P_{veh}(t)$ & $P_{infra}(t)$ with the same timestamp t
  Apply sensor motion compensation (deskewing) to $P_{veh}(t)$
  Obtain sensor motion prediction and adaptive threshold for both $P_{veh}(t)$ & $P_{infra}(t)$
  Subsample $P_{veh}(t)$ & $P_{infra}(t)$
  Read Vehicle GPS & IMU data $D_{veh\_gps\_imu}(t)$ for timestamp t
  Compute initial transformation matrix $\mathbf{T}_0$ using $D_{veh\_gps\_imu}(t)$ and known $D_{infra\_gps\_imu}$
  Transform $P_{veh}(t)$ using $\mathbf{T}_0$ to get $P'_{veh}(t)$
  Apply KISS-ICP to register $P'_{veh}(t)$ to $P_{infra}(t)$
end for

Algorithm 3 presents a structured and detailed view of the online registration process using KISS-ICP, ensuring effective integration of the vehicle's dynamic point clouds with the static infrastructure data. Including sensor motion compensation (deskewing) and adaptive thresholding distinguishes KISS-ICP from Classical P2P ICP, facilitating a more refined registration essential for dynamic environments.

[4] https://github.com/PRBonn/kiss-icp

Offline Registration Implementation

In this section, we describe the procedure for offline registration using the adapted KISS-ICP method for vehicle-infrastructure localization. Similar to its online counterpart, the offline implementation of KISS-ICP involves specific adaptations, particularly in handling the local map of the infrastructure. The KISS-ICP offline process, depicted in Figure 4.7, begins by utilizing the previously generated and stored local map of the infrastructure. This static map serves as the target point cloud for registration.

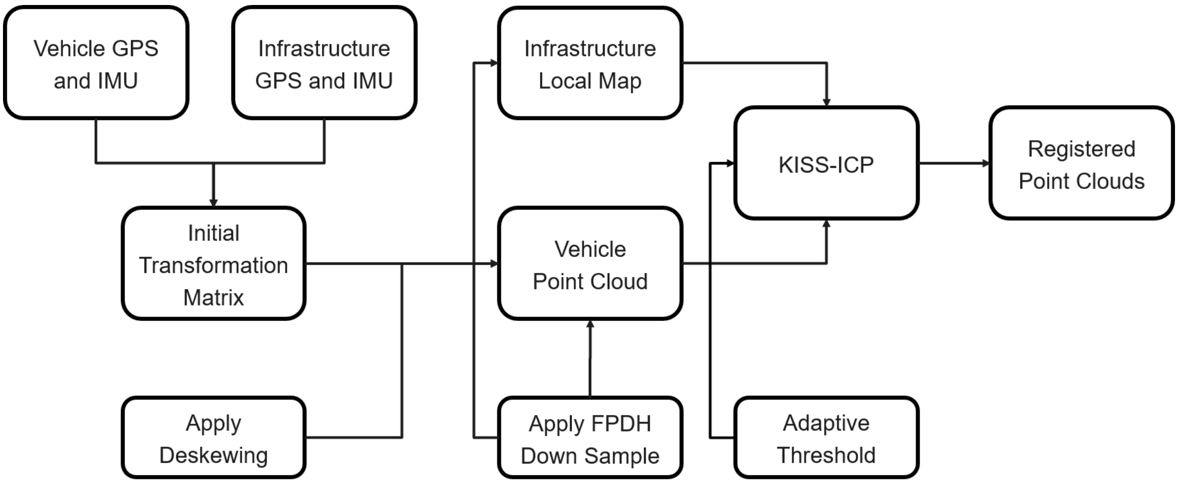


Figure 4.7: Process flow of offline KISS-ICP registration for a specific timestamp.

The registration process incorporates sensor motion compensation (Deskewing) and adaptive thresholding to refine the vehicle point cloud, followed by a subsampling step for efficiency. Crucial to this process is the creation of an initial transformation matrix, which relies on the vehicle's GPS and IMU data. This matrix provides an essential initial guess to align the vehicle point cloud relative to the infrastructure local map. The KISS-ICP algorithm is then employed to register the vehicle point cloud to this local map.

Notably, the offline version of KISS-ICP diverges from the original implementation in its use of the infrastructure local map and the absence of local map updates. The GPS and IMU data's role in generating the initial transformation matrix is also a key aspect of this adaptation.

Finally, we conclude our discussion on the implementation of KISS-ICP methods for vehicle-infrastructure localization in both online and offline settings. In the following sections, we shed light on MULLS and MULLS-ICP.

4.2.6 MULLS

In this section, we delve into how MULLS & MULLS-ICP is utilized in this research. As elaborated in Section 3.1.1, MULLS authors main contributions are MULLS and MULLS-ICP [Pan+21]. MULLS is a scan-line independent LiDAR-only SLAM solution. On the other hand, MULLS-ICP is a point cloud local registration algorithm. MULLS contribution, in general, is a versatile localization method comprising two components: the front-end and the back-end [Pan+21]. Moreover, having been tested against three datasets, MULLS emerges as a strong candidate for vehicle-infrastructure localization benchmarking.

Online Registration Implementation

For online registration, we followed the original implementation of MULLS-ICP. Where MULLS-ICP, as demonstrated in the case study by Yue Pan [Pan], is using neighborhood category context (NCC) & TEASER++ first, then refine the registration using MULLS-ICP [YSC20]. First, the geometric feature points are extracted using dual-threshold ground filtering and non-ground points classification based on PCA. Subsequently, NCC determines the initial correspondences according to the cosine similarity among vertex keypoints. Further, TEASER++ takes the correspondences from NCC to estimate the initial pose guess. This essentially corresponds to the front-end component depicted in Figure 3.3. MULL-ICP sequential steps are captured in Algorithm 4.

Algorithm 4 MULLS-ICP for Online Registration

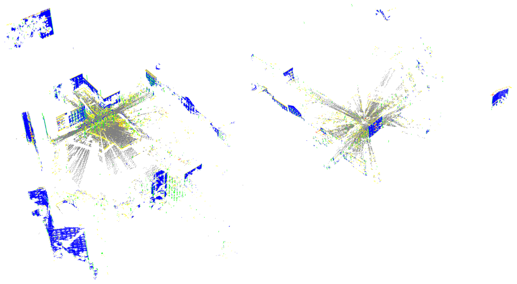
1: **Input:** Timestamps T, Transformed Vehicle point clouds P'_{veh}, Infrastructure point clouds P_{infra}, Vehicle GPS & IMU data $D_{veh\_gps\_imu}$, Known infrastructure GPS & IMU data $D_{infra\_gps\_imu}$
2: **Output:** Registered point clouds
3: **for** each timestamp $t \in T$ **do**
4: Read $P'_{veh}(t)$ & $P_{infra}(t)$ with the same timestamp t
5: Extract geometric features from $P'_{veh}(t)$ & $P_{infra}(t)$
6: Apply NCC to keypoints and determine correspondences
7: Apply TEASER++ based on these correspondences
8: Refine the registration results using MULLS-ICP
9: **end for**

Moreover, MULLS-ICP steps are visually depicted in Figure 4.8, which shows in Figure 4.8a the geometric feature points in target and source point clouds [YSC20]. Then, in Figure 4.8b, the correspondences are determined by NCC, and the registration results of apply TEASER [YSC20]. Further, the fine-tuning using MULLS-ICP is depicted in Figure 4.8c. MULLS-ICP offers a structured approach to integrating dynamic vehicle point clouds with static infrastructure data for effective online registration.

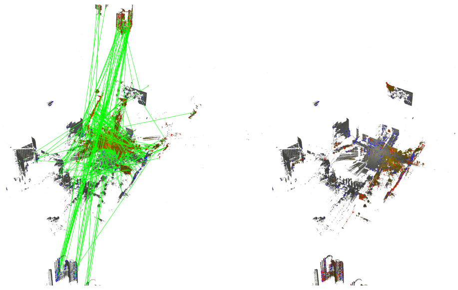
Offline Registration Implementation

For offline registration, MULLS SLAM is used out of the box. No modification needed to be done. Thus, as described in more detail in Section 3.1.1, MULLS SLAM is divided into two parts: Front-end and back-end. For the front end feature points such as ground, facade, pillar, and beam, using a combination of dual-threshold ground filtering and principal component analysis. Then the registration process between the current frame and the local submap is executed efficiently via a multi-metric linear least square iterative closest point algorithm. Point-to-point (plane, line) error metrics within each point class are jointly optimized with a linear approximation to estimate the ego motion. Static feature points of the registered frame are appended to the local map to keep it updated. For the back end, hierarchical pose graph optimization is conducted among regularly stored history submaps to reduce the drift resulting from the dead reckoning.

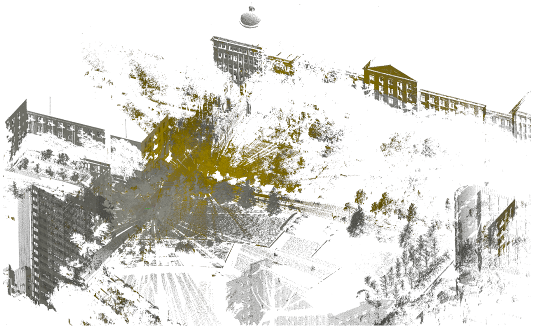
In the offline registration process, MULLS SLAM was directly applied without necessitating any modifications. As delineated in Section 3.1.1, MULLS SLAM fundamentally splits its operations into two primary components, as shown in Figure 3.3: the front-end and the back-end.



(a) Geometric feature points in target (left) and source (right) point cloud: (gray: ground, blue: facade, green: pillar, yellow: beam, red: roof, purple: vertex keypoint) [Pan].



(b) Left: correspondences determined by NCC feature of the keypoints; Right: registration result of TEASER [Pan].



(c) Source and target point cloud after TEASER and MULLS-ICP fine-tune [Pan].

Figure 4.8: A visual representation of MULLS-ICP's steps [Pan].

Front-end: The front-end process initiates with the extraction of broadly categorized feature points (including ground, facade, pillar, beam, and others) from each frame. This extraction is facilitated by a combination of dual-threshold ground filtering techniques and principal components analysis as in the above online registration section. Following this, the algorithm establishes the registration between the current frame and a local submap. This registration is efficiently achieved via the newly proposed multi-metric linear least square iterative closest point algorithm, wherein error metrics corresponding to point-to-point (plane, line) within each point classification are concurrently optimized. This uses a linear approximation to discern ego motion. After successful registration, static feature points from the frame are incorporated into the local map, ensuring its continuous update.

Back-end: The back-end process focuses on the hierarchical optimization of the pose graph. This is done across systematically stored historical submaps. The aim here is to mitigate any drift that might arise from dead reckoning.

Chapter 5

Experiments

This chapter delineates the experimental procedure and comprehensively presents the results obtained. Initially, we explain the experimental setup. Subsequently, a detailed examination of each method's performance in online and offline registration is undertaken. Concluding the chapter with a comparison of the results of different ICP methods for both online and offline setups

5.1 Experimental Setup

In the context of vehicle-infrastructure localization experiments, various factors come into play. In this section, we first delve into the specifics of the equipment utilized for data collection. Subsequently, we show the driving path and ground truth trajectory for our experiments and infrastructure local map. We conclude this section by introducing the evaluation metrics employed to assess the performance of each ICP method.

5.1.1 Sensors

For the collection of the dataset, two LiDAR systems were employed. A dynamic RoboSense RS-32 LiDAR was affixed atop a vehicle, as depicted in Figure 5.1a. Concurrently, a static Ouster OS1-64 LiDAR was positioned atop Providentia++ infrastructure, illustrated in Figure 5.1b. In addition to LiDAR, data was also gathered from GPS and IMU systems to calculate the vehicle's initial pose estimate. Specifically, the RTK EMLID REACH RS2+ GPS, shown in Figure 5.1c, was positioned 80 cm from the RoboSense LiDAR at the center of the car's roof. The Xsens MTi-30 IMU, presented in Figure 5.1d, and the Basler ace acA1920-48gc camera were subsequently mounted in front of the static Ouster LiDAR. All the aforementioned sensors operated at a frequency of 10 Hz.

5.1.2 Driving Path and Ground Truth Trajectory

Figures 5.2 & 5.3 present both the real-life view and the plotted trajectory of the taken path. This trajectory, spanning approximately 140 meters, serves as the ground truth for all subsequent experiments. The driving speed during this trajectory collection averaged between 20 to 30 km/hour, reflecting typical urban driving conditions and ensuring the captured data's fidelity under such scenarios.

The driving path involved traveling in a straight line up to the intersection where the Ouster LiDAR was mounted atop the infrastructure. Upon reaching the intersection, the

(a) RoboSense RS-32 LiDAR [Rob].

(b) Ouster OS1-64 LiDAR [Ous].

(c) RTK EMLID REACH RS2+ GPS [Rea].

(d) Xsens MTi-30 IMU [Xse].

Figure 5.1: Sensor setup for dataset collection

vehicle waited for the traffic light to turn green. Afterward, the vehicle continued its course, driving straight through and beyond the intersection.

5.1.3 Local Map

The infrastructure local map depicted in Figure 5.4 was created using KISS-ICP. Moreover, for all offline experiments, this local map is the target point cloud. The figure shows, in the middle, road poles on four different roads (colored in Red), a trail of a moving bus, buildings on the right side, and a big road sign on the left side. During the experiments, the car drives from right to left through the infrastructure.

5.1.4 Evaluation Metrics

The evaluation metrics for vehicle-localization benchmarking will follow the state-of-the-art (SOTA) evaluation metrics used for odometry and SLAM methods. Furthermore, these metrics are encapsulated in a Python package named EVO, provided by Michael Grupp: 'Python package for the evaluation of odometry and SLAM' [Gru17]. The following metrics will be used in the evaluation:

- **Absolute Pose Error**: Evaluates the absolute pose error, often referred to as absolute trajectory error. It tests the global consistency of a trajectory by comparing corresponding poses directly between an estimate and a reference. APE is based on the absolute relative pose between two poses $P_{\text{ref},i}^{-1} P_{\text{est},i} \in \mathrm{SE}(3)$ at timestamp i:

$$E_i = P_{\text{est},i} \ominus P_{\text{ref},i} = P_{\text{ref},i}^{-1} P_{\text{est},i} \in \mathrm{SE}(3) \tag{5.1}$$

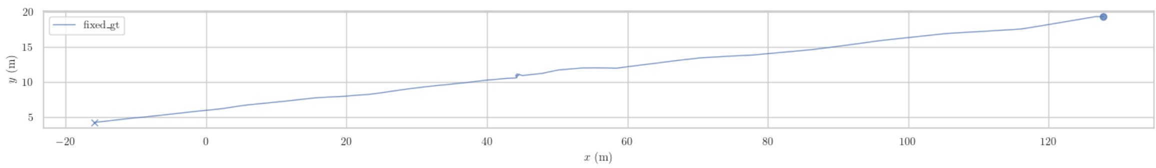


Figure 5.2: The trajectory plot of the ground truth.

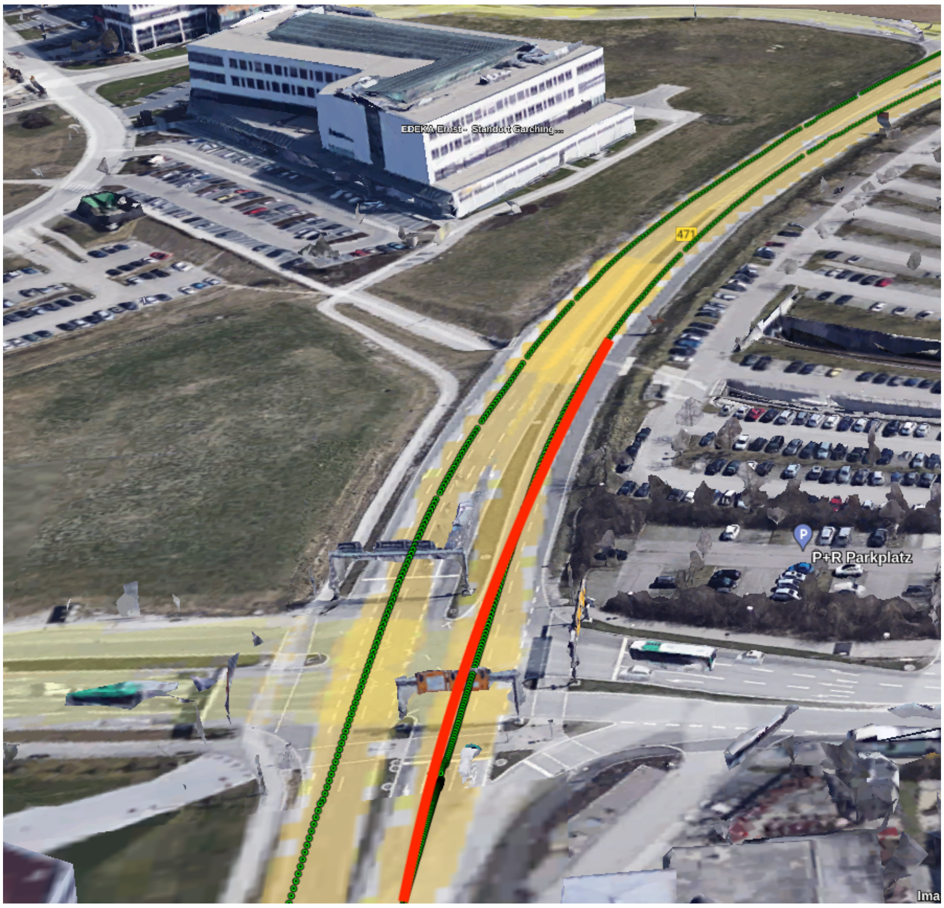


Figure 5.3: Real-life view of the ground truth trajectory.

Where $\ominus$ is the inverse compositional operator, which takes two poses and gives the relative pose [LM97]. Different pose relations to calculate the APE can be used: Translation, Rotation, or full Transformation relations. For instance, Equation 5.2 shows the Transformation APE.

$$APE_{\text{full},i} = ||E_i - I_{4\times4}||_F \tag{5.2}$$

In this context, E_i represents the error transformation matrix at timestamp i. The Frobe-

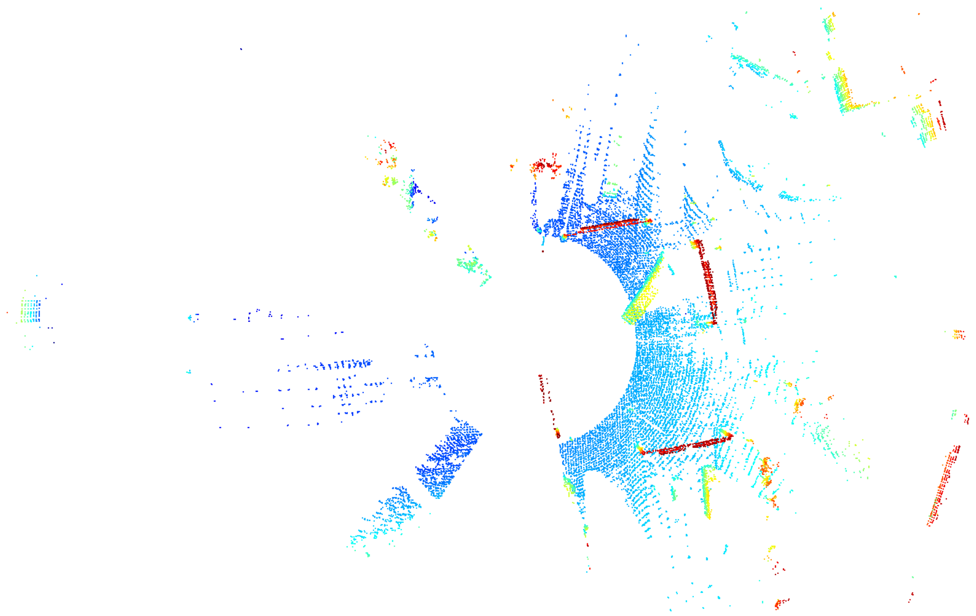


Figure 5.4: Infrastructure Local Map created by using KISS-ICP. It is the target point cloud in all offline experiments.

nius norm, denoted by $\|\cdot\|_F$, quantifies the magnitude of the error matrix. This computation reflects the total error in the estimated pose, considering both rotational and translational components, providing a comprehensive measure of the SLAM and Odometry system's accuracy.

- **Relative Pose Error**: Assesses the relative pose error by comparing motions or "pose deltas". This metric provides insights into local accuracy or drift, such as translational or rotational drift per meter. The relative pose error for poses between timestamps i and j is defined as:

$$E_{i,j} = \delta_{\text{est},i,j} \ominus \delta_{\text{ref},i,j} = (P_{\text{ref},i}^{-1} P_{\text{ref},j})^{-1} (P_{\text{est},i}^{-1} P_{\text{est},j}) \in \text{SE}(3) \tag{5.3}$$

Here, $\ominus$ denotes the inverse compositional operator, and $\delta_{\text{est},i,j}$ and $\delta_{\text{ref},i,j}$ are the relative poses in the estimated and reference trajectories, respectively. Various metrics can be derived from $E_{i,j}$ such as translation, rotation, or the full transformation. The RPE for full transformations is given by:

$$RPE_{\text{full},i,j} = \|E_{i,j} - I_{4\times 4}\|_F \tag{5.4}$$

This metric examines the fidelity of pose estimation over short trajectory segments and can be helpful in evaluating the incremental performance of odometry systems.

Further, the following visualization methods will be used in the evaluation of the vehicle-infrastructure localization:

- **Trajectories Plotting**: Plot trajectories against the ground truth to provide qualitative results for different methods.

- **Results Comparison**: Compare outcomes from various methods rather than comparing the results of a single method to the ground truth. This has the advantage of allowing a comparison between the results of SLAM algorithm X to another algorithm Y.

5.2 Classical Point-to-Point ICP

The Classical Point-to-Point ICP method is explored in its Online and Offline localization variants. Each variant encompasses One critical experimental variable: Correspondence threshold. The correspondence threshold is the maximum correspondence points-pair distance (measured in meters)

5.2.1 Online Registration

This subsection unveils the results of the online registration experiment. As previously highlighted, this experiment varied in one input, the correspondence threshold. Initially, we present the outcomes of the Online P2P registration, evaluated across three distinct correspondence thresholds (1, 3, 5).

Correspondence threshold 1: The trajectory plot presented in Figure 5.5 depicts the estimated trajectory with the ground truth trajectory. Upon close observation, The estimated poses of the vehicle are slightly off the ground truth trajectory. During these first 50 meters, the car is still far away from the intersection, hence sparse infrastructure point clouds. Furthermore, as the vehicle gets closer to the intersection, the two trajectories exhibit a strong overlap, implying that the vehicle's localization was largely accurate. However, there are regions, especially at the end of the drive, where slight deviations are noticeable.

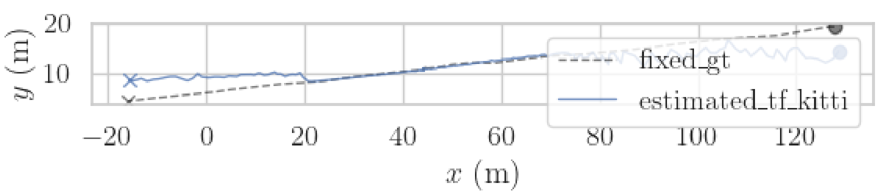


Figure 5.5: Trajectory plot for P2P Online Registration with ground with correspondence threshold 1.

The Absolute Pose Error (APE) analysis comprehensively assesses the alignment between the estimated and ground truth trajectories. Observing the full APE, APE w.r.t full transformation, presented in Figure 5.6, we notice that the full APE is relatively high at the start and the end of the drive compared to the timestamps in between. Moreover, the full APE presents a Root Mean Square Error (RMSE) of 1.478 unit-less. Furthermore, Delving deeper, the specific components of the error can be elucidated through the translation and rotation APEs depicted in Figure 5.7, where both of the translation and rotation APE follows the same pattern as the full APE however, the translation APE has a higher error value where it peaked at 7.137 meters error.

An in-depth qualitative analysis of the registered point clouds using Classical Point-to-Point ICP in online setup is provided in Figure 5.14. The results showcase the registration of the vehicle point cloud to the Infrastructure point cloud at timestamps 50 and 450. These illustrations facilitate a clearer understanding of the alignment proficiency achieved by the Classical Point-to-Point ICP method. At timestamp 50, there's a subtle shift between the two point clouds, particularly noticeable in buildings and other static landmarks, likely because the vehicle is still at a significant distance from the infrastructure. By timestamp 450, there's a pronounced overlap between the point clouds. However, upon closer inspection, a slight forward shift in the vehicle's point cloud prevents a perfect alignment.

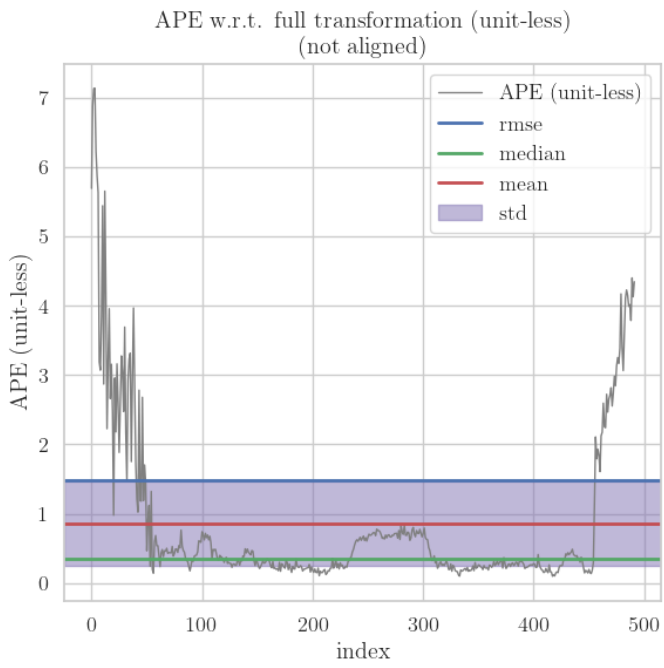
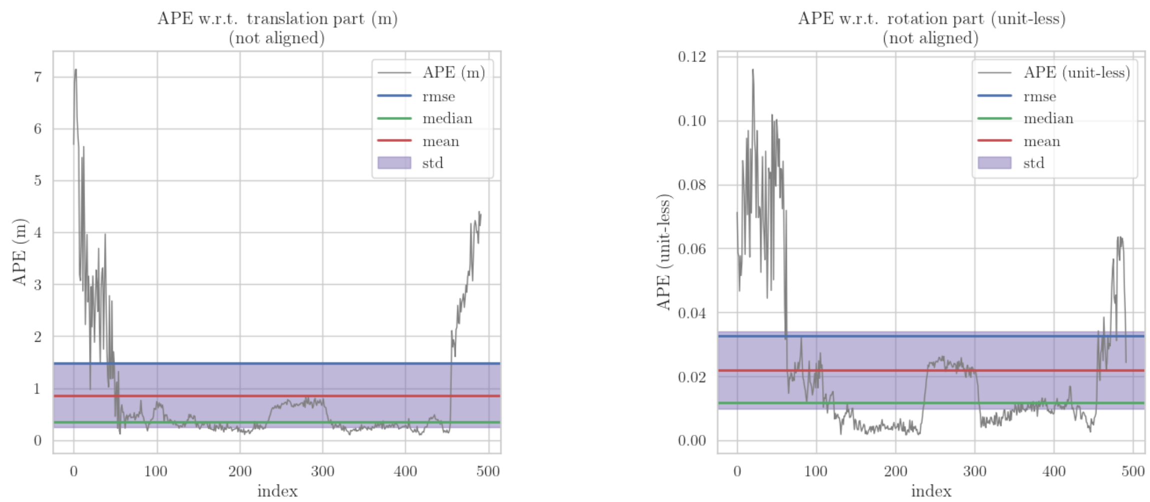


Figure 5.6: APE for P2P Online Registration with ground against the full transformation for correspondence threshold 1.



(a) Translation APE plot.

(b) Rotation APE plot.

Figure 5.7: APE plots for P2P Online Registration with ground against translation and rotation for correspondence threshold 1.

Correspondence threshold 3: Furthermore, experiments with correspondences threshold 3 are presented where the first plot is the trajectory plot, which is given in Figure 5.8. The figure shows the estimated trajectory with the ground truth trajectory. Upon close observation, The estimated poses of the vehicle are highly inaccurate compared to the ground truth trajectory during the first 30 meters, where the vehicle is still far away from the intersection, hence sparse infrastructure point clouds. However, the two trajectories align more closely as the vehicle approaches the intersection, suggesting a largely accurate vehicle localization. Notably, in contrast to the correspondence threshold 1 experiment, this close alignment between the two trajectories persists until the end of the test.

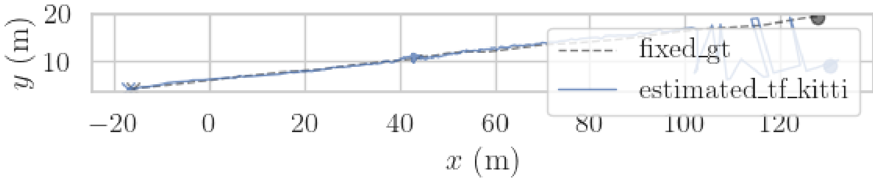


Figure 5.8: Trajectory plot for P2P Online Registration with ground with correspondence threshold 3.

The Absolute Pose Error (APE) analysis also reflects what has been presented in the trajectory plot. Observing the full APE, shown in Figure 5.10, we notice that the error is noticeably high for the first 30 timestamps then the error decreases to remain under 2.5 for the rest of the experiment until the final 50 frames the error is very close to zero. Furthermore, the translation and rotation APEs are depicted in Figure 5.11a and Figure 5.11b respectively.

Given that the RMSE for P2P online experiment with the correspondence threshold 3 (2.403) is higher than the RMSE for the P2P online experiment with correspondence threshold 1 (1.478), the qualitative results & the trajectory plot indicate otherwise as depicted in Figure 5.15, the visualization of the registered point clouds at the 50th timestamp shows that the point clouds have a better alignment than the one in the previous experiment (correspondences threshold 1). Further, the qualitative result at timestamp 450 shows the point clouds are aligned with no noticeable deviations.

Correspondence threshold 5: Following, the results of the P2P online registration with correspondences threshold 5 experiment are presented next. Setting the correspondence threshold to 5 means that the P2P Online Registration has more freedom to choose more Correspondences, where the number of correspondences ranged from 10000 to 30000 for most timestamps.

The trajectory plot presented in Figure 5.9 depicts at first look, that the estimated trajectory is similar to the previous experiment (correspondences threshold 3). However, upon closer look, the estimated trajectory has sharp edges due to a small deviation in contrast to the previous experiment, where the plot is smoother. Furthermore, the estimated trajectory for correspondence threshold 5 aligns earlier than any other experiment after only 20 meters of driving.

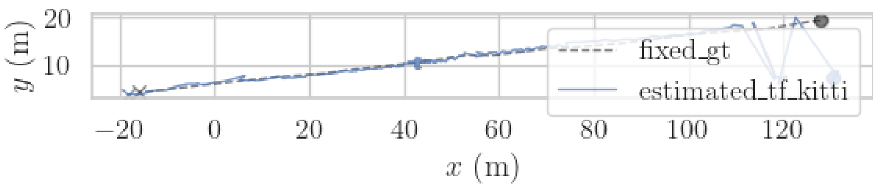


Figure 5.9: Trajectory plot for P2P Online Registration with ground with correspondence threshold 5.

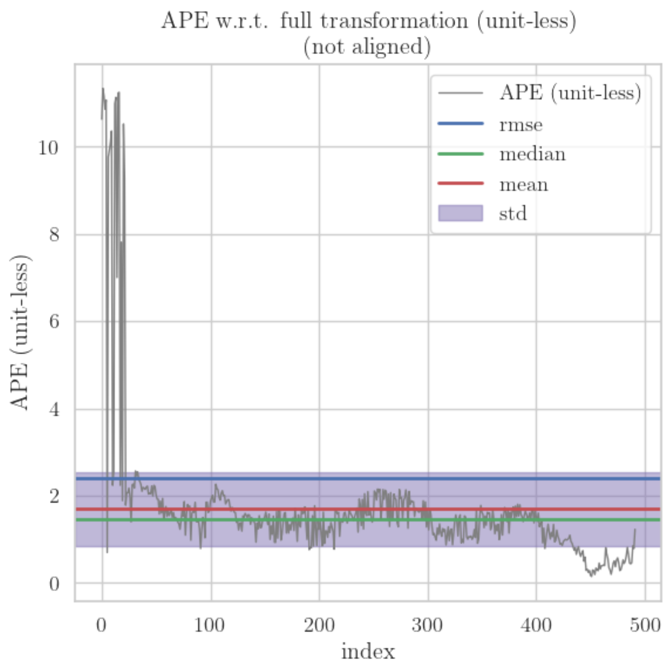
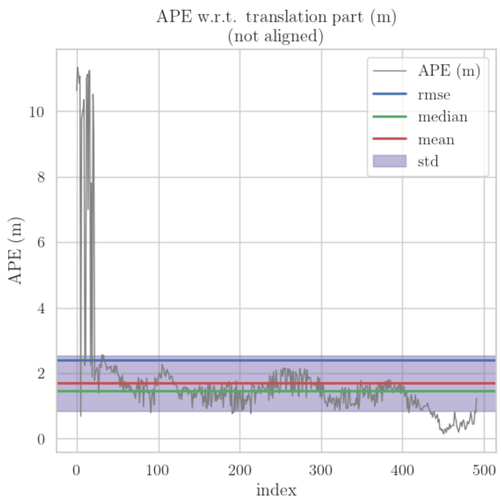
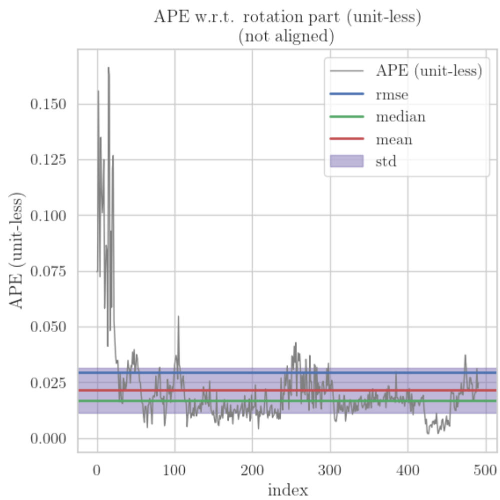


Figure 5.10: APE for P2P Online Registration with ground against the full transformation for correspondence threshold 3.



(a) Translation APE plot.



(b) Rotation APE plot.

Figure 5.11: APE plots for P2P Online Registration with ground against translation and rotation for correspondence threshold 3.

The APE analysis, presented in Figure 5.12 here shows similar behavior to the previous experiment (correspondence threshold 3) with small differences:

- Full APE analysis peaked at 13.768.
- At the end of the experiment, the translation APE had a higher value (≈ 4) than the RMSE (2.611), which is also illustrated in the trajectory plot.

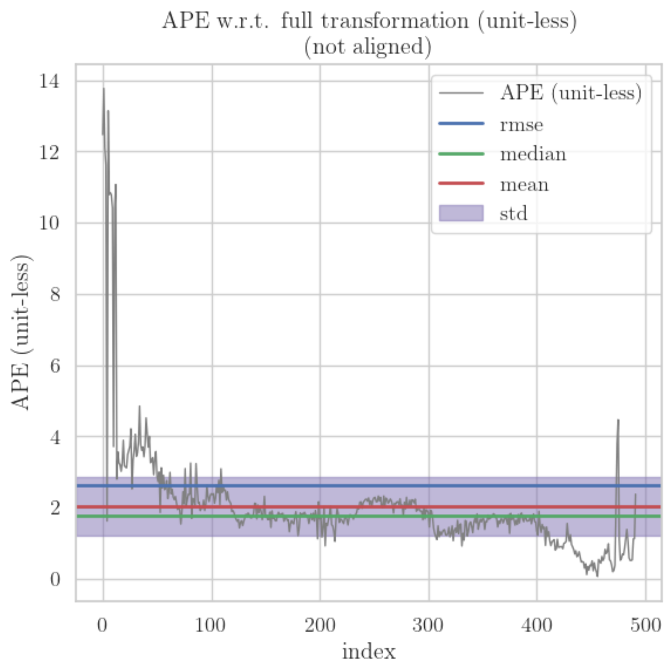
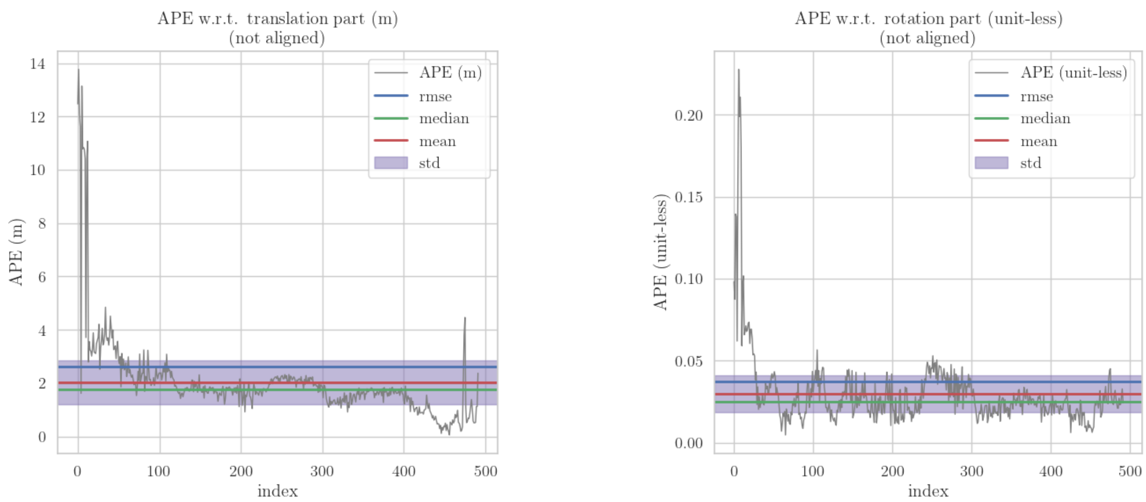


Figure 5.12: APE for P2P Online Registration with ground against the full transformation for Threshold 5.



(a) Translation APE plot.

(b) Rotation APE plot.

Figure 5.13: APE plots for P2P Online Registration with ground against translation and rotation for correspondence threshold 5.

The qualitative analysis in Figure 5.16 illustrates good registration for the two timestamps however, the vehicle point cloud shows a slight shift to the right by looking at the static traffic sign on the far left of the visualization.

5.2.2 Offline Registration

This subsection unveils the results of the offline registration experiment. As previously highlighted, this experiment varied two inputs: ground and threshold. Initially, we present the

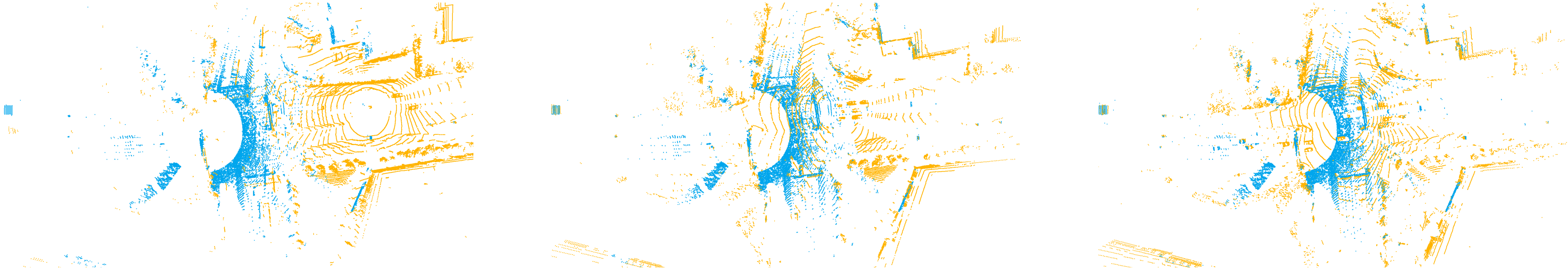


Figure 5.14: P2P Online Registration with ground Qualitative results for correspondence threshold 1 at timestamps 50, 250, and 450.

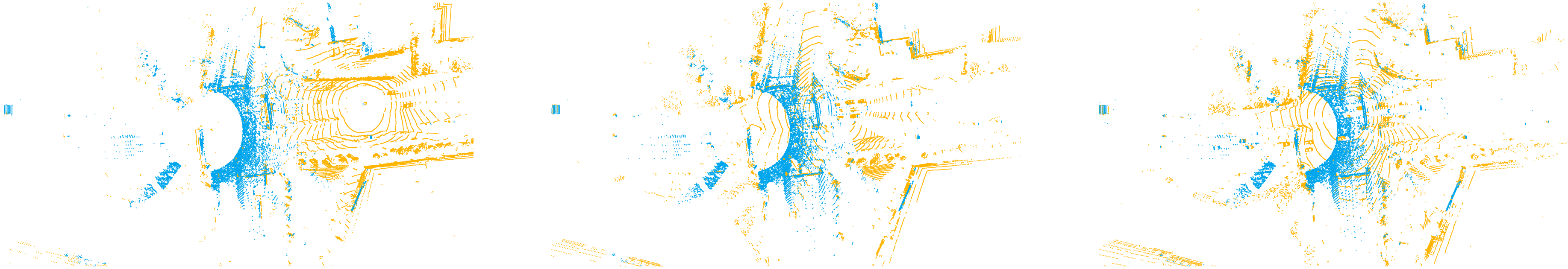


Figure 5.15: P2P Online Registration with ground Qualitative results for correspondence threshold 3 at timestamps 50, 250, and 450.

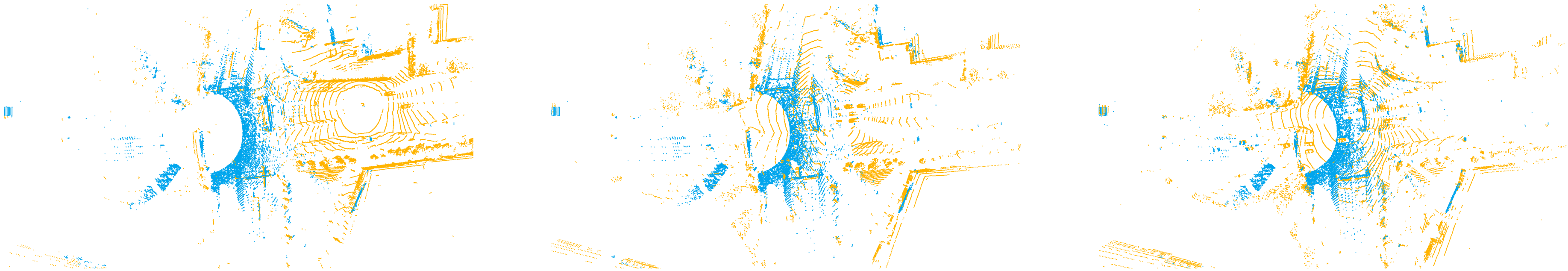


Figure 5.16: P2P Online Registration with ground Qualitative results for correspondence threshold 5 at timestamps 50, 250, and 450.

outcomes of the Offline P2P registration without the removal of the ground, evaluated across three distinct correspondence thresholds (1, 3, 5). As mentioned in Section 5.1.3, the local map is generated by the infrastructure point clouds using KISS-ICP. Furthermore, Figure 5.4 shows the infrastructure local map that would be used as a target for the Offline P2P registration.

Correspondence threshold 1: In this paragraph, we show the results for Offline P2P registration with correspondence threshold set to 1. Firstly, the analysis of the trajectory plot of Offline P2P registration correspondence threshold 1 presented in Figure 5.17, is very similar to the trajectory plot of Online P2P registration with correspondence threshold 1 in Section 5.2.1, However, with a small difference, the estimated trajectory from Offline P2P has more deviation than the Online P2P registration with the same correspondence threshold. Furthermore, the analysis of the full, translation, and rotation APEs, depicted in Figure 5.19, 5.20a, and 5.20b respectively, shows similar pattern to the online version of this experiment, demonstrated in Section 5.2.1, however with small differences like the maximum value of the full APE is 7.604 instead of 7.137.

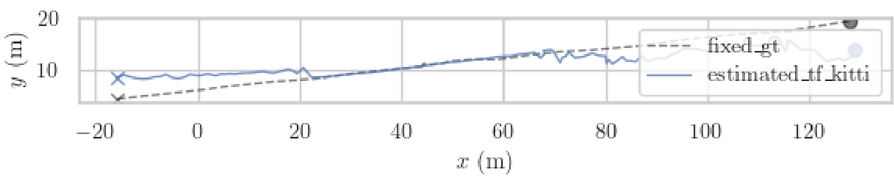


Figure 5.17: Trajectory plot for P2P Offline Registration with ground for correspondence threshold 1.

Figure 5.26 depicts the visualization of the registered point clouds using Offline P2P registration with correspondences threshold 1. Moreover, the visualization shows a big deviation between the vehicle point clouds, colored in yellow, and the infrastructure local map, colored in blue, this was also reflected in the trajectory plot and the APEs plots. However, at timestamp 450, the points clouds align with a high degree of accuracy.

Correspondence threshold 3: Unlike the previous experiment, Offline P2P with correspondence threshold 1, the results of this experiment, Offline P2P with correspondence threshold 3, is different than its online version. The trajectory plot in Figure 5.18 starts with high fluctuations between highly accurate estimations and poor estimations for approximately the first 30 meters. Subsequently, the trajectories align with high accuracy for approximately 15 meters, and then the fluctuations begin again for a short distance. Further, for the rest of the experiments, the estimated trajectory aligns with the ground truth trajectory however, with multiple small deviations, the estimated trajectory is not smooth, with a lot of sharp edges, a sign of instability and inconsistency.

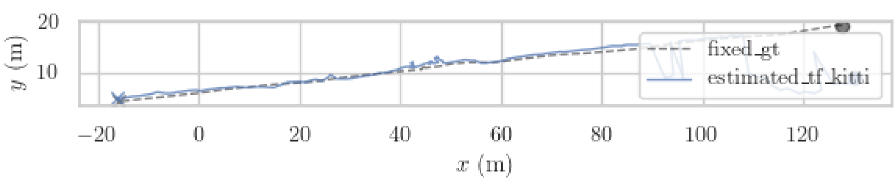


Figure 5.18: Trajectory plot for P2P offline Registration with ground with correspondence threshold 3.

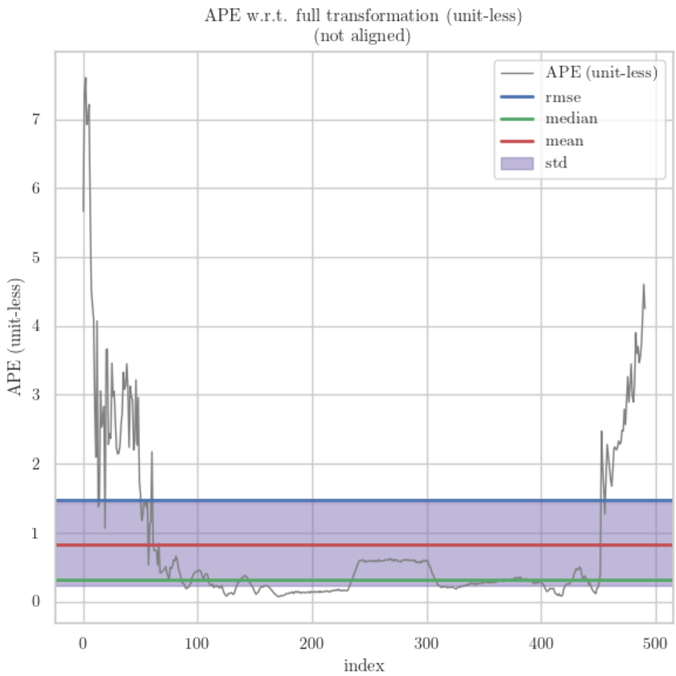
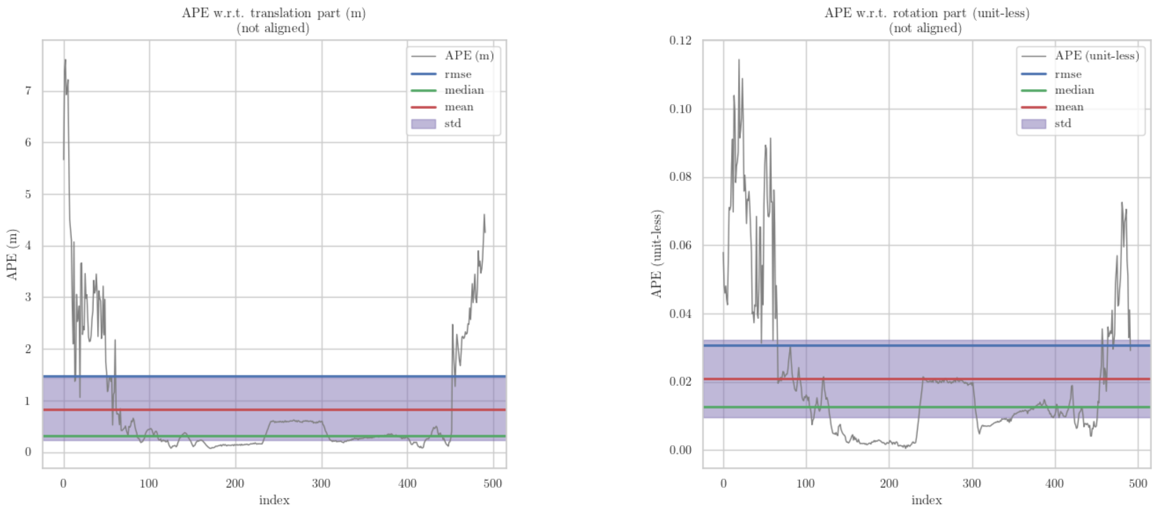


Figure 5.19: APE for P2P Offline Registration with ground against the full transformation for Threshold 1.



(a) Translation APE plot.

(b) Rotation APE plot.

Figure 5.20: APE plots for P2P offline Registration with ground against translation and rotation for correspondence threshold 1.

The APEs plots represented in Figures 5.21, 5.22a, and 5.22b also shows the fluctuations appeared in the trajectory plot. Moreover, compared to the online version of P2P with correspondences experiment, the maximum value of the full APE increased to 12.561, and the RMSE increased to 2.770. The Rotation APE persists more than the translation RMSE at first, then the error decreases through the middle of the experiment. However, it starts to fluctuate again at the end of it.

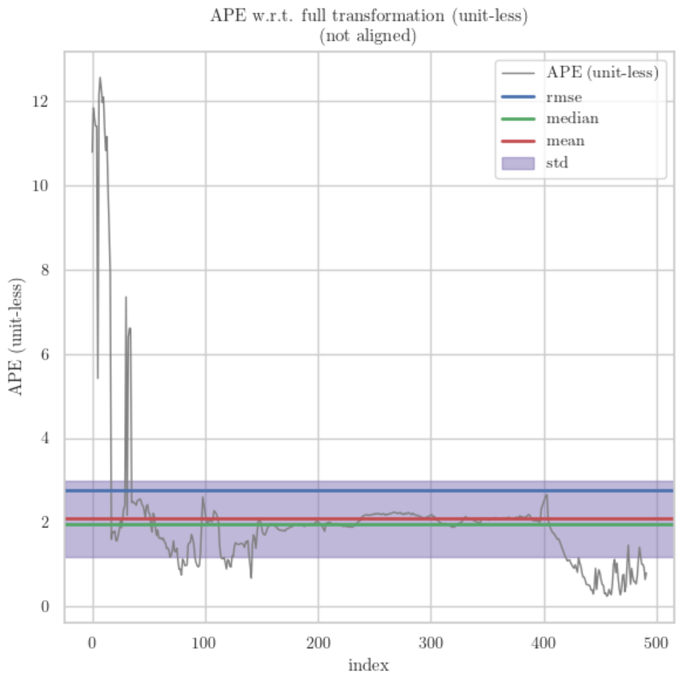
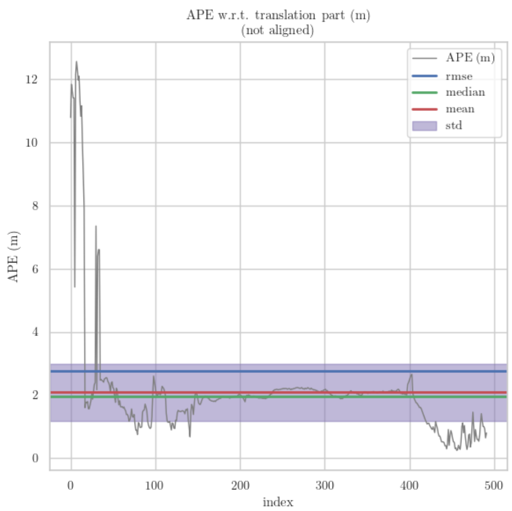
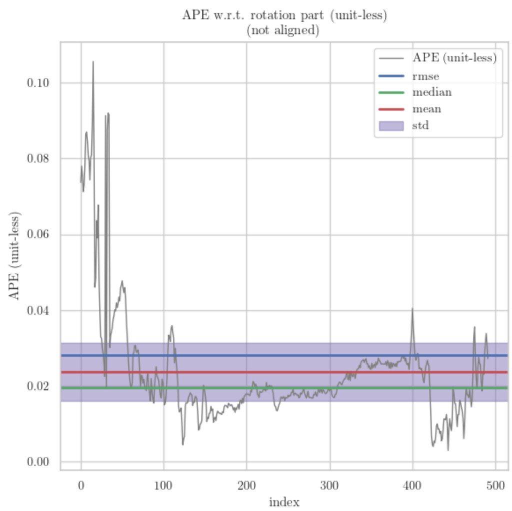


Figure 5.21: APE for P2P offline Registration with ground against the full transformation for Threshold 3.



(a) Translation APE plot.



(b) Rotation APE plot.

Figure 5.22: APE plots for P2P offline Registration with ground against translation and rotation for correspondence threshold 3.

The qualitative results depicted in Figure 5.27 also indicate the small deviations that appeared in the trajectory plots. Upon closer observation, there is a slight forward shift between the vehicle point clouds and the infrastructure local map.

Correspondence threshold 5: Furthermore, for the last Offline P2P experiment, correspondence threshold 5 is used, which means more flexibility to choose more Correspondences. However, the results indicate the gained flexibility affected the estimated trajectory poorly, as the trajectory plot illustrates in Figure 5.23. First, estimated poses are aligned to the ground truth very quickly after only 10 meters. However, the estimated poses keep deviating from the ground truth till the end of the experiment when the estimated poses get off the ground truth completely. In contrast to the online version of this experiment, Online P2P Registration with correspondence threshold 5, the estimated trajectory is completely inconsistent.

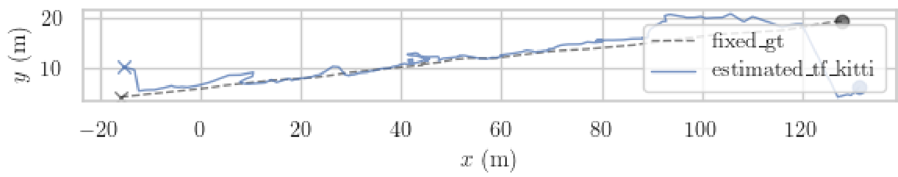


Figure 5.23: Trajectory plot for P2P offline Registration with ground with correspondence threshold 5.

Moreover, the APEs analysis reflects numerically what was depicted in the trajectory plot as shown in the full, translation, and rotation APEs plots in Figures 5.24, 5.25a, and 5.25b respectively. Where the full APE shows the highest maximum error value for all Classical P2P ICP at 15.383 and RMSE at 4.034. Moreover, the fluctuations persist almost until the 100th frame when the vehicle is close to the infrastructure. In contrast to the online version of this experiment, Online P2P Registration with correspondence threshold 5, the APEs increase again at the end of the experiment.

The visualization of the registered vehicle point cloud and the infrastructure point cloud depicted in Figure 5.28 shows poor alignment at the 50th timestamp where the buildings point clouds are not aligned, there is a noticeable front shift. Moreover, at the 450th timestamp, the visualization shows a slight rotational error, which prevents good alignment.

5.2.3 Classical Point-2-Point ICP Comparison

In this subsection, we present Table 5.1, which contains the qualitative numerical results for all variants from the Classical Point-2-Point. Moreover, this table contains both the Absolute Pose Error (APE) & Relative Pose Error (RPE). Delving deep into the statistical data presented in the table, Online & Offline P2P with correspondences threshold 1 experiments have the best results. However, this contradicts the trajectory plots, APEs plot, and the qualitative results depicted in Sections 5.2.1 & 5.2.2 reflects Online P2P with correspondences threshold 3 has the best qualitative and quantitative results. Moreover, the contradiction between different data can be justified by the maximum error values impact on RMSE and Mean error. Usually, the maximum error values exist during the first 50 frames however, it impacts the the rest of the mean errors.

On the other hand, the table reflects the effect of using the infrastructure local map on the Classical P2P. All APE metrics for all correspondence thresholds notably increased compared to the Online Classical P2P.

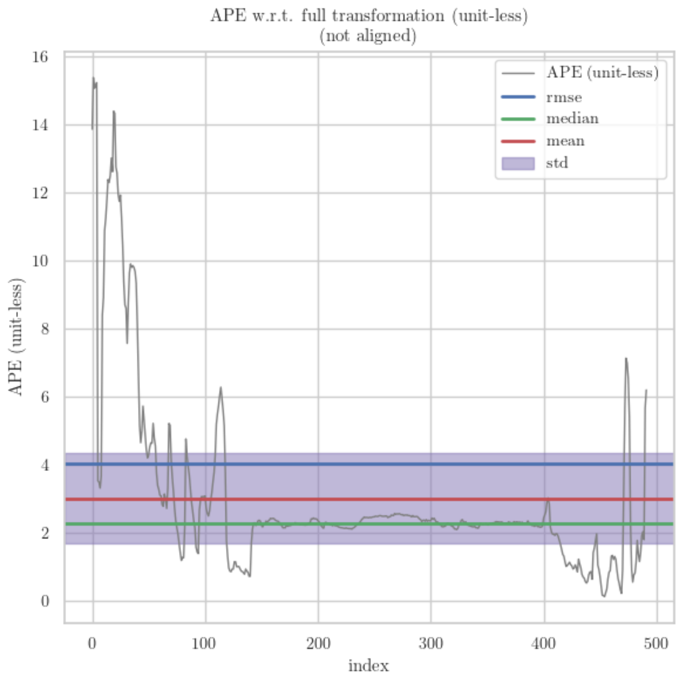
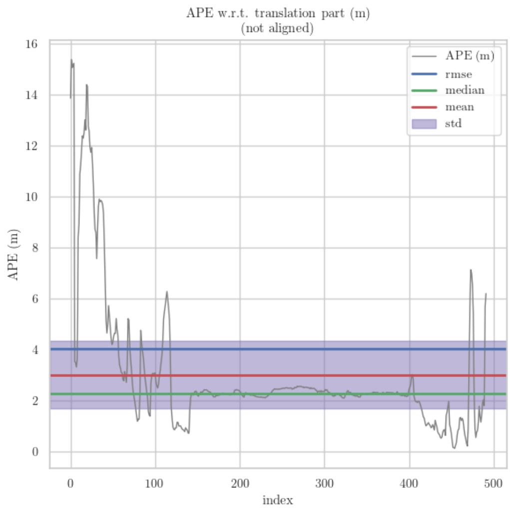
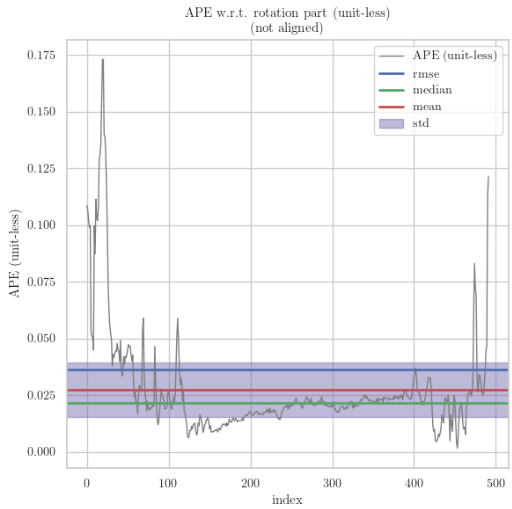


Figure 5.24: APE for P2P Offline Registration with ground against the full transformation for Threshold 5.



(a) Translation APE plot.



(b) Rotation APE plot.

Figure 5.25: APE plots for P2P Offline Registration with ground against translation and rotation for correspondence threshold 5.

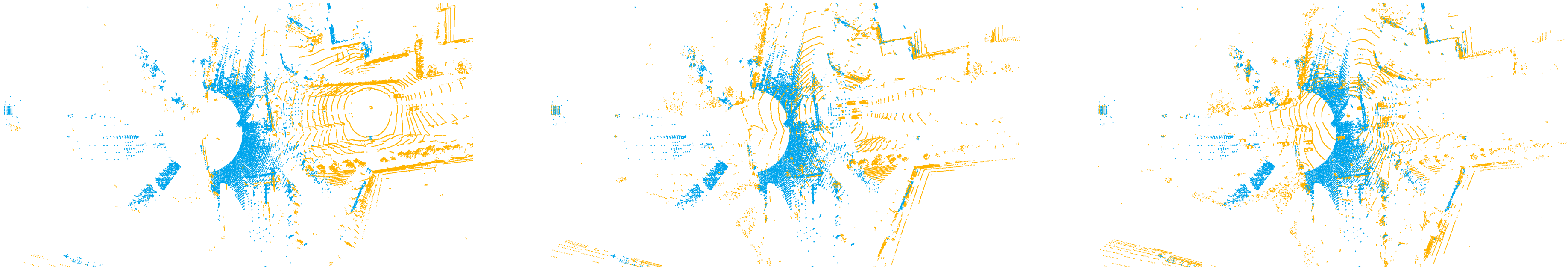


Figure 5.26: P2P Offline Registration with ground Qualitative results for correspondence threshold 1 at timestamps 50, 250, and 450.

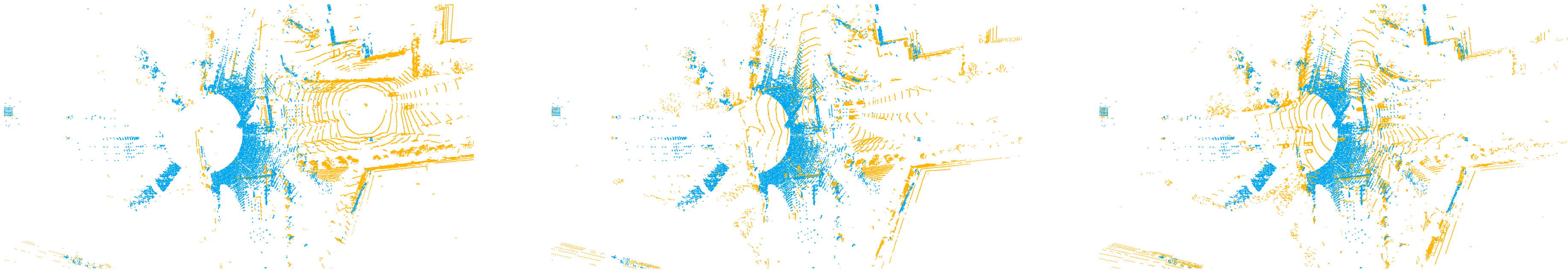


Figure 5.27: P2P Offline Registration with ground Qualitative results for correspondence threshold 3 at timestamps 50, 250, and 450.

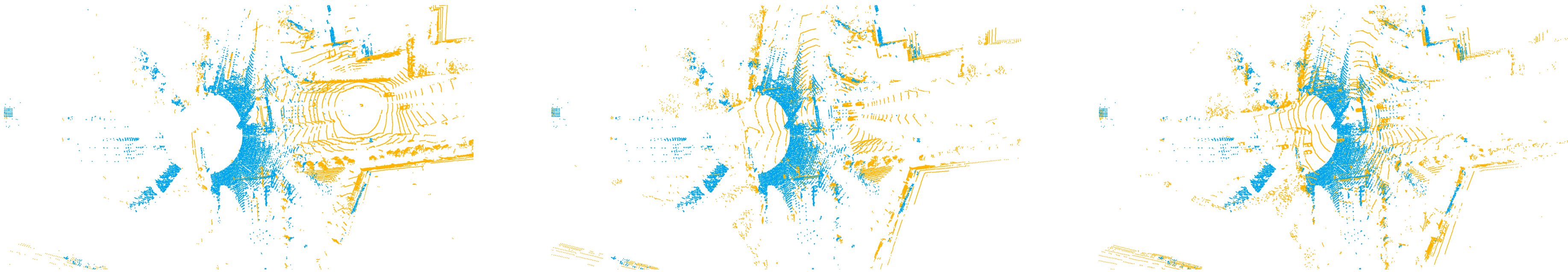


Figure 5.28: P2P Offline Registration with ground Qualitative results for correspondence threshold 5 at timestamps 50, 250, and 450.

Table 5.1: APE and RPE Statistics for Classical P2P ICP

	APE				RPE			
Version / Experiment	RMSE	Mean	Max	Min	RMSE	Mean	Max	Min
Online								
correspondences threshold 1	1.478	0.856	7.137	0.099	0.565	0.258	3.233	0.012
correspondences threshold 3	2.403	1.693	11.339	0.144	1.587	0.660	12.884	0.022
correspondences threshold 5	2.611	2.029	13.768	0.064	1.469	0.620	14.225	0.020
Offline								
correspondences threshold 1	1.473	0.833	7.604	0.068	0.448	0.177	3.307	0.002
correspondences threshold 3	2.769	2.085	12.561	0.248	0.961	0.255	8.483	0.003
correspondences threshold 5	4.034	3.022	15.383	0.130	0.954	0.296	16.736	0.005

5.3 KISS ICP

KISS ICP experiments are also divided into Online and Offline registration experiments. However, as opposing the Classical Point-to-Point, the KISS-ICP experiments do not have a correspondence threshold as a variant input since one of the main contributions of KISS ICP authors [Viz+23] is the adaptive threshold as discussed in 3.1.1. Moreover, for the adaptive threshold, we set the initial threshold to 2.0. On the other hand, KISS-ICP introduces a different variant than the correspondence threshold, it introduces the deskew variable. Deskewing is applying motion compensation to the vehicle point clouds. The results of the Online and Offline KISS ICP registration are presented in the upcoming subsections.

5.3.1 Online Registration

Online Registration with deskew

The results of Online KISS-ICP registration with deskew will be presented in the upcoming paragraphs. Firstly, the trajectory plot shown in Figure 5.29 depicts the estimated trajectory with the ground truth trajectory. Upon close observation, it becomes evident that the estimated trajectory oscillates between the ground truth and poor estimations for the first 70 meters. However, after the first 30 meters, the poor estimations become closer to the ground truth until they align perfectly with the ground truth. Furthermore, the oscillations start again to appear towards the end of the experiment.

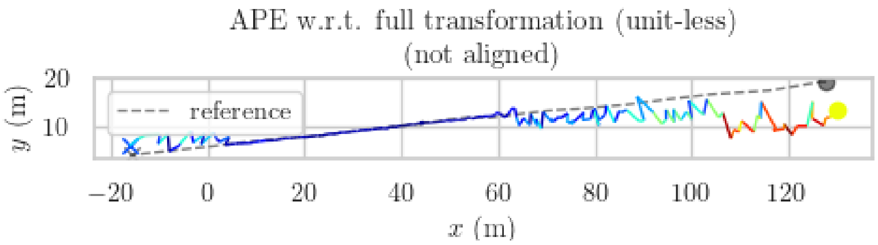


Figure 5.29: Trajectory plot for KISS-ICP Online Registration with ground with deskew.

Subsequently, the APE analysis provides a comprehensive assessment of the alignment between the estimated and ground truth trajectories. Observing the full APE, presented in Figure 5.30, the error starts with a high value and gradually starts to decrease until comes close to zero for most frames. Conversely, Online KISS-ICP with deskew does not generate good pose estimations for the vehicle for the first 50 frames as Online P2P registration. However, after the first 50 frames, Online KISS-ICP with deskew provides outstanding results and alignments with all APEs hovering around zero as shown in the Full APE plot. Additionally, depicted in translation and rotation APEs plots in the following Figures 5.31a & 5.31b.

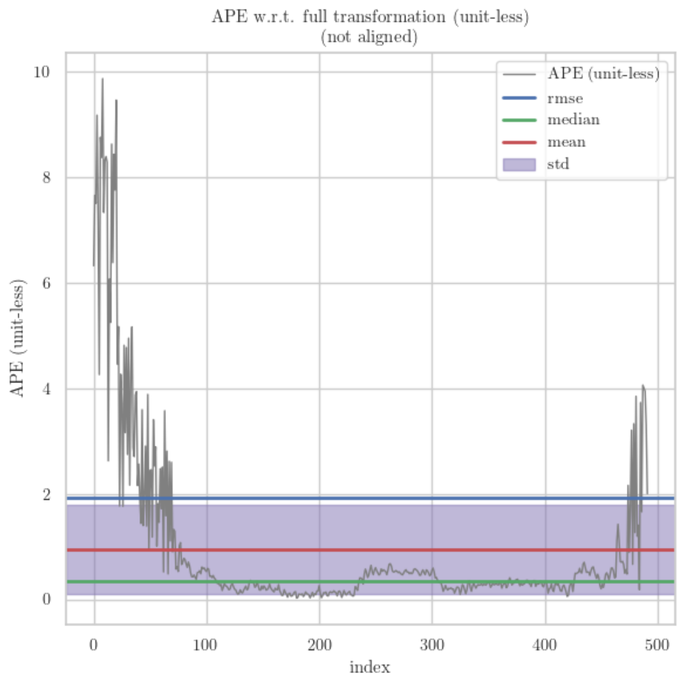
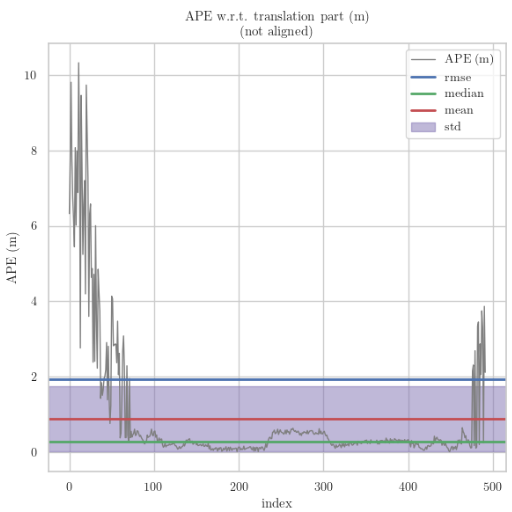
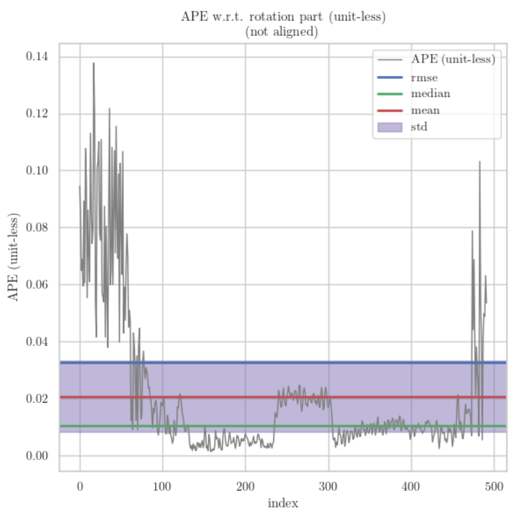


Figure 5.30: APE for KISS-ICP Online Registration with ground against the full transformation with deskew to vehicle point cloud.



(a) Translation APE plot.



(b) Rotation APE plot.

Figure 5.31: APE plots for KISS-ICP Online Registration with ground against translation and rotation with deskew to vehicle point cloud.

On the other hand, the qualitative results presented in Figure 5.35 of the Online KISS-ICP reflect what was indicated by the qualitative results in previous paragraphs. At timestamp 50, the registered point clouds are not fully aligned as the vehicle point clouds, colored yellow, are shifted slightly in the direction of driving. However, at timestamp 250, the visualization depicts how effectively Online KISS-ICP with deskew aligns the vehicle point cloud and the infrastructure point cloud at this frame. At timestamp 490, the Online KISS-ICP with deskew preforms start to decrease again, as reflected in the trajectory plots and the APEs plots.

Online Registration with No deskew:

With the deskew disabled, we explore the effect of the deskew within the vehicle-infrastructure localization context. The visualizations illustrated in Figure 5.32, 5.33, 5.34a, and 5.34b show similar analysis of the trajectory plot & APEs plots of the Online KISS-ICP with deskew, demonstrated in Section 5.3.1.

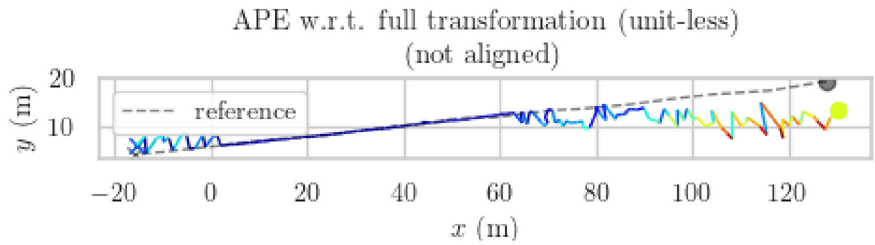


Figure 5.32: Trajectory plot for KISS-ICP Online Registration with ground with no deskew.

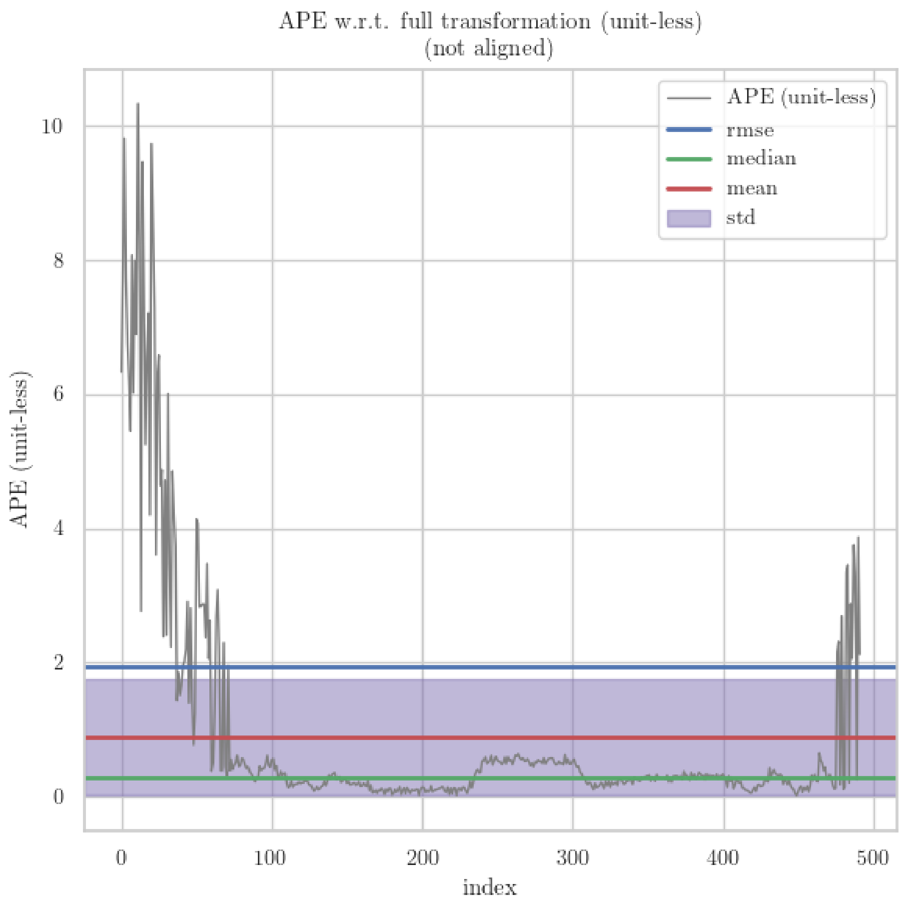


Figure 5.33: APE for KISS-ICP Online Registration with ground against the full transformation with no deskew to vehicle point cloud.

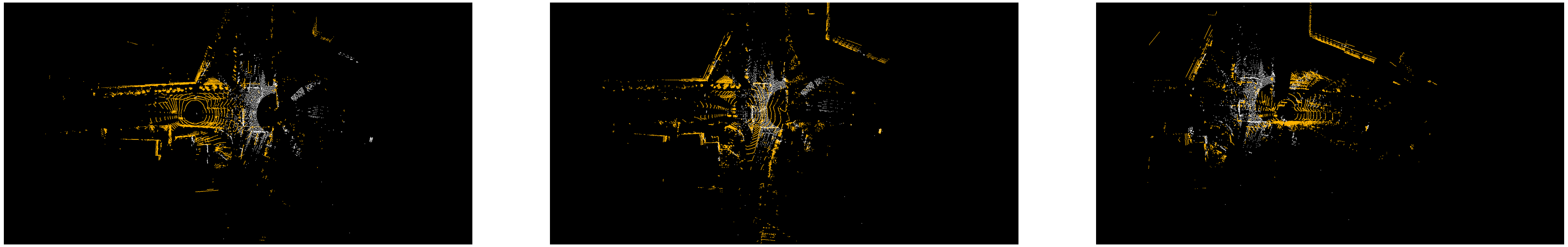


Figure 5.35: KISS-ICP Online Registration with ground Qualitative results with deskew at timestamps 50, 250, and 490.

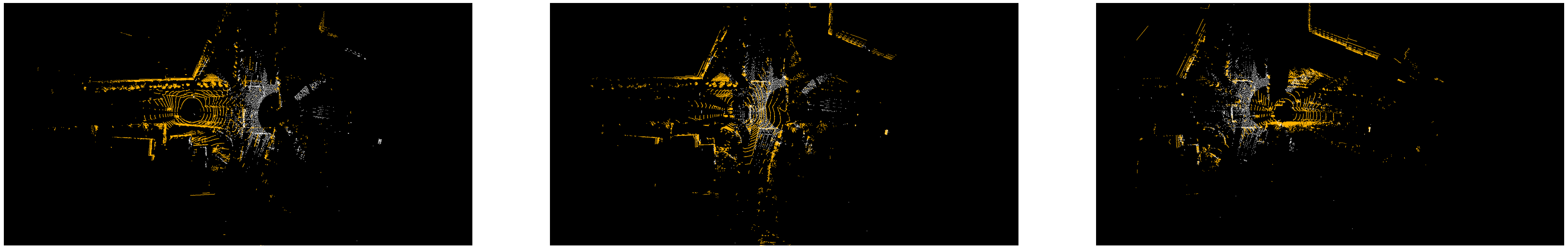
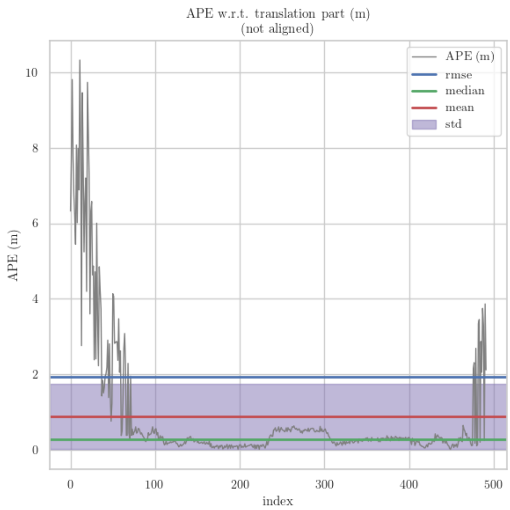
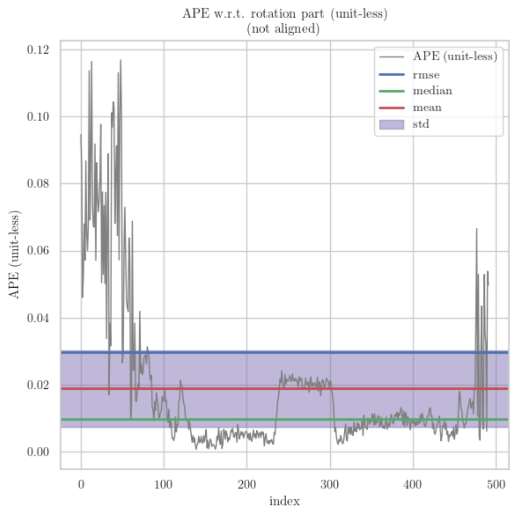


Figure 5.36: KISS-ICP Online Registration with ground Qualitative results with no deskew at timestamps 50, 250, and 490.



(a) Translation APE plot.



(b) Rotation APE plot.

Figure 5.34: APE plots for KISS-ICP Online Registration with ground against translation and rotation with no deskew to vehicle point cloud.

Conversely, the qualitative results in Figure 5.36 of the Online KISS-ICP with no deskew shows better alignments than Online KISS-ICP with deskew experiment. At timestamp 50, a small deviation between the point clouds still exists. Moreover, At timestamps 250 & 490, the visualizations show two point clouds aligned effectively. However, the contradiction between the quantitative and qualitative results is because of the osculation between good and poor estimations and the qualitative results we captured in the good estimation oscillation.

5.3.2 Offline Registration

This subsection unveils the results of the Offline KISS-ICP registration experiments. As previously highlighted, the experiments for KISS-ICP varied in one configuration, enabling or disabling the deskewing of the vehicle point clouds. Initially, we present the outcomes of the Offline KISS-ICP registration experiments with deskew. Then, the results of the same experiment with no deskew. As mentioned in Section 5.1.3, the infrastructure local map is generated using KISS-ICP. Furthermore, Figure 5.4 shows the infrastructure local map that would be used as a target for the Offline KISS-ICP registration.

Offline Registration with deskew

The trajectory plot in Figure 5.37 shows how the vehicle point clouds align with the infrastructure local map using KISS-ICP with deskew enabled. Furthermore, the plot indicates that the Offline KISS-ICP performance is very close to the Online KISS-ICP, where the estimated trajectory is similar to the one explained in Section 5.3.1. In addition, the full, translation, and rotation APEs results depicted in Figures 5.38, 5.39a, 5.39b respectively, also show similar results to the Online version of the experiment. However, with a slight difference, the maximum value of full APE increased to 11.650.

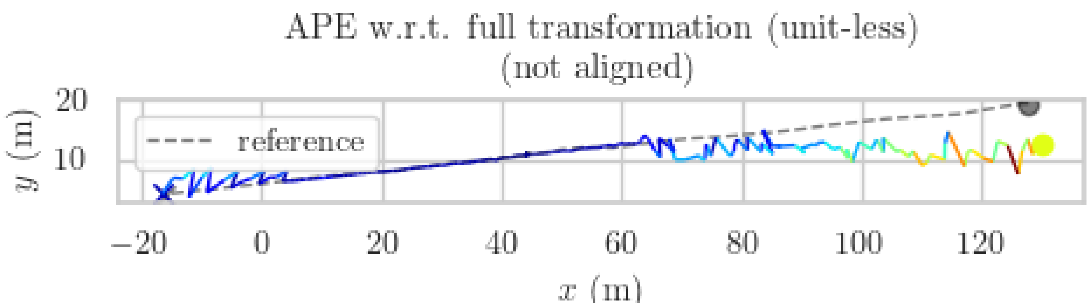


Figure 5.37: Trajectory plot for KISS-ICP Online Registration with ground with deskew.

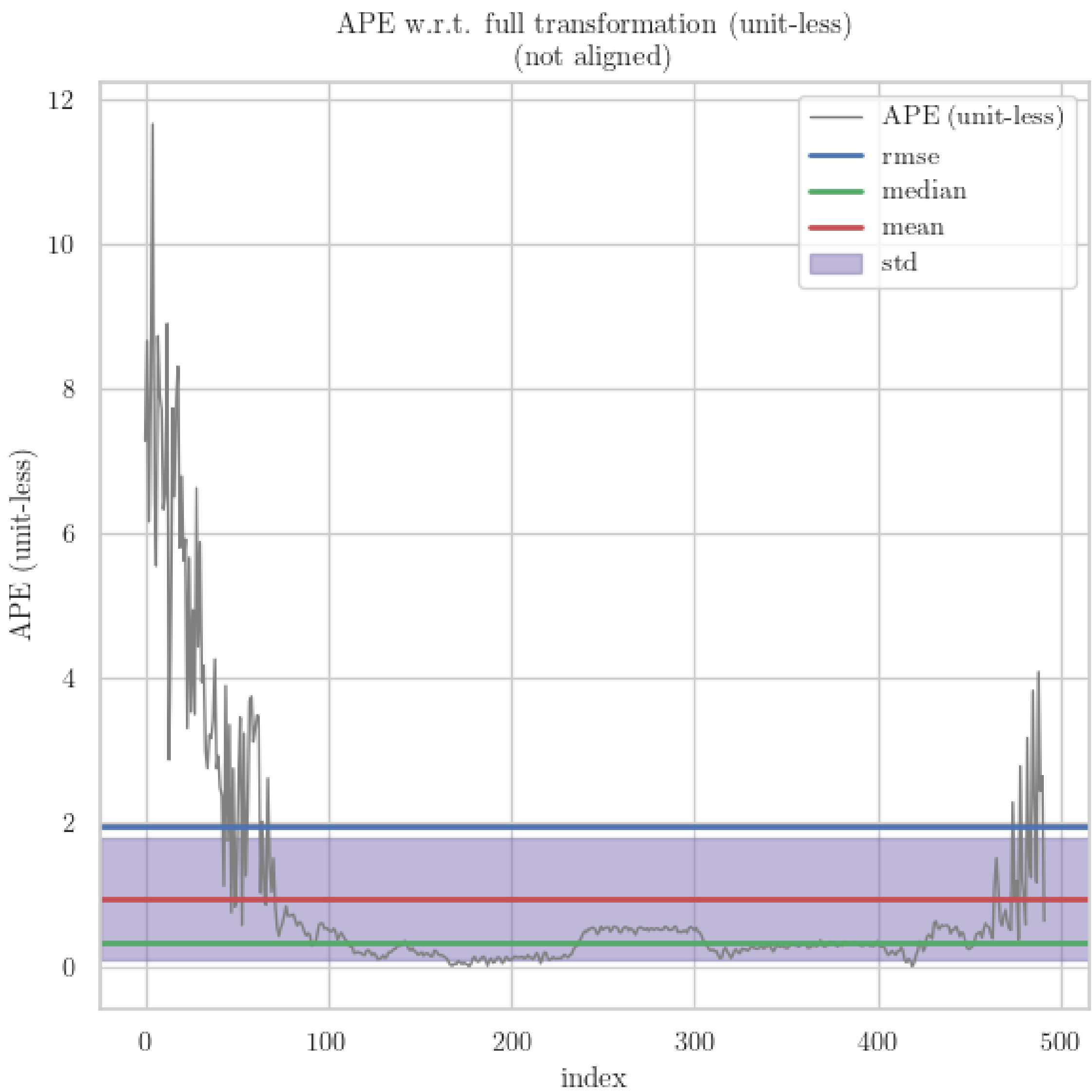
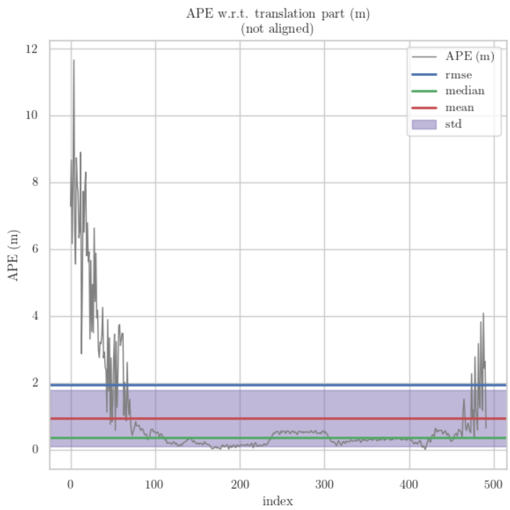
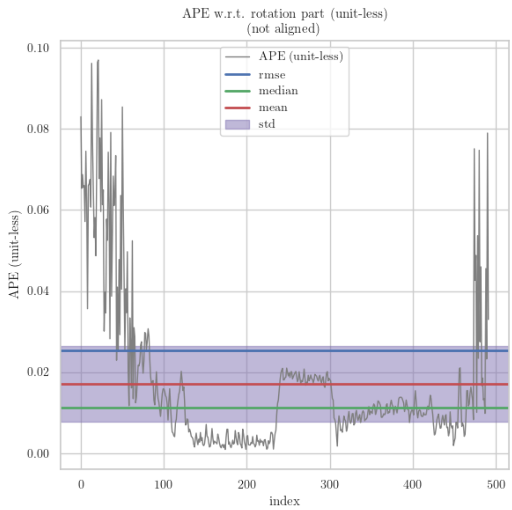


Figure 5.38: APE for KISS-ICP Online Registration with ground against the full transformation with deskew to vehicle point cloud.



(a) Translation APE plot.



(b) Rotation APE plot.

Figure 5.39: APE plots for Offline KISS-ICP Registration with ground against translation and rotation with deskew to vehicle point cloud.

In contrast, the qualitative results depicted in Figure 5.42 show different results than the experiment's online version. At timestamp 50, the visualization shows accurate alignment between the vehicle point clouds and the infrastructure local map. Moreover, at timestamp 490, the vehicle point cloud aligns accurately with the target local map.

Offline Registration with No deskew

The last subsection for the KISS-ICP experiments delves into the results of Offline KISS-ICP of vehicle point cloud to the infrastructure local map with disabling deskewing of the vehicle point cloud. The trajectory plot in Figure 5.40 depicts the estimated trajectory. Furthermore, the full, translation, rotation APEs plots are presented in Figures 5.41, 5.44a, and 5.44b respectively. The analysis of the above referenced figures are similar to the analysis of the Offline KISS-ICP with deskew in Section 5.3.2. Conversely, qualitatively at timestamp 50, the point clouds are not accurately registered. However, the rest of the time stamps (250 & 490) align with great accuracy.

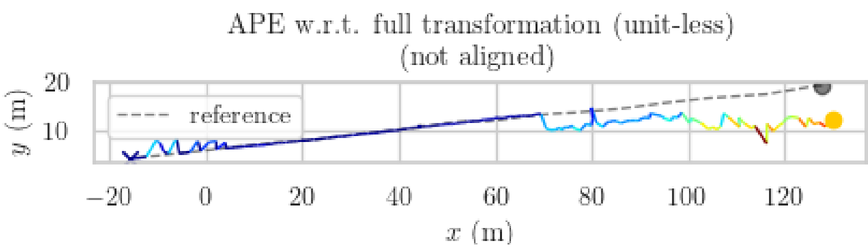


Figure 5.40: Trajectory plot for KISS-ICP Offline Registration with ground with no deskew.

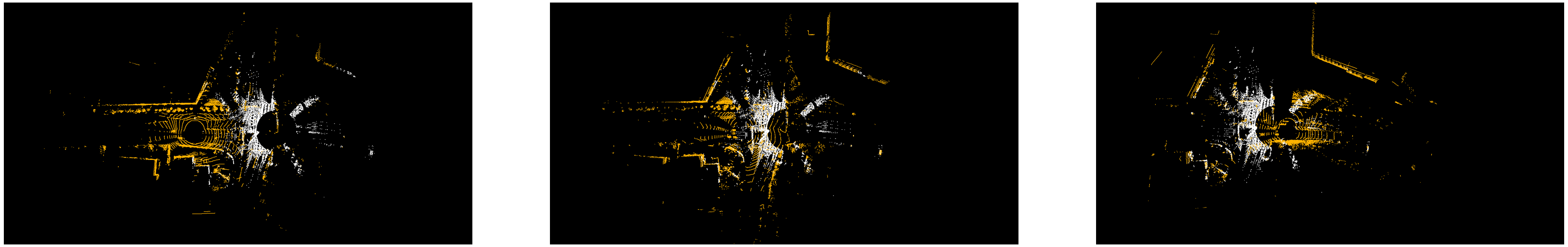


Figure 5.42: KISS-ICP Offline Registration with ground Qualitative results with deskew at timestamps 50, 250, and 490.

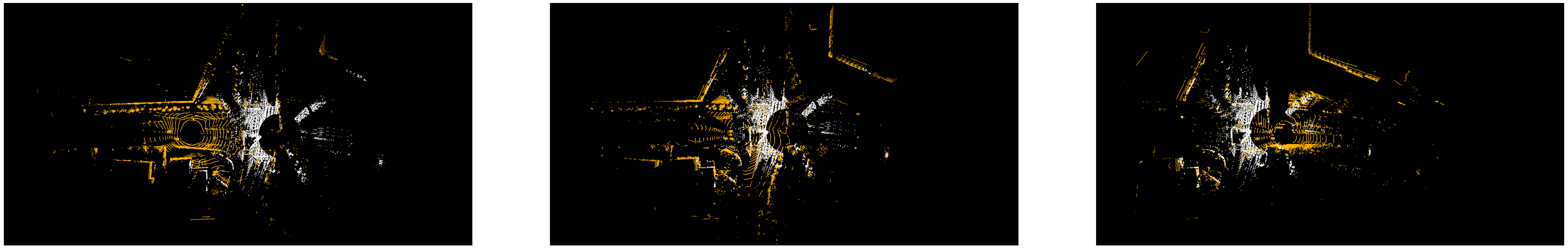


Figure 5.43: KISS-ICP Offline Registration with ground Qualitative results with no deskew at timestamps 50, 250, and 490.

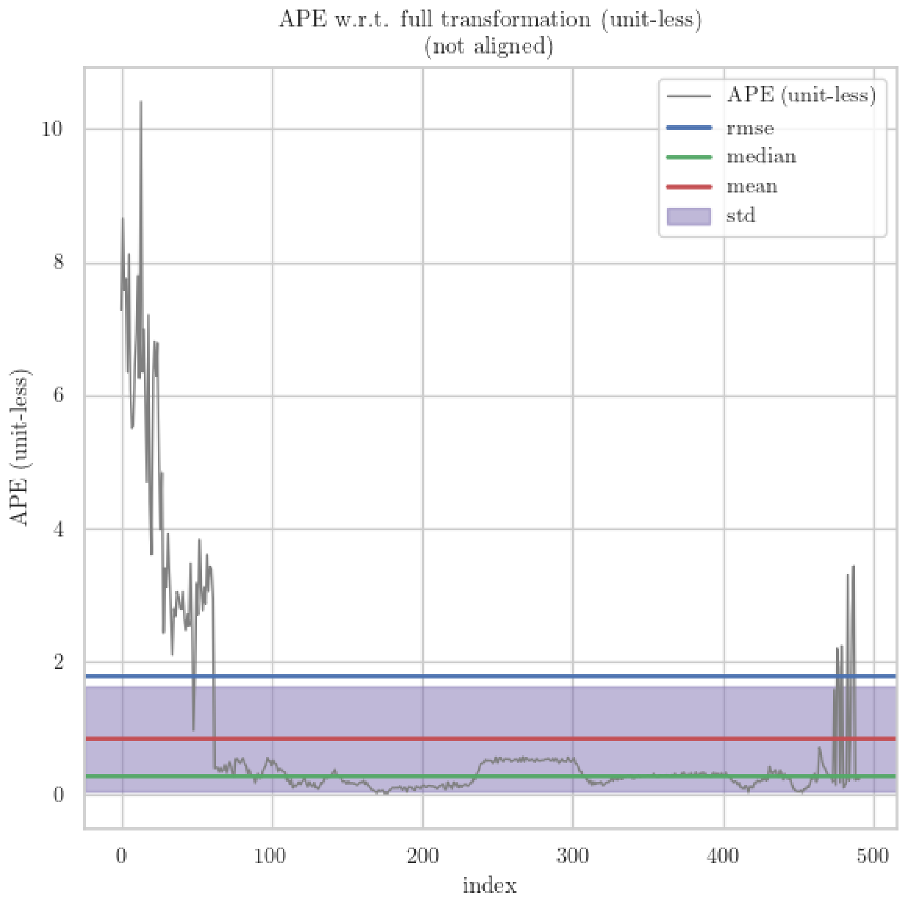
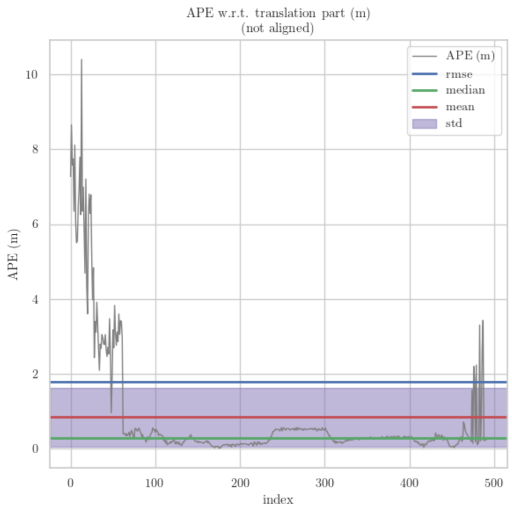
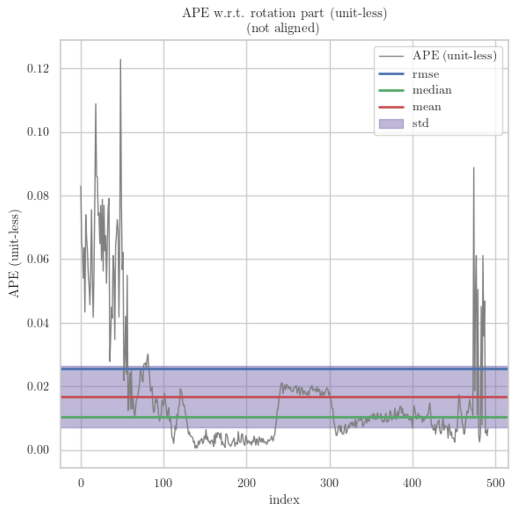


Figure 5.41: APE for KISS-ICP Offline Registration with ground against the full transformation with no deskew to vehicle point cloud.



(a) Translation APE plot.



(b) Rotation APE plot.

Figure 5.44: APE plots for KISS-ICP Offline Registration with ground against translation and rotation with no deskew to vehicle point cloud.

5.3.3 KISS ICP Comparison

KISS-ICP experiments show slight differences between each variant. Moreover, Table 5.2 presents the numerical values for each KISS-ICP experiment variant which reflects the slight differences between the KISS-ICP experiments. Furthermore, the table shows the best result from the KISS-ICP experiments is the **Offline KISS-ICP with no deskew** where the RMSE is 1.801 which is the lowest RMSE compared to the rest of the KISS-ICP experiments.

Table 5.2: APE and RPE Statistics for KISS-ICP

	APE				RPE			
Version / Setting	RMSE	Mean	Max	Min	RMSE	Mean	Max	Min
Online								
deskew	1.939	0.955	9.873	0.027	1.058	0.476	6.050	0.007
No deskew	1.947	0.895	10.322	0.018	0.989	0.423	7.043	0.011
Offline								
deskew	1.934	0.948	11.650	0.011	0.984	0.416	6.041	0.006
No deskew	1.801	0.841	10.405	0.020	0.659	0.267	4.266	0.009

5.4 MULLS ICP

In our experimentation with MULLS ICP, we encountered some complexities primarily stemming from its implementation in C++. As delineated in 3.1.1, MULLS has two distinct modes: MULLS and MULLS-ICP. Fundamentally, MULLS is designed for offline experiments, leveraging SLAM methodologies. A feasible approach for offline experiments involved integrating the Infrastructure Local Map as the first frame for processing. Conversely, MULLS-ICP is designed for online experiments. Its primary function is to register online frames consecutively. Yet, our efforts with MULLS-ICP failed because of two challenges: the complexity of inputting the initial transformation matrix and the frequent inability of the system to identify sufficient correspondences between frames. Moreover, our experimental outcomes suggested that the system largely disregarded the infrastructure point clouds.

5.5 Ablation Study

In this section, we will look into the effect of removing the initial transformation matrix, created from GPS and IMU data, on vehicle-infrastructure localization.

5.5.1 Influence of Initial Transformation Matrix

For this ablation study, we used offline KISS-ICP, where we used the infrastructure local map. Figure 5.45 depicts the trajectory plot of the estimated and the ground truth trajectories where the estimated trajectory is incredibly inaccurate. Moreover, the qualitative result shown in Figure 5.46 also confirms the inaccuracy. Concluding, the current SOTA LiDAR Odometry needs an initial transformation matrix to work.

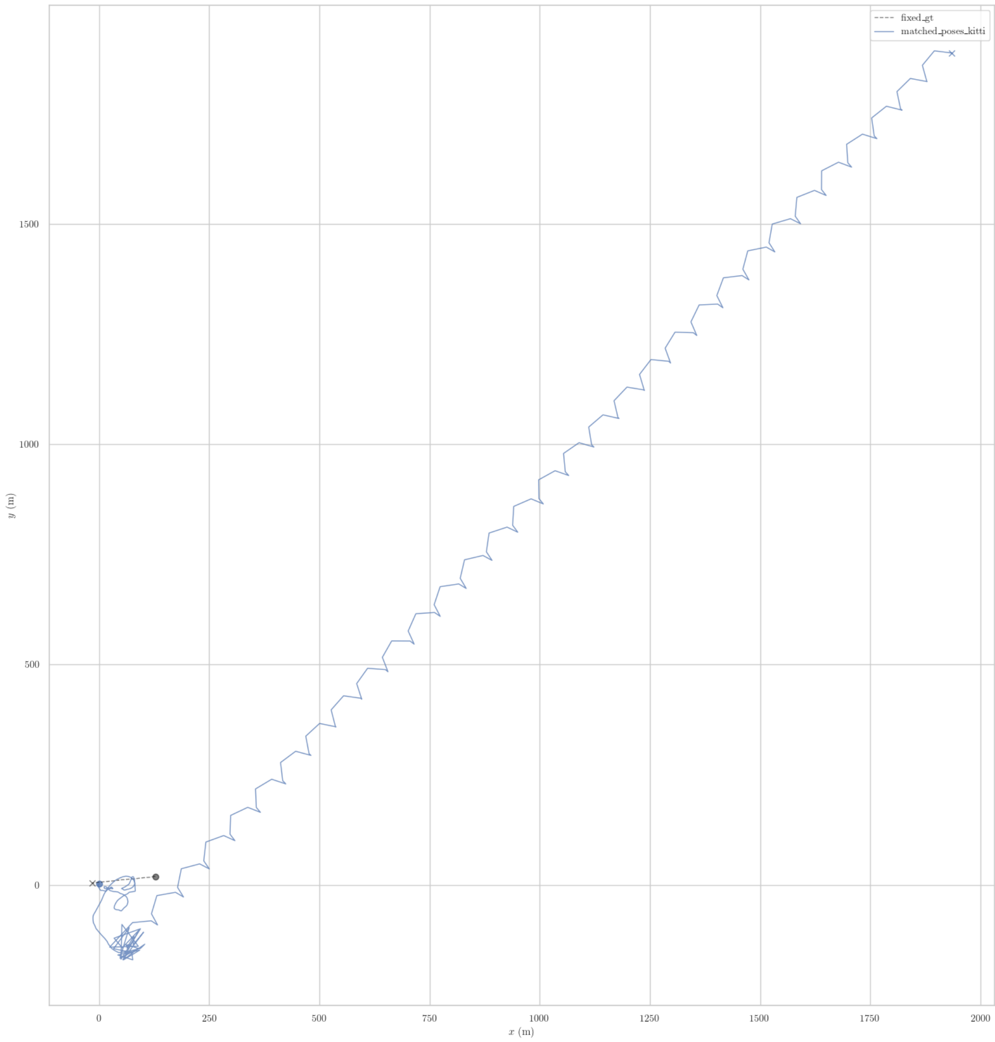


Figure 5.45: Trajectory plot for offline KISS-ICP without using the initial transformation matrix.

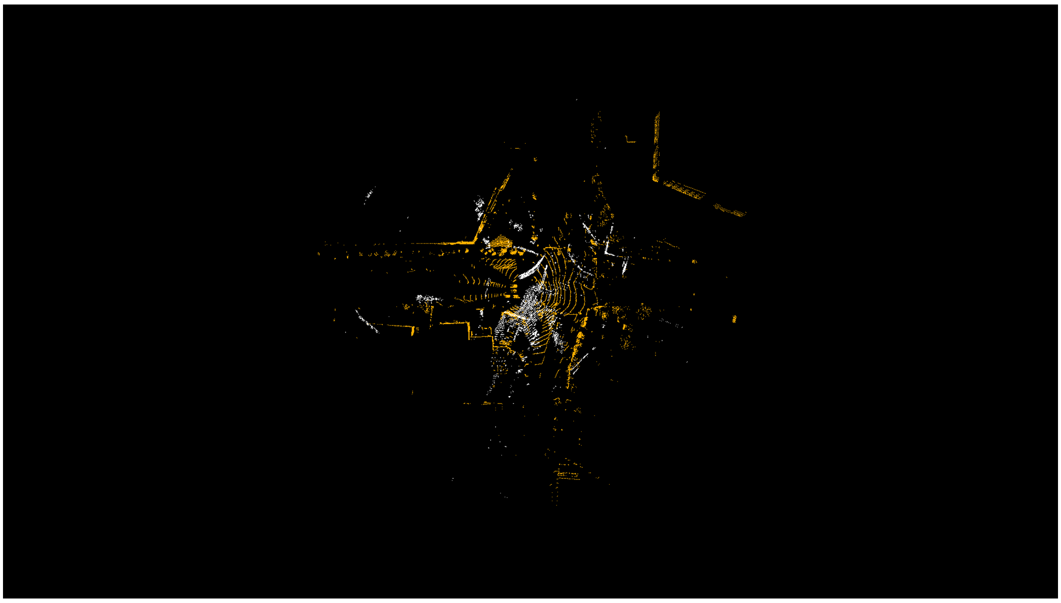


Figure 5.46: Qualitative results of offline KISS-ICP in vehicle-infrastructure localization at timestamp 250.

5.6 Discussion

In this section, we delve into our findings, offering a comprehensive analysis of different ICP methods with different experimental setups used in the vehicle-infrastructure localization context. After presenting the qualitative and quantitative results for two different ICP methods, Classical Point-to-Point ICP, and KISS-ICP, with different configurations each, for discussion, we choose the following experiments, as they cover the best results in each version (Online & Offline) for the two methods:

- **Online P2P ICP with correspondence threshold 3.**
- **Offline P2P ICP with correspondence threshold 3.**
- **Online KISS-ICP with no deskew.**
- **Offline KISS-ICP with no deskew.**

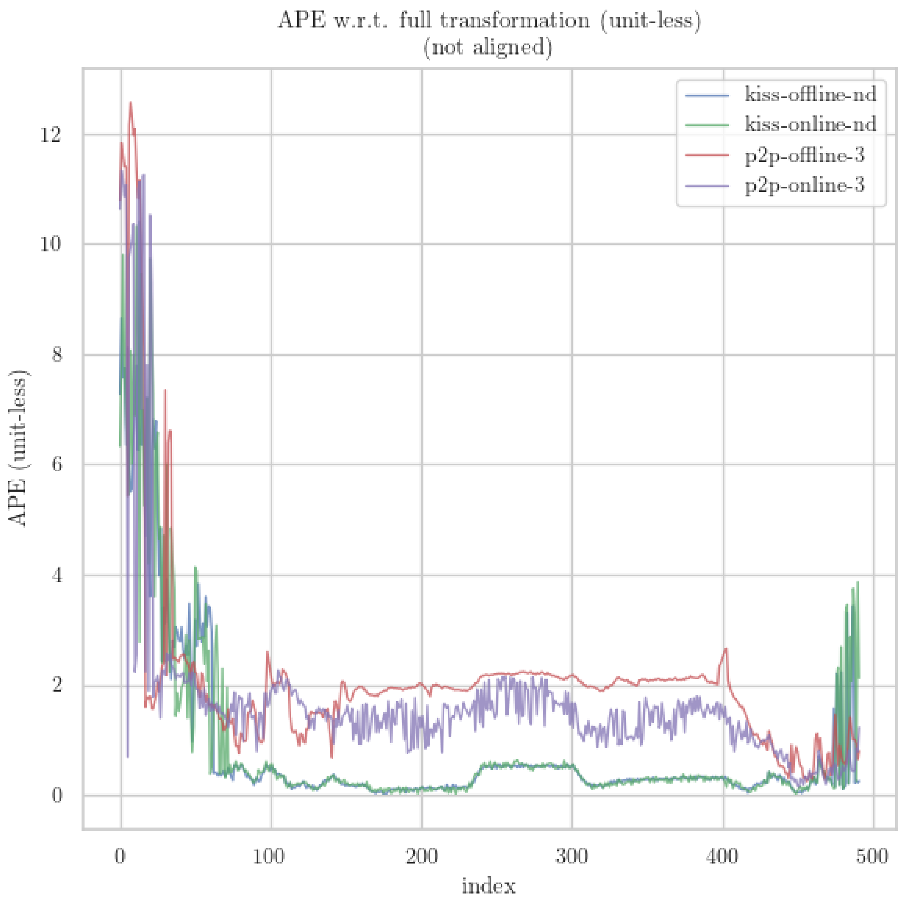


Figure 5.47: APE Plots for Online & Offline Classical P2P ICP and Online & Offline KISS ICP.

In the following paragraph, we delve into the discussion between the APEs results and the estimated trajectories of the four experiments listed above. Figure 5.47 depicts the four full APEs of each experiment. The figure shows that all methods have a similar behavior at the beginning of the experiment, which is all methods struggle to estimate poses with high accuracy when the vehicle is still far away from the infrastructure. On the one hand, after a few frames, Online and Offline Point-to-Point (P2P) begin to yield improved poses. On the other hand, Online and Offline KISS-ICP generate better pose estimates a few frames later after the P2P method. Moreover, P2P keeps a persistent full APE value (between 1 and 2) till the 400th frame. Conversely, KISS-ICP yields the smallest error between the two ICP methods almost till the final frames. This indicates how efficient KISS-ICP is once the car is close to the infrastructure, hence more overlapping point clouds. At the end of the experiments, when the vehicle is driving through and away from the infrastructure, P2P generates more accurate poses than KISS-ICP, which starts to generate poor estimates. From the statistical data depicted in Figure 5.48, we can deduce that the Online and Offline KISS-ICP outperform Online and Offline P2P quantitative results. In contrast, the trajectory plot presented in Figure 5.49 shows the top view of the trajectories where Online and Offline P2P align to the ground truth much faster. Additionally, these P2P pose estimates consistently overlap with the ground truth til the end. On the other hand, Online and Offline KISS-ICP align to the ground truth trajectory later than P2P, where the estimations are constantly getting better until KISS-ICP trajectories align completely. Furthermore, KISS-ICP shows more inconsistent pose estimations towards the end. Delving deep into the trajectories XYZ view illustrated in Figure 5.50, all methods with different versions and configurations generate highly accurate $X-axis$ estimates. Furthermore, the $Y-axis$ estimates are similar to the trajectory plots discussed above. However, for the $Z-axis$, Online and Offline KISS-ICP yields more accurate results than P2P, which shows how efficiently KISS-ICP aligns the road in the vehicle and infrastructure point clouds. Moreover, the above observation also explains the difference between the qualitative and quantitative results previously.

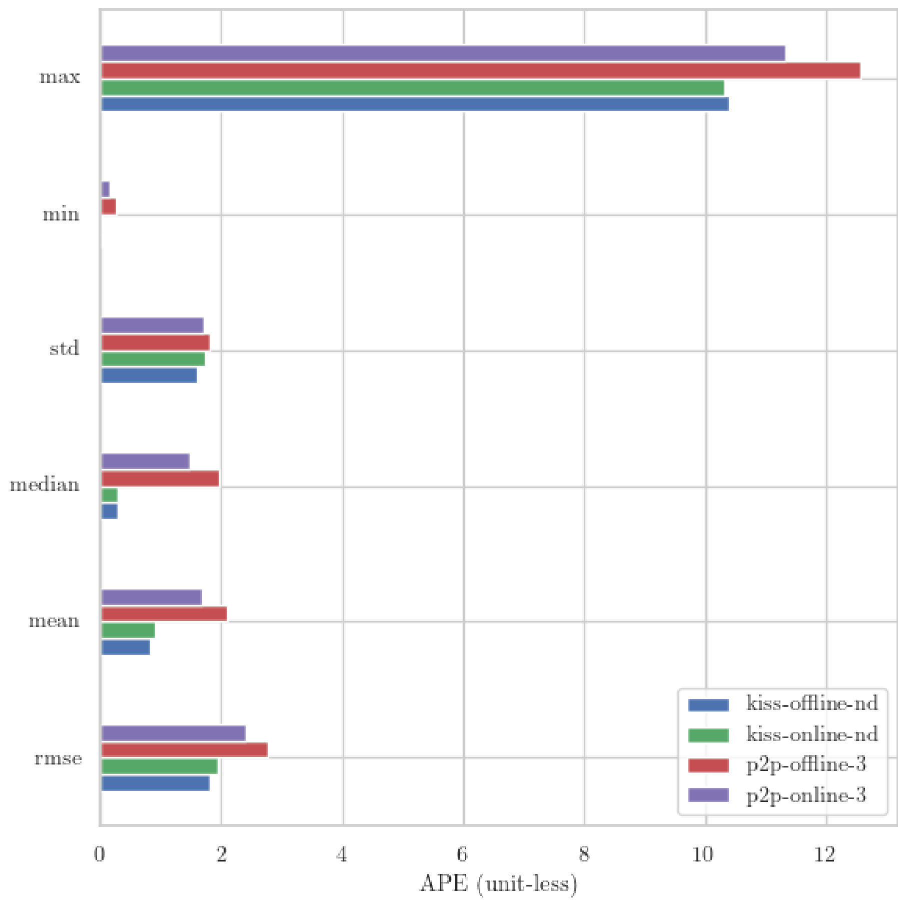


Figure 5.48: APE Statistical Data for Online & Offline Classical P2P ICP and Online & Offline KISS ICP.

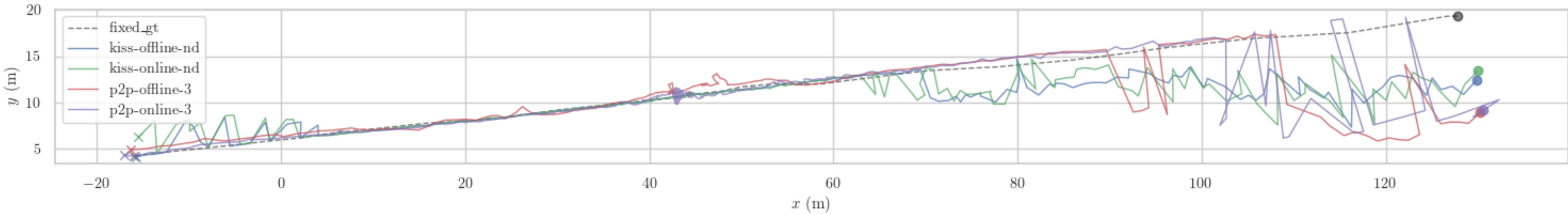


Figure 5.49: Online & Offline Classical P2P ICP and Online & Offline KISS ICP trajectories plot with XY view.

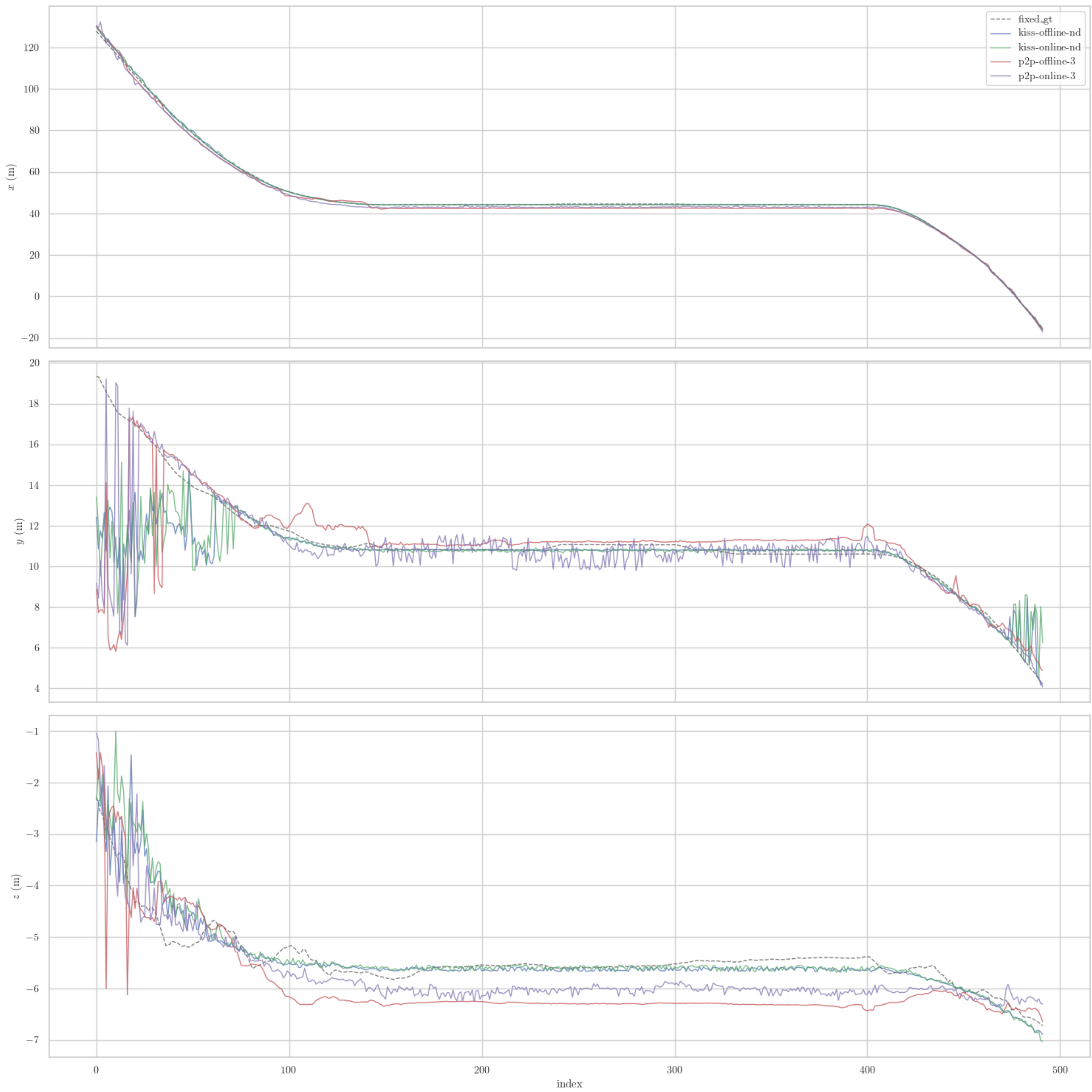


Figure 5.50: Online & Offline Classical P2P ICP and Online & Offline KISS ICP trajectories plot with XYZ view.

In addition, the trajectory plots and the APEs results highlight the performance of using the same method with online or offline infrastructure point clouds. Upon closer observations, Point-to-Point ICP has a noticeable impact by switching from online setup to offline setup, where the APE increases for the frames in the middle. However, For KISS-ICP, the switching from online setup to offline setup, hardly affects the estimated poses as KISS-ICP produces the same estimates, and errors for both offline and online setups

Chapter 6

Conclusion & Future Work

In the preceding chapters, we examined the effect of adding a secondary input, infrastructure LiDAR, on vehicle localization. We delved into the background and fundamentals of vehicle localization. Then, we discussed the SOTA LiDAR Odometery methods and the methodology of how classical and SOTA methods are modified and used in this thesis. Moreover, we conducted experiments using ICP methods and critically analyzed the results. Finally, this chapter aims to discuss the key findings, draw conclusions, highlight blockers, point to improvement areas, and reflect upon potential topics for future research.

6.1 Key Findings

This research explored how adding point clouds from infrastructure LiDAR to vehicle point clouds can improve vehicle localization. To do this, we carefully gathered data from both sources, the vehicle and the infrastructure. We then applied classical and state-of-the-art ICP methods to align the vehicle's data with the infrastructure point clouds or local map. Furthermore, we draw the following conclusions for vehicle-infrastructure localization:

- Dataset creation is a multi-step task that requires high effort to collect, clean, and annotate the dataset. Moreover, the collection part needs a high level of calibration between the sensors to skip manual steps to generate ground truth.
- The lack of research on vehicle-infrastructure localization topic, as the SOTA localization methods only leverage one source of point clouds. Resulting in the lack of vehicle-infrastructure LiDAR Odometry benchmark.
- The complexity of registering vehicle point cloud to infrastructure point cloud due to:
  - Limited overlap between the two point clouds.
  - Varied Field of Views (FoVs) of the vehicle and infrastructure LiDARs. As each LiDAR is mounted on different heights.
- The need for an initial transformation matrix that utilizes the GPS and IMU data for vehicle-infrastructure point cloud registration
- Classical Point-to-Point ICP is a consistent and reliable ICP method for registering the vehicle to infrastructure point clouds. Even though there were small errors. Making Point-to-Point ICP a reliable base method for upcoming novel vehicle-infrastructure localization methods. In other words, vehicle-infrastructure LiDAR Odometry

- The efficiency of KISS-ICP in achieving near-perfect registration, though its performance decreases with increasing distance between the vehicle and infrastructure.
- The Offline version, using the infrastructure local map, affects the ICP methods in different ways compared to the online version, using infrastructure live point clouds of the ICP methods:
  – Classical P2P ICP accuracy decreased when using the infrastructure local map.
  – Online KISS-ICP performance kept consistent with the offline version.

Furthermore, we were able to create a simple vehicle-infrastructure dataset for this research, which revealed challenges that needed to be solved to create a larger and more sophisticated dataset for vehicle-infrastructure localization using Providentia++ infrastructure [Krä+21]. Subsequently, the creation of a code base for vehicle-infrastructure acts as a starting point for subsequent research in a similar direction[1]. This code base includes the Classical P2P ICP method for vehicle-infrastructure localization, modified KISS-ICP for including secondary input and extracting and importing local map, and LiDAR odometry toolkit (scripts needed for generating ground truth, converting data format, ground removal, and automation of evaluation). Moreover, we provided highly comprehensive experimental results for using Classical P2P ICP and KISS-ICP in vehicle-infrastructure localization that can be used as a benchmark for subsequent research.

6.2 Challenges and Limitations

These results show the effect of registering the vehicle point cloud to the infrastructure point cloud and how it impacts vehicle localization. However, it's important to note the following blockers and limitations:

- Dataset collection task, especially the ground truth generation, as it required manual annotation of datasets.
- Variability in system requirements: Each method demands specific Linux versions and carries its dependency list.
- Inconsistency in containerization: Some methods are containerized, enhancing portability, whereas others are not.
- Absence of standardization: The execution protocols for these methods aren't uniform, varying with the chosen localization method.
- The lack of an ICP method to leverage LiDAR information from multiple sources. Resulting in modifying state-of-the-art ICP methods.

6.3 Future Work

Given the research's findings and challenges, several promising directions emerge for subsequent studies:

[1] https://github.com/OSobky/veh-infr-loc/tree/master

- Multi-sensor system creation for data collection and creation for vehicle-infrastructure localization dataset, also including cameras for visual data.
- Publish vehicle-infrastructure large-scale dataset using Providentia++ infrastructure and instituting a leaderboard for various ICP methods [Krä+21].
- Crafting an innovative and novel vehicle-infrastructure LiDAR Odometry that benefits from object detection and completion before registration [Zag+22].
- Eliminating the need for an initial transformation matrix for vehicle-infrastructure localization.
- Vehicle-vehicle localization where localization happens between dynamic LiDARs.
- Investigate beyond ICP methods as learning-based approaches for vehicle-infrastructure localization.

Appendix A

Classical P2P ICP

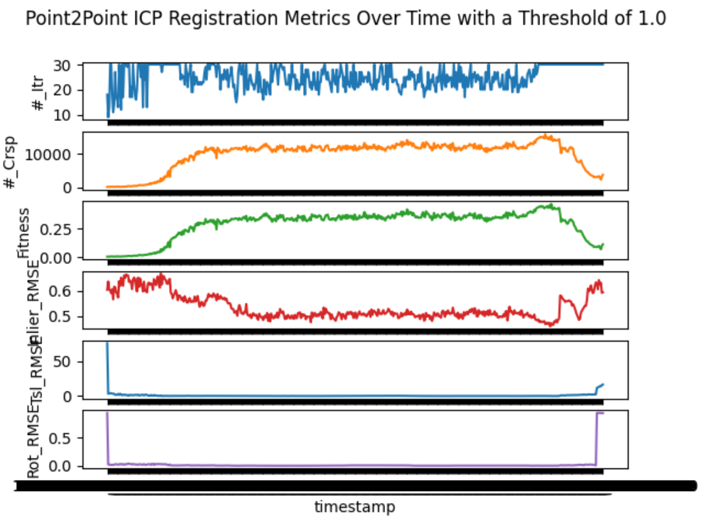


Figure A.1: Different plots metrics for online P2P ICP with threshold 1.

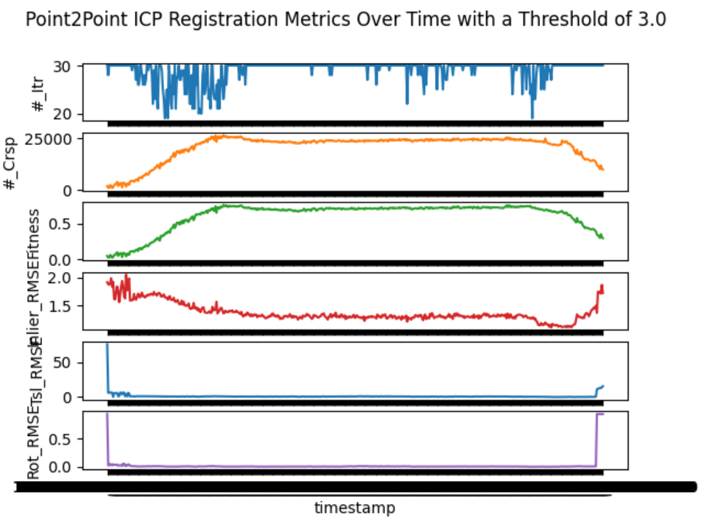


Figure A.2: Different plots metrics for online P2P ICP with threshold 3.

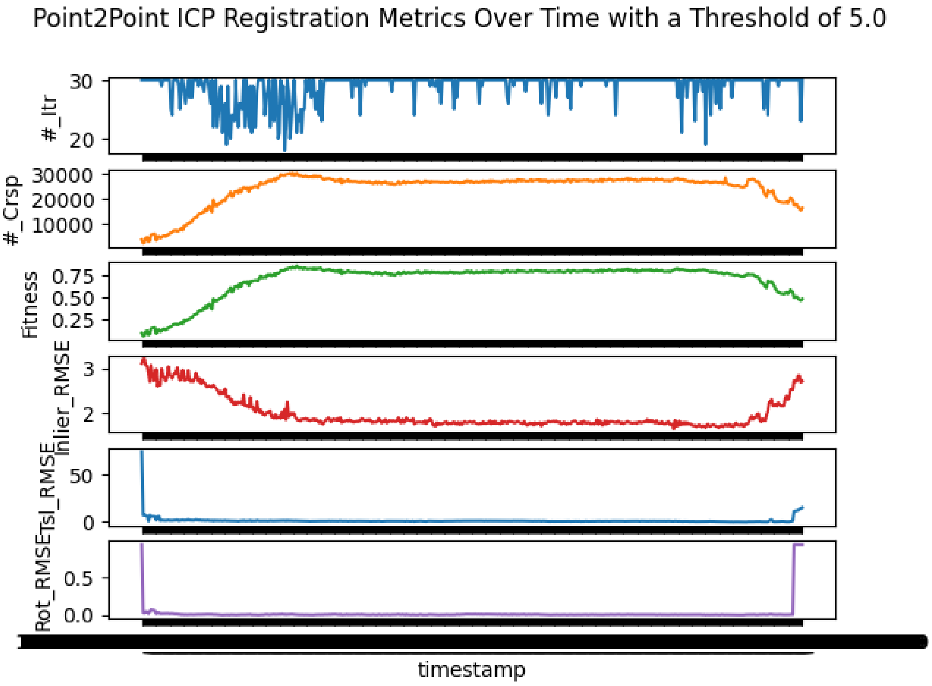


Figure A.3: Different plots metrics for online P2P ICP with threshold 5.

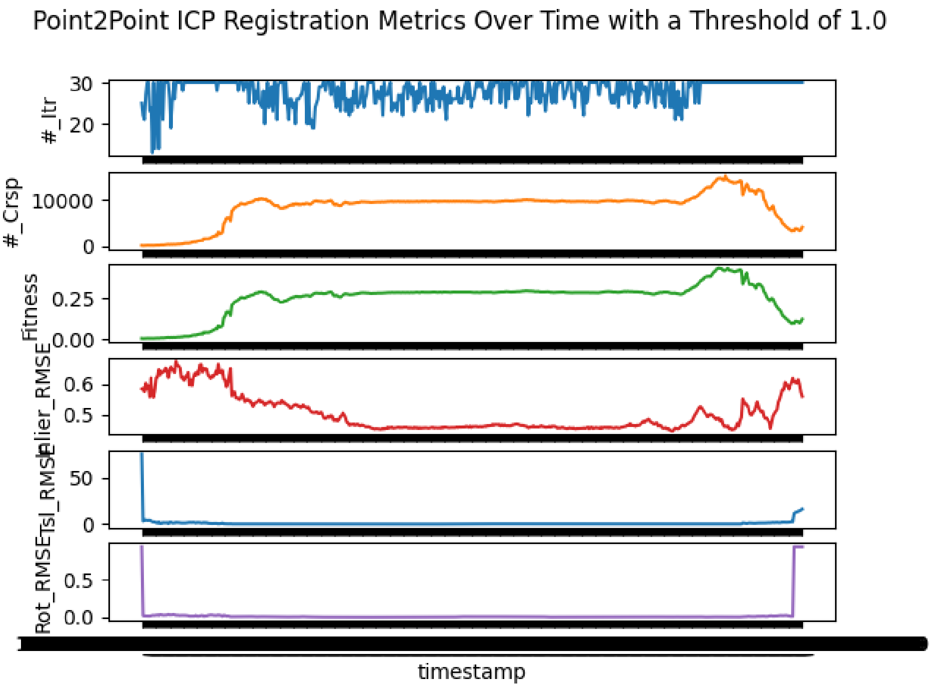


Figure A.4: Different plots metrics for offline P2P ICP with threshold 1.

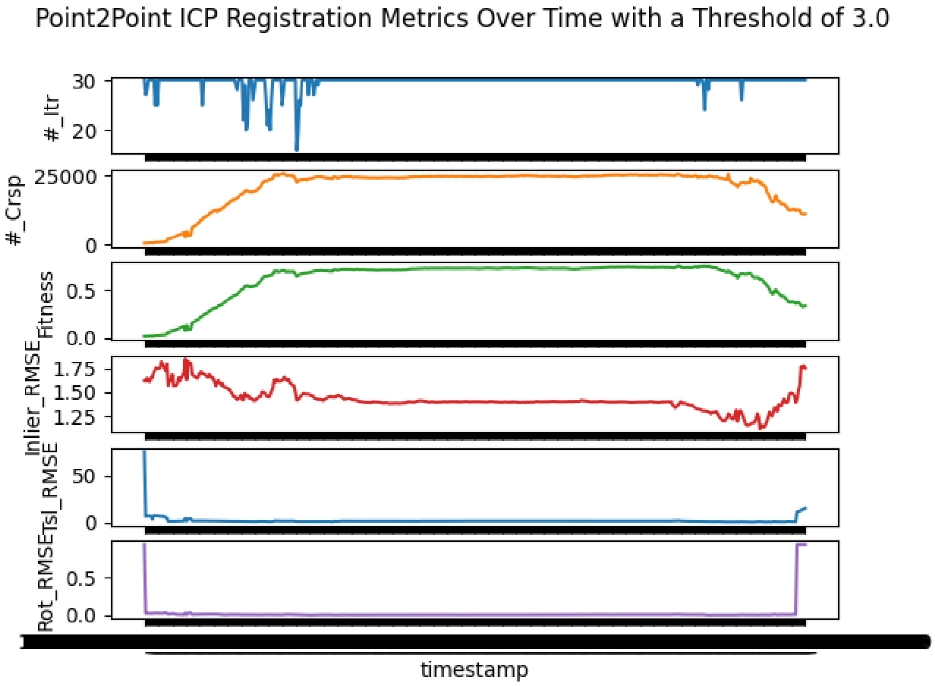


Figure A.5: Different plots metrics for offline P2P ICP with threshold 3.

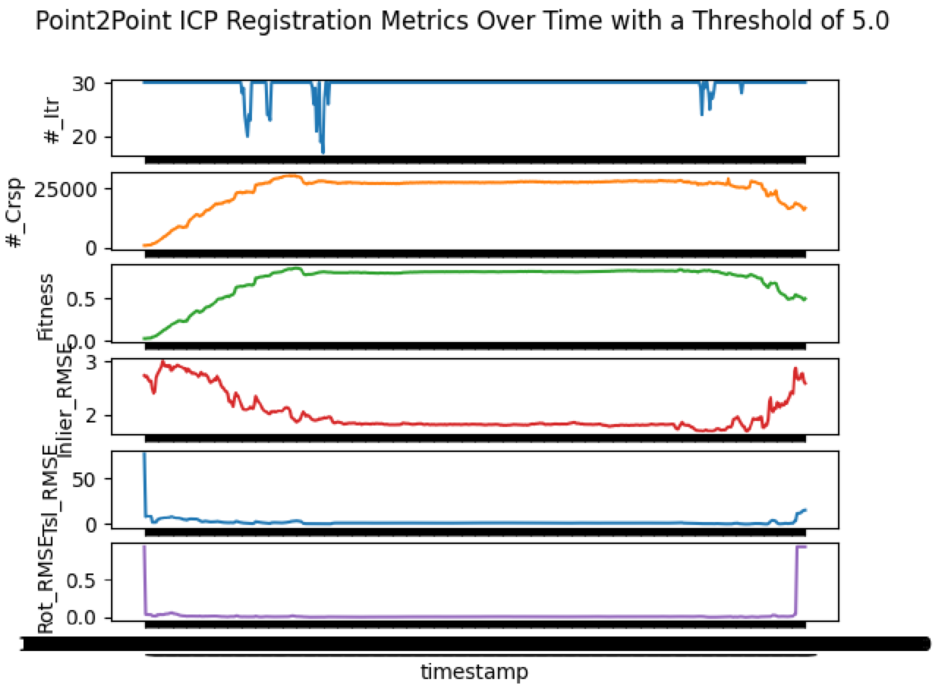


Figure A.6: Different plots metrics for offline P2P ICP with threshold 5.

Bibliography

[Alo+11] Alonso, F., Naranjo, J., Garcia, F., Zato, J., Armingol, J., Escalera, A. de la, and Izquierdo, F. “Limitations of positioning systems for developing digital maps and locating vehicles according to the specifications of future driver assistance systems”. In: *Intelligent Transport Systems, IET* 5 (Apr. 2011), pp. 60–69. DOI: 10.1049/iet-its.2010.0042.

[BS18] Behley, J. and Stachniss, C. “Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments”. In: June 2018. DOI: 10.15607/RSS.2018.XIV.016.

[BM92] Besl, P. and McKay, N. D. “A method for registration of 3-D shapes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256. DOI: 10.1109/34.121791.

[Cad+16] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age”. In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332. DOI: 10.1109/TRO.2016.2624754.

[CUE16] Carlevaris-Bianco, N., Ushani, A. K., and Eustice, R. M. “University of Michigan North Campus long-term vision and lidar dataset”. In: *The International Journal of Robotics Research* 35.9 (2016), pp. 1023–1035.

[CM92] Chen, Y. and Medioni, G. “Object modelling by registration of multiple range images”. In: *Image and vision computing* 10.3 (1992), pp. 145–155.

[con23] contributors, W. *Slerp – Quaternion Slerp*. [Online; accessed November 5, 2023]. Accessed 2023. URL: https://en.wikipedia.org/wiki/Slerp#Quaternion_Slerp.

[Del+22] Dellenbach, P., Deschaud, J.-E., Jacquet, B., and Goulette, F. “CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure”. In: *2022 International Conference on Robotics and Automation (ICRA)*. 2022, pp. 5580–5586. DOI: 10.1109/ICRA46639.2022.9811849.

[Des18] Deschaud, J. “IMLS-SLAM: scan-to-model matching based on 3D data”. In: *CoRR* abs/1802.08633 (2018). arXiv: 1802.08633. URL: http://arxiv.org/abs/1802.08633.

[DB06] Durrant-Whyte, H. and Bailey, T. “Simultaneous localization and mapping: part I”. In: *IEEE Robotics & Automation Magazine* 13.2 (2006), pp. 99–110. DOI: 10.1109/MRA.2006.1638022.

[Fed23] Federal Statistical Office (Destatis). *Road traffic accidents in February 2023: 5% more persons injured than in the same month a year earlier*. Apr. 2023. URL: https://www.destatis.de/EN/Themes/Society-Environment/Traffic-Accidents/_node.html.

[Fen23] Fennelly, J. “Inertial measurement unit sensors lower risk for autonomous vehicles”. In: (2023). Accessed: 2023-10-28. URL: https://www.controleng.com/articles/inertial-measurement-unit-sensors-lower-risk-for-autonomous-vehicles/.

[GSG22] Garigipati, B., Strokina, N., and Ghabcheloo, R. *Evaluation and comparison of eight popular Lidar and Visual SLAM algorithms*. 2022. arXiv: 2208.02063 [cs.RO].

[GLU12] Geiger, A., Lenz, P., and Urtasun, R. "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

[Gru17] Grupp, M. *evo: Python package for the evaluation of odometry and SLAM.* https://github.com/MichaelGrupp/evo. 2017.

[IT20] Ilci, V. and Toth, C. "High Definition 3D Map Creation Using GNSS/IMU/LiDAR Sensor Integration to Support Autonomous Vehicle Navigation". In: *Sensors* 20.3 (2020). ISSN: 1424-8220. DOI: 10.3390/s20030899. URL: https://www.mdpi.com/1424-8220/20/3/899.

[23] *Inertial measurement unit*. Accessed: 2023-10-28. 2023. URL: https://en.wikipedia.org/wiki/Inertial_measurement_unit.

[IM23] Innovation and Mobility. *Projekt Providentia*. Accessed: 2023-09-28. 2023. URL: https://innovation-mobility.com/projekt-providentia/.

[INR] INRIX Press Releases. *Germany Scorecard Traffic Congestion Cities*. https://inrix.com/press-releases/germany-scorecard-traffic-congestion-cities/. Accessed: [31-07-2023].

[Kan+] Kannan, K., Chakrabarty, A., Baculi, J. E., Kawamura, E., Holforty, W., and Ippolito, C. A. "Comparison of Visual and LiDAR SLAM Algorithms using NASA Flight Test Data". In: *AIAA SCITECH 2023 Forum*. DOI: 10.2514/6.2023-2679. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2023-2679. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2023-2679.

[Kim+20] Kim, G., Park, Y. S., Cho, Y., Jeong, J., and Kim, A. "MulRan: Multimodal Range Dataset for Urban Place Recognition". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 6246–6253. DOI: 10.1109/ICRA40945.2020.9197298.

[Krä+21] Krämmer, A., Schöller, C., Gulati, D., Lakshminarasimhan, V., Kurz, F., Rosenbaum, D., Lenz, C., and Knoll, A. *Providentia – A Large-Scale Sensor System for the Assistance of Autonomous Vehicles and Its Evaluation*. 2021. arXiv: 1906.06789 [cs.RO].

[Lam19] Lamprecht, S. "Pyoints: A Python package for point cloud, voxel and raster processing." In: *Journal of Open Source Software* 4.36 (2019), p. 990. DOI: 10.21105/joss.00990. URL: https://doi.org/10.21105/joss.00990.

[LI20] Li, Y. and Ibanez-Guzman, J. "Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems". In: *IEEE Signal Processing Magazine* 37.4 (2020), pp. 50–61. DOI: 10.1109/MSP.2020.2973615.

[Lid23] Lidar, V. *What is LiDAR?* Accessed: 2023-10-28. 2023. URL: https://velodynelidar.com/what-is-lidar/.

[LM97] Lu, F. and Milios, E. "Globally Consistent Range Scan Alignment for Environment Mapping". In: *Autonomous Robots* 4 (1997), pp. 333–349. DOI: 10.1023/A:1008854305733. URL: https://doi.org/10.1023/A:1008854305733.

[Ous] Ouster LiDAR. https://ouster.com/. Accessed: 2023-10-05.

[Pan] Pan, Y. *Case study: example on TLS point cloud registration*. https://github.com/YuePanEdward/MULLS/wiki/Case-study:-example-on-TLS-point-cloud-registration. Accessed: 2023-10-02.

[Pan+21] Pan, Y., Xiao, P., He, Y., Shao, Z., and Li, Z. "MULLS: Versatile LiDAR SLAM via Multi-metric Linear Least Square". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 11633–11640. DOI: 10.1109/ICRA48506.2021.9561364.

[Rea] Reach RS2+ | Emlid. https://emlid.com/reachrs2plus/. Accessed: 2023-10-05.

[Rei+19] Reid, T. G., Houts, S. E., Cammarata, R., Mills, G., Agarwal, S., Vora, A., and Pandey, G. "Localization Requirements for Autonomous Vehicles". In: *SAE International Journal of Connected and Automated Vehicles* 2.3 (Sept. 2019). DOI: 10.4271/12-02-03-0012. URL: https://doi.org/10.4271%2F12-02-03-0012.

[Rob] RoboSense LiDAR. https://www.robosense.ai/en. Accessed: 2023-10-05.

[RL01] Rusinkiewicz, S. and Levoy, M. "Efficient variants of the ICP algorithm". In: *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. 2001, pp. 145–152. DOI: 10.1109/IM.2001.924423.

[Sta93] Stanford University Computer Graphics Laboratory. *Stanford Bunny*. https://graphics.stanford.edu/data/3Dscanrep/. Accessed on January 17, 2019. 1993.

[Stu+12] Sturm, J., Engelhard, N., Endres, F., Burgard, W., and Cremers, D. "A Benchmark for the Evaluation of RGB-D SLAM Systems". In: *Proc. of the International Conference on Intelligent Robot Systems (IROS)*. Oct. 2012.

[Viz+23] Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., and Stachniss, C. "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way". In: *IEEE Robotics and Automation Letters (RA-L)* 8.2 (2023), pp. 1029–1036. DOI: 10.1109/LRA.2023.3236571.

[Wik23a] Wikipedia. *Lidar*. Accessed: 2023-10-28. 2023. URL: https://en.wikipedia.org/wiki/Lidar.

[Wik23b] Wikipedia contributors. *Vehicle-to-everything — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Vehicle-to-everything. [Online; accessed 1-November-2023]. 2023.

[Xse] Xsens Motion Tracking Products. https://www.movella.com/products/xsens. Accessed: 2023-10-05.

[YSC20] Yang, H., Shi, J., and Carlone, L. *TEASER: Fast and Certifiable Point Cloud Registration*. 2020. arXiv: 2001.07715 [cs.RO].

[Yan+22] Yang, M., Sun, X., Jia, F., Rushworth, A., Dong, X., Zhang, S., Fang, Z., Yang, G., and Liu, B. "Sensors and Sensor Fusion Methodologies for Indoor Odometry: A Review". In: *Polymers* 14.10 (2022). ISSN: 2073-4360. DOI: 10.3390/polym14102019. URL: https://www.mdpi.com/2073-4360/14/10/2019.

[Yu+22] Yu, H., Luo, Y., Shu, M., Huo, Y., Yang, Z., Shi, Y., Guo, Z., Li, H., Hu, X., Yuan, J., and Nie, Z. "Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 21361–21370.

[Yu+23] Yu, H., Yang, W., Ruan, H., Yang, Z., Tang, Y., Gao, X., Hao, X., Shi, Y., Pan, Y., Sun, N., Song, J., Yuan, J., Luo, P., and Nie, Z. “V2X-Seq: A large-scale sequential dataset for vehicle-infrastructure cooperative perception and forecasting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[Zag+22] Zagar, B. L., Yurtsever, E., Peters, A., and Knoll., A. C. “Point Cloud Registration with Object-Centric Alignment”. In: *IEEE Access* (2022), pp. 1–1. DOI: 10.1109/access.2022.3191352.

[ZS14] Zhang, J. and Singh, S. “LOAM: Lidar Odometry and Mapping in Real-time”. In: *Proceedings of Robotics: Science and Systems (RSS ’14)*. July 2014.

[Zha+23] Zhang, L., Wang, J., Jie, L., Chen, N., Tan, X., and Duan, Z. “LMBAO: A Landmark Map for Bundle Adjustment Odometry in LiDAR SLAM”. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10097064.

[ZPK18] Zhou, Q.-Y., Park, J., and Koltun, V. “Open3D: A Modern Library for 3D Data Processing”. In: *arXiv:1801.09847* (2018).