# GNSS-RTK ADAPTIVELY INTEGRATED WITH MULTI-ROBOT LiDAR SLAM FOR EFFICIENT MAPPING IN LARGE AREAS

SUN YUNJUAN

MPhil

The Hong Kong Polytechnic University

2024

This page is intentionally left blank

The Hong Kong Polytechnic University

Department of Land Surveying and Geo-Informatics

# GNSS-RTK Adaptively Integrated with Multi-Robot LiDAR SLAM for Efficient Mapping in Large Areas

SUN YUNJUAN

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Philosophy

June 2024

This page is intentionally left blank

**CERTIFICATE OF ORIGINALITY**

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

SUN YUNJUAN (Name of student)

This page is intentionally left blank

**Abstract**

Multi-Robot Cooperative Simultaneous Localization and Mapping (SLAM) extends traditional single-robot SLAM by enabling multiple robots to collaboratively map environments and localize themselves. Recent advancements have integrated The Light Detection and Ranging (LiDAR) and Inertial Measurement Unit (IMU) data through factor graph optimization (FGO) named 3D LiDAR-based frameworks for multi-robot SLAM. While LiDAR and IMU provide accurate short-term motion estimates, they suffer from drift over time, especially in large areas. The Global Navigation Satellite System Real-time Kinematic (GNSS-RTK) offers precise absolute positioning but is less effective in urban canyons. This thesis addresses these challenges by proposing an adaptive integration of GNSS-RTK with LiDAR and Inertial Odometry (LIO) to enhance mapping efficiency and continuous positioning in urban environments.

The proposed method assesses GNSS-RTK solution quality using the incrementally produced point cloud map from LIO, with the mean elevation angle (MEA) mask indicating the openness of the surrounding area. A smaller angle suggests a more open area and a more reliable GNSS-RTK. Global FGO merges reliable GNSS-RTK data with LiDAR and inertial odometry (LIO), incorporating global constraints to counteract pose drift. Testing in Hong Kong's urban canyons

demonstrated significant improvements, reducing absolute pose error (APE) by over 75% compared to conventional methods without GNSS-RTK.

In the two-stage global and local graph optimization, inter-robot constraints are used to determine transformations between robot coordinate systems and are converted to virtual intra-robot constraints. The effectiveness of these virtual constraints in multi-robot SLAM integrated with GNSS-RTK is evaluated, showing varying contributions across datasets. To optimize local pose graphs, correcting inter-robot constraints is necessary to ensure their positive contribution.

This work offers a promising approach for efficient and accurate mapping in complex urban environments, with significant implications for future large-scale mapping projects.

## List of Publications

**Articles**:

1 Sun, Y., Huang, F., Wen, W., Hsu, L. T., & Liu, X. (2023). Multi-Robot Cooperative LIDAR Slam for Efficient Mapping in Urban Scenes. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *48*, 473-478.

2 Sun, Y., Huang, F., Wen, W., Liu, X., & Hsu, L. T. Adaptive Integration of GNSS-RTK with Multi-Robot LiDAR SLAM for Efficient Mapping in Large Urban Areas. *Applied Science* (minor revision).

**Acknowledgement**

I would like to express my deepest gratitude to my supervisor, Dr. Liu, and my co-supervisors, Dr. Wen and Dr. Hsu. Your unwavering support, insightful guidance, and constant encouragement have been instrumental in the completion of this thesis. These past two years have been a journey filled with challenges and tears, but your mentorship has made all the difference. Embarking on research in a field where I had no prior background was daunting, yet I learned to embrace new concepts and face difficulties head-on. Despite the numerous tough moments, I have grown both personally and academically. I also want to acknowledge the mental health and family issues I encountered during this period. These experiences tested my resilience, but with the support of my teachers, family, and my own determination, I managed to persevere and accomplish my goals. To my family, your love and understanding have been my anchor, providing me with the strength to continue. Lastly, I would like to thank myself for staying committed and pushing through every obstacle. This achievement is a testament to the support from my supervisors, family, and my own perseverance. Thank you all for being a part of this journey.

## Table of Contents

This page is intentionally left blank.

**List of Figures**

This page is intentionally left blank

**List of Tables**

This page is intentionally left blank

# Chapter 1 Introduction

## 1.1 Background

In unfamiliar environments, SLAM, as described by Cadena et al. (2016) , is a critical technology that enables robots to navigate and map the area. Distributed multi-robot collaboration, as suggested by Xu et al. (2022), can effectively address many issues in extensive settings. Multi-robot cooperative mapping may offer greater scalability and efficiency than single-robot mapping for tasks that cover broad areas or require quick response, such as in transportation and rescue operations (P.-Y. Lajoie et al., 2021; Nieto-Granda et al., 2014; Rone & Ben-Tzvi, 2013). In multi-robot settings, it is essential for robots to integrate all available data to create a unified global map and achieve accurate self-localization within it (Denniston et al., 2022; Z. Wang et al., 2007) . Benefits of multi-robot SLAM include accelerated mission completion and increased system robustness against individual robot failures, though these advantages require a complex system with coordinated robot cooperation. Precise map integration is vital, necessitating the resolution of transformations among robots and the enhancement of local SLAM accuracy. Inter-robot loop closures allow for deriving transformations between robots from pose measurements (Do et al., 2020; Dubois et al., 2022; X. Zhang et al., 2021) . Detecting these closures and establishing precise inter-robot measurement constraints with perception-derived descriptors is crucial. The main

tasks involve using these descriptors to effectively identify inter-loop closures, setting up precise measurement constraints, and optimizing transformations from any robot frame to the base frame while refining local SLAM. DiSCo-SLAM (Huang et al., 2022) introduces a multi-robot LiDAR SLAM system that employs a two-stage global-local graph optimization to determine inter-robot transformations and each robot's local pose. Nonetheless, the efficiency of translating inter-robot constraints into virtual intra-robot constraints in local pose graphs requires further evaluation.

To enhance local SLAM accuracy, numerous multi-sensor fusion SLAM initiatives based on LiDAR have been developed to achieve a more stable and robust system (L. Chang et al., 2019; He et al., 2021; Hening et al., 2017; Qian et al., 2017; Su et al., 2021a; Tang et al., 2015) . Integrating sensors that are insensitive to environmental factors helps reduce long-term error accumulation. The Global Navigation Satellite System (GNSS) plays a crucial role in global positioning. Real-Time Kinematic (RTK) can achieve centimeter-level accuracy in open environments (Shen et al., 2019; Zangenehnejad & Gao, 2021) . However, GNSS is susceptible to reflection and obstructions, specifically Non-Line of Sight (NLOS) receptions, which can cause positioning errors greater than 10 meters in urban settings like Hong Kong using commercial receivers (Hsu, 2018; Wen et al., 2021). Numerous frameworks already integrate GNSS with LiDAR and IMU, but

all these methods merge every received GNSS measurement. The strategy for adaptive integration of GNSS remains unclear.

## 1.2 Previous works

### 1.2.1 Multi-robot simultaneous localization and mapping

The implementation of multiple-robot SLAM is predicated on the understanding that collaborative robots can expedite and enhance the accuracy of exploration and mapping tasks compared to solitary robotic endeavors. This collective approach to SLAM offers advantages such as accelerated mission completion and resilience against the malfunctioning of individual robots. However, these advantages necessitate a sophisticated system characterized by significant coordination and collaboration among the robots. In contemporary contexts, robots find extensive applications across industrial, military, and domestic environments. Mobile robots, which include categories such as cleaning, entertainment, and demining robots, are often deployed in substantial numbers, necessitating dependable perception capabilities to fulfill their designated functions. Certain tasks, particularly those requiring rapid and autonomous execution like rescue missions in forested or urban settings, security, surveillance, and maintenance investigations as outlined by Saeedi et al. (2016 ), rely heavily on proficient localization and mapping. Distributed robotic systems enhance robustness, since the failure of a single robot does not compromise the entire operation (Birk & Carpin, 2006).

SLAM methodologies vary and can be classified based on the primary sensors employed and the optimization techniques utilized. Predominant SLAM technologies include visual SLAM and LiDAR SLAM. Main SLAM algorithms are distinguished based on their data-processing algorithms, encompassing filtering SLAM, smoothing SLAM, and AI-driven SLAM approaches. The technique of computing the relative motion from successive LiDAR scans is termed LiDAR odometry or scan matching. Smoothing SLAM employs methods to estimate the complete trajectory, and Graph-SLAM represents one of the overarching methodologies in this category. Within Graph-SLAM, the robot's poses are denoted as nodes in a graph, with edges representing motion and observation constraints. These constraints are then optimized to define the spatial distribution of the nodes. Graph-SLAM involves optimizing a system of equations to minimize the errors induced by these constraints. Subsequently, researchers have introduced various solutions, including incremental smoothing and mapping (iSAM) (Kaess, Ranganathan, & Dellaert, 2007) which is prevalently utilized in LiDAR-based SLAM.

Conventional LiDAR-based mapping tasks that require covering an entire environment alone often lead to prolonged construction periods and significant demands on computational resources. Multi-robot cooperative SLAM can mitigate the costs, global error accumulation, computational burden, and risk concentration

that are typical of single-robot SLAM approaches, providing a more robust and stable framework. Generally, multi-robot SLAM algorithms are extensions of single-robot SLAM algorithms; however, the inclusion of multiple robots introduces additional complexities. In multi-robot SLAM systems, the type of data shared among agents and the methodologies for processing this data are crucial considerations. Data interchange between robots is facilitated through communication channels where bandwidth plays a critical role in SLAM performance. Historical approaches to collaborative SLAM are typically categorized by whether they involve the sharing of raw sensor data or processed data. Sharing raw sensor data offers greater flexibility but demands high bandwidth, whereas sharing processed data places lesser demands on processing power and bandwidth. Multi-robot systems may be centralized or distributed; in distributed systems, computational tasks are divided among the robots within the team. The most challenging aspect of multi-robot SLAM involves resolving the relative poses of the robots and integrating all local maps created by individual robots into a coherent global map. Each local map is derived from the local SLAM, based on coordinated measurements. The challenge of determining relative robot poses is intertwined with the issue of multi-robot data association. Once data association and correspondence between duplicate locations across robots are established, the relative poses are updated, and the consolidated global map is refined. The primary

objective of multi-robot SLAM is to accurately determine the transformations between different robots. Key elements of this task include the use of perception-derived descriptors for effective detection of inter-loop closures, establishing precise inter-robot measurement constraints, and optimizing transformations from any robot frame to the base frame.

### 1.2.2 3D LiDAR based multi-robot cooperative SLAM

3D LiDAR-based SLAM offers distinct advantages in providing intuitive map representation, resistance to changes in illumination or environments lacking textual cues, and the capability to capture detailed spatial information over extended distances (Khan et al., 2021; Su et al., 2021b; B. Zhou et al., 2021) . 2.

Over recent years, a range of studies has emerged focusing on single robot SLAM utilizing 3D LiDAR sensors, notable among these are LOAM (Lidar Odometry and Mapping) (Zhang & Singh, 2014) and LIO-SAM (Tightly coupled Lidar Inertial Odometry via Smoothing and Mapping) (Shan et al., 2020). Various experimental assessments have affirmed the efficacy of these algorithms. The realm of multi-robot SLAM has seen considerable progress in recent years, with emerging research increasingly concentrating on LiDAR-based systems and collaborative methodologies. From its foundational applications in offline contexts, this technology has advanced to enable real-time operations in multi-robot SLAM.

This progression facilitates the rectification of accumulated positional errors and the prompt update of maps, thereby enhancing overall accuracy.

Furthermore, multi-robot SLAM frameworks are proposed using the single SLAM as local SLAM(Cao & Beltrame, 2021; Y. Chang et al., 2022; P. Y. Lajoie & Beltrame, 2024; Tian et al., 2022). The first approach to addressing the online multi-robot SLAM challenge with 3D LiDAR was introduced by (Dubé et al., 2017) through SegMap. This system integrates odometry factors derived from wheel encoders and IMU data via an Extended Kalman Filter (EKF). Further expanding on multi-robot SLAM, Zhou et al. (2022) developed an online system that combines range measurements from UWB sensors with LiDAR data across multiple mobile robots. This system is structured with a UWB localization module hosted on a cloud server, individual robot modules on each mobile robot, and a master module on a master computer. The UWB localization module is tasked with initializing the UWB coordinate system during the initial epoch and updating the coordinates of the UWB sensors at subsequent epochs. Each mobile robot's module calculates the transformation relationship between the LiDAR and UWB coordinate systems, facilitating the conversion of LiDAR data from its native coordinate system to the UWB system. The Master module aggregates trajectory estimations and point cloud maps from all participating robots, merging them into

a globally consistent map. This architecture enables the robots to accomplish cooperative mapping without the necessity of direct interaction.

Huang et al. (2022) introduced a distributed multi-robot SLAM framework designed for real-time applications using 3D LiDAR. Upon receiving a LiDAR scan, the system concurrently activates the local SLAM thread and each map fusion thread. The framework employs LIO-SAM (Shan et al., 2020) as the local SLAM protocol. Initially, the local SLAM node estimates the pose transformation for each individual robot, followed by the transmission of key frame information to each robot's fusion node. This information encompasses pose estimations and the entire scan, which is then forwarded to the map fusion node. The map fusion node employs the point cloud data from the key frames to publish a lightweight spatial feature descriptor known as Scan Context (Kim & Kim, 2018) . Scan Context messages are generated by projecting the LiDAR scan onto a 2D plane, subsequently dividing the resulting 2D scan image into grid cells aligned with a predefined number of sectors (Ns) and rings (Nr). To extract a ring key feature, the maximum intensity value of all points within a cell is calculated. For each ring, a feature of dimension Nr is derived by counting the non-zero values in each cell. A KD tree for ring keys is constructed to facilitate the search for nearest neighbor descriptors and acquire a list of nearest indices. The Scan Context features are shifted along the sector axis to maintain rotation invariance, with the shifting angle

also providing an initial rotation estimate for the Iterative Closest Point (ICP) scan-matching process when no prior coordinate transformation history exists. To address potential errors in loop closure recognition, which may occur due to environmental similarities or repetitive object patterns, the Pairwise Consistent Measurement Set Maximization (PCM) technique (Mangelson et al., 2018) is applied. Following the application of PCM, the accepted inter-robot loop closures are utilized to refine the transformations from local robots to the global coordinate frame. This global optimization step treats the transformations derived from inter-robot loop closures as measurements and employs the Georgia Tech Smoothing and Mapping library (GTSAM) for optimization. Separator poses, which are critical poses within the multi-robot system, facilitate connections between different robots. After the global optimization phase, inter-loop closures are transformed into virtual intra-robot loop closures. To compute the relative poses among these virtual closures, the related poses are multiplied with the relative transformation from the inter-robot loop closure. Figure 1.1 delineates the transition between global optimization and local SLAM in a two-robot scenario. The local SLAM communicates the LiDAR scan and pose estimation to the map fusion node, which, after detecting an inter-loop closure and optimizing transformations among different robots, transfers these inter-loop constraints to the local pose graph for further refinement by establishing transformations from global

to local coordinates. Subsequent to integrating the virtual intra-robot observation, the local SLAM incorporates the pose constraints between the pertinent key frame and the local pose graph, which is then optimized. The indices of the related key frames and constraints are derived from the inter-loop queues and dispatched by the map fusion node.

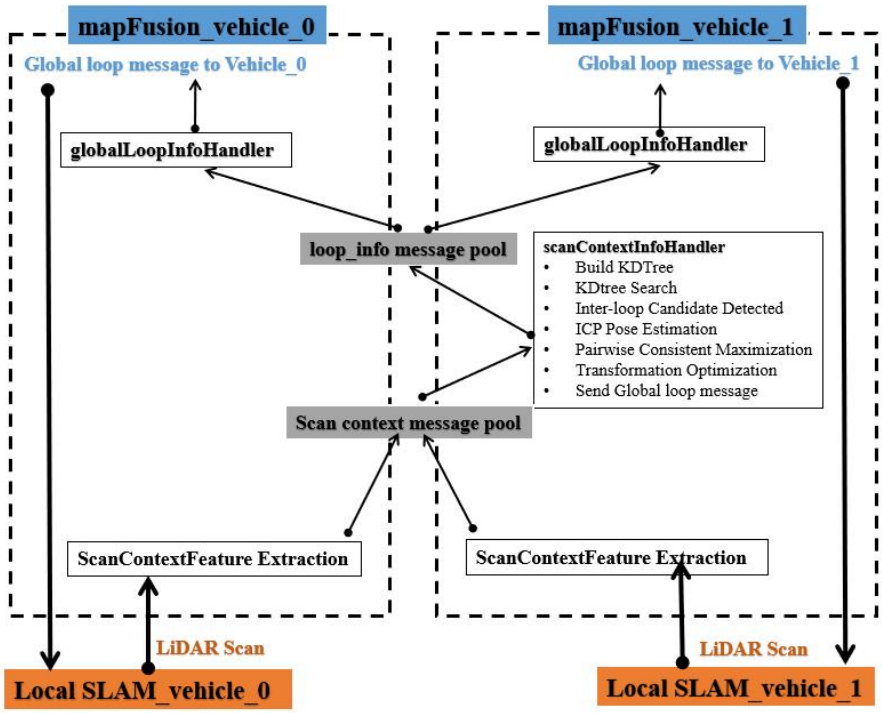



Figure 1.1 Inter-robot constraints to local SLAM.

Furthermore, Zhong et al. (2022) introduce DCL-SLAM, a fully distributed collaborative LiDAR SLAM framework specifically crafted for robotic swarms navigating uncharted territories with minimal information exchange. This framework is composed of three primary elements: a versatile front-end LiDAR odometer; a distributed loop closure module capable of detecting overlaps among

robots; and a distributed back-end module that optimizes the graph to adjust positions collaboratively and eliminate erroneous loop closures. The framework demonstrates considerable flexibility and adaptability by being compatible with various front-end LiDAR odometries. Additionally, it offers superior accuracy and reduced bandwidth requirements compared to other contemporary multi-robot LiDAR SLAM systems, making it particularly suitable for robotic swarms in scenarios constrained by communication limitations. To comply with these communication restrictions, a compact global descriptor, LiDAR-Iris (Y. Wang et al., 2020), is integrated to efficiently describe laser scans and identify loop closures without necessitating the exchange of additional raw data. Subsequently, a three-stage distributed loop-closure detection and outlier management process, based on the Pairwise Consistent Measurement Set Maximization (PCM) approach, is implemented to secure relative pose transformations.

Wu et al. (2022) contribute to the field by developing a map merging method for collaborative LiDAR-based SLAM that utilizes GPS measurements alongside an enhanced Iterative Closest Point (ICP) algorithm. While effective, the ICP algorithm is known for its slow convergence and propensity for local optimization pitfalls. Additionally, it requires precise initial positioning between the point clouds to be registered, where GPS provides a superior initial positioning solution for widely spaced point clouds. The overlap region for machine scanning

is initially ascertained within the Euclidean space defined by GPS coordinates, and the initial attitude transformation matrix between the robots is calculated to serve as the starting point for the ICP process. An improved version of the traditional ICP algorithm then accurately localizes the point cloud in the overlapping region using KD-tree structures and normal vectors. The transformation matrix derived from this precise registration is applied to rotate and translate the source local point cloud, thereby assembling the comprehensive global point cloud map. In practical experiments, the initial transformation matrix, ascertained through GPS measurements, furnishes an advantageous starting attitude for the ICP algorithm, thus diminishing the time requirement and enhancing the algorithm's accuracy.

### 1.2.3 Multi-Sensor fusion for the 3D LiDAR SLAM

In environments characterized by high dynamism or sparse features, the performance of LiDAR-based SLAM systems diminishes, impacting the quality of localization and mapping outcomes. Over recent years, there has been a notable increase in multi-sensor fusion SLAM research employing LiDAR to achieve systems that are more stable and robust. Concurrently, the proliferation of 3D LiDAR technology and advances in the computational capabilities of embedded processors have accelerated the development of positioning technologies based on 3D LiDAR (Xu et al., 2022). Additionally, the integration of odometry data with 3D LiDAR has enhanced positioning accuracy, establishing it as a predominant

sensor in various applications including autonomous vehicle navigation and robotic autonomous navigation. To achieve real-time, high-precision estimation of 6-DOF (degrees of freedom) states for mobile robots, researchers have explored both vision-based and LiDAR-based SLAM methodologies, which represent the principal single-sensor SLAM approaches currently being advanced. However, reliance on a single sensor type presents inherent limitations. Vision sensors, for instance, are highly susceptible to lighting variations, which can lead to system instability. Similarly, the sparse data output by LiDAR can lead to a rapid decline in localization capabilities in highly dynamic environments with limited textural information. Moreover, rapid motion patterns and cumulative errors can compromise the reliability of odometry data. Consequently, the inclusion of other sensors, such as IMUs (Inertial Measurement Units), GPS (Global Positioning System), and UWB (Ultra-Wideband), which are less sensitive to environmental factors, has been proposed to mitigate these challenges (J. Zhang et al., 2022).

The inception of loosely coupled systems marked a new phase in the evolution of multi-sensor fusion systems, appearing prominently in earlier research. These systems are primarily categorized into two types: the LiDAR-IMU loosely coupled system and the LiDAR-Visual-IMU loosely coupled system. Despite their early development, these systems did not adequately mitigate the measurement bias inherent in IMUs, with the IMU serving primarily as an ancillary component.

A seminal contribution in LiDAR-based 3D SLAM is Zhang's development of the LOAM algorithm, notable for its efficient extraction of edge and planar features from intricate point clouds. This approach involves using distances from points to lines and planes to formulate a loss function, optimizing the pose transformation. Additionally, the IMU provides an initial pose estimation to enhance the accuracy of the laser odometry. However, this initial framework lacked capabilities for loop closure detection and back-end optimization, areas that have seen significant advancements in subsequent studies. The loosely coupled inertial system primarily processes IMU data for point cloud distortion correction and provides initial pose estimates, though the impact of sensor fusion within this framework remains constrained. The primary innovations lie in refining accurate front-end matching and back-end optimization techniques. While not groundbreaking in terms of data fusion, this phase significantly propelled the development of SLAM technologies and experimented with early sensor fusion stages. Loosely coupled systems offer advantages in real-time processing and lower computational complexity, yet their accuracy under high-speed motion remains challenging to maintain. IMUs, with their high-frequency motion response characteristics, have become essential in mobile robotics. Notably, tightly coupled systems that incorporate IMU data have achieved significant progress in visual odometry (Forster et al., 2015) . This research deduced the IMU pre-integration formula, error transfer model, and

defined residual terms, profoundly influencing the subsequent development of LIO and VIO systems. With these formulations, the interaction between the IMU and the world coordinate system can be decoupled during joint optimization, allowing for the updating of IMU biases to optimize data integration (Xu et al., 2022). The developers of LeGO-LOAM subsequently introduced LIO-SAM (Shan et al., 2020), which constructs LiDAR-inertial odometry on a factor graph, incorporating multiple relative and absolute measurements, including closed loops, as factors in the system. Furthermore, the system integrates a GPS absolute positioning factor to correct long-term system drift. Nevertheless, since feature extraction depends on geometric characteristics of the environment, this approach struggles to operate effectively over extended periods in open settings. With advancements in IMU pre-integration theory, the coupling between LiDAR odometry systems and IMUs has strengthened, enhancing the positioning accuracy of SLAM systems. However, tightly coupled systems involve extensive computations, and finding a balance between processing speed and accuracy remains a critical challenge.

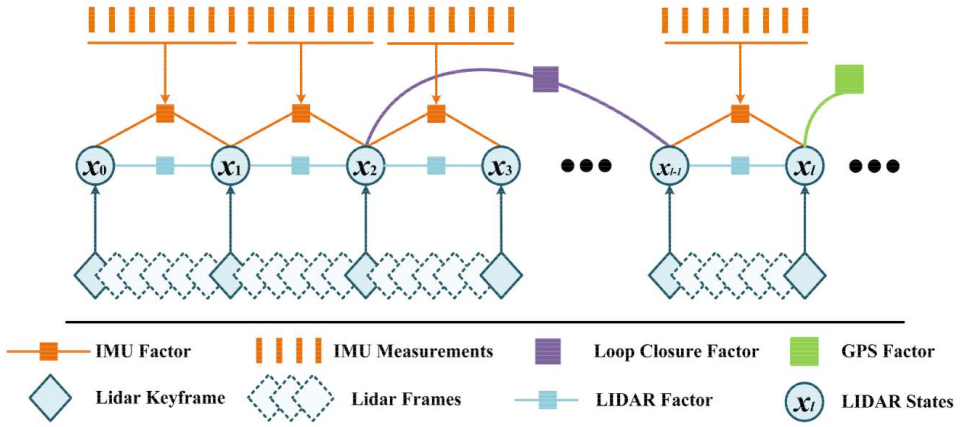



Figure 1.2 Factor graph structure of LIO-SAM.

The fusion of GNSS and IMU is proposed in many frameworks. It is critical to mitigate performance degradation from the GNSS anomalous solutions and to limit IMU bias in time during GNSS downtimes. Some works using the indicators output by the GNSS receivers like the number of satellites which is not enough for quality inspection of GNSS (G. Zhang & Hsu, 2018). Li et al. (2018) adopted Outlier-resistant Ambiguity Resolution (AR) to reject the outlier based on residua. However, this method can easily perform false-negative judgement. The integration of GNSS-RTK and LIO is also a reasonable direction to correct the accumulated drift of local sensors and provide globally referenced positioning (J. Gao et al., 2024; X. Li et al., 2023; Liu et al., 2024). Gao et al. (2015) using GPS and LiDAR to correct the error of IMU via EKF estimator. Chiang et al. (2020) adopted two EKFs to provide velocities ad fused position estimates respectively. The former one is derived from LiDAR and IMU and the later one is estimated

from GNSS and IMU. The EKF-based method is extensively used; however, this method fails to take historical information into consideration due to the principle of the Markov chain which results in unsatisfactory precision. In conclusion, all the above methods are crude fusions of all received GNSS measurements. It is not possible to eliminate as much as possible the errors caused by gross mismeasurements.

## 1.3 Research objectives

(1) We intend to propose a GNSS-RTK availability assessment method, during which, unreliable GNSS measurements rejections are performed through mean elevation mask evaluation using the cooperatively registered 3D map that was incrementally constructed by DISCO-SLAM (Huang et al., 2022). Then, we intend to propose a GNSS-RTK/LIO integration scheme based on FGO. The global positioning from GNSS-RTK, IMU, and relative pose estimation are integrated using the FGO to perform the local SLAM optimization.

(2) The effectiveness of the inter-robot constraints in local pose optimization will be evaluated on two typical urban scenarios collected in Hong Kong. Furthermore, the transmission principle and mechanism of the virtual intra-robot constraints in local pose graph will be examined. The performance and

reasons for different scenarios will be discussed in detail, and different optimization methods will be proposed based on these scenarios.

## 1.4 Thesis outline

Chapter 1 provides the background and motivation for the research. Chapter 2 introduces the key notations and definitions used throughout the thesis. Chapter3 presents the methodology for integrating GNSS-RTK with LIO. It details the adaptive sensor fusion pipeline developed to enhance positioning accuracy in urban environments. Chapter 4 explores the two-stage graph optimization approach, focusing on the evaluation of inter-robot constraints into local pose graph optimization and highlighting the potential benefits and limitations. The final chapter summarizes the research contributions, findings, and implications of the thesis.

## Chapter 2 Key notations and definitions

The definitions related to the research questions are given in this chapter and used consistently throughout the thesis. For convenience, the key notations of the thesis are given firstly (shown in Table 2.1)

Table 2.1 Key Definitions and Notations.

| Variable | Description |
|---|---|
| $\mathbf{x}_i$ | A 6DoF robot pose. $\mathbf{X} = \{\mathbf{x}_0, ..., \mathbf{x}_i\}$ represent a set of 6DoF robot poses spanning from time 0 to time $i$, where $\mathbf{X} \subset \mathrm{SE}(3)$. $\mathbf{X}_\alpha$ represents the poses of robot α. |
| ${}_\alpha\mathbf{X}_{\alpha_i}$ | The pose of robot α at time $i$ in robot α's local coordinates. |
| $\mathbf{z}_{ij}$ | The observed transformation between $\mathbf{x}_i$ $and$ $\mathbf{x}_j$. |
| $\mathbb{X}$ | The collection of the poses of all n robots, $\mathbb{X} = \{\mathbf{X}_\alpha \| \alpha \in \mathcal{N}\}$. |
| $\mathbf{e}_{ij}$ | The discrepancy between the observed transformation and the expected transformation. |
| $\mathbf{z}_{ij}$ | The observed transformation between $\mathbf{x}_i$ $and$ $\mathbf{x}_j$. |
| $\mathbf{z}_{\alpha_i\beta_j}$ | An inter-robot transformation measurement from the separator |

poses $\left\langle \mathbf{X}_{\alpha_i} \quad , \mathbf{X}_{\beta_j} \right\rangle$

$\mathbf{z}_{\alpha_i \alpha_k}$ An virtual intra-robot transformation measurement from two distinct sets of separator poses $\left\langle \mathbf{X}_{\alpha_i} \quad , \mathbf{X}_{\beta_j} \right\rangle$ and $\left\langle \mathbf{X}_{\alpha_k} \quad , \mathbf{X}_{\gamma_l} \right\rangle$

$\mathbf{T}_{\beta\alpha}$ The transformation which converts the poses in robot α's local coordinates to robot β's local coordinates.

$\mathbf{g}_k$ The GNSS-RTK measurement transformed into the East, North, Up frame and denoted as $\mathbf{g}_k = \left[x_{\mathbf{g}_k}, y_{\mathbf{g}_k}, z_{\mathbf{g}_k}\right]^T$.

$\mathbf{f}_i$ A point on the 3D LiDAR map and denoted as $\mathbf{f}_i = \left[x_{\mathbf{f}_i}, y_{\mathbf{f}_i}, z_{\mathbf{f}_i}\right]^T$.

$el_{\mathbf{f}_i}$ The elevation angle of the point on the 3D LiDAR map.

$el_{\mathbf{g}_k}$ The mean elevation angle for the GNSS-RTK measurement $\mathbf{g}_k$

---

**Definition 1.** Simultaneous Localization and Mapping (SLAM). SLAM is a technique where a robot or autonomous system builds a map of an unknown environment while simultaneously keeping track of its own location.

**Definition 2.** Global Navigation Satellite System Real-Time Kinematic(GNSS-RTK). GNSS-RTK is a satellite navigation technique used to enhance the precision

of position data derived from GNSS systems.

**Definition 3.** Factor Graph Optimization (FGO). FGO is a mathematical framework used in SLAM to optimize the robot's trajectory by minimizing the errors in a graph. A pose graph is a graph-based representation of the SLAM problem where nodes correspond to robot poses (positions and orientations) and edges correspond to spatial constraints or relative transformations between these poses.

**Definition 4.** Absolute Pose Error (APE). APE measures the difference between the estimated pose and the ground-truth pose of a robot. It is commonly used to evaluate the accuracy of SLAM algorithms by quantifying the deviation of the estimated trajectory from the true trajectory.

**Definition 5.** Inter-Robot Constraints. Inter-robot constraints are measurements between different robots in a multi-robot SLAM system. These constraints typically derived from observations when robots detect and recognize each other or share common landmarks.

**Definition 6.** Intra-Robot Constraints. Intra-robot constraints are measurements within a single robot in a multi-robot SLAM system. These constraints are derived from the robot's own sensors, such as LiDAR, IMU, and odometry data.

# Chapter 3 GNSS-RTK adaptive integration with LIO

## 3.1 Introduction

Multi-robot SLAM has advantage in robustness and scalability which is widely used in time-sensitive and large-scale mapping. The loop closure of each robot is crucial in drift elimination but is absent in long distance mapping. Considering the absence of intra-robot constraints of each robot, combining globally referenced positioning where GNSS plays an important role is promising. In previous studies, there are many SLAM frameworks that rely purely on LiDAR and IMU data fusion due to the application scenarios such as underground or underwater GPS-Denied scenarios. Regarding to those brutely fusing all the received GNSS solutions and LiDAR/IMU where GNSS is available, the abnormal measurements will lead to limited precision from the source. How to identify the quality of GNSS solutions and perform the adaptive integration and effective fusion remains an open question to address. This chapter adopted a GNSS-RTK selection method, during which the MEA is used to select those solution from relatively open space. The process of calculating the MEA and the validation on two typical urban street datasets is delivered in following paragraph.

## 3.2 Methodology

### 3.2.1 Local map extraction

The initial procedure involves the extraction of a local map. Each GNSS-RTK solution, resolved using RTKLIB under the Earth-centered, Earth-fixed (ECEF) coordinate system, is transformed into the East, North, Up (ENU) frame via GeographicLib. In the context of single robot SLAM, the local map proximal to the GNSS-RTK position is derived from the global map. Due to the limitations of LiDAR technology, the local map is defined in the east and north directions by fixed distance parameters. Conversely, in the vertical direction, the inclusion of points is comprehensive to accurately represent the density of obstacles encountered. By leveraging the integrated map progressively constructed through cooperative SLAM, the extraction of the local map from the global map is enhanced to incorporate additional components critical for assessing the availability of GNSS-RTK. (Figure 3.1)

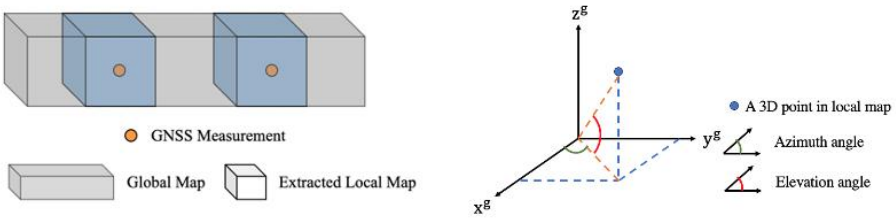



Figure 3.1 GNSS-RTK acquisition point openness assessments.

### 3.2.2 Calculation of mean elevation angle

The subsequent stage involves the segmentation of this area into subregions, considering uniformity and stability. The 360-degree area surrounding the GNSS rover is segmented into 36 subregions based on azimuth angle (Figure 3.1); for instance, points with an azimuth angle less than 10 degrees are assigned to the first subregion, while those with an angle greater than 10 degrees but less than 20 degrees are allocated to the second subregion. It is important to note that the reliability of GNSS measurements remains unknown until the completion of an evaluation. As a result, rather than measuring the relative distances between the GNSS rover and the objects detected, the temporally nearest pose, as determined by local FGO, is utilized. The coordinates of the points within the local map will be adjusted so that the GNSS rover serves as the origin.

For the k-th GNSS-RTK $\mathbf{g}_k = \left[x_{\mathbf{g}_k}, y_{\mathbf{g}_k}, z_{\mathbf{g}_k}\right]^T$ , the corresponding position determined by local FGO is denoted by $\tilde{\mathbf{g}}_k = \left[x_{\tilde{\mathbf{g}}_k}, y_{\tilde{\mathbf{g}}_k}, z_{\tilde{\mathbf{g}}_k}\right]^T$ . Regarding a point $\mathbf{f}_i = \left[x_{\mathbf{f}_i}, y_{\mathbf{f}_i}, z_{\mathbf{f}_i}\right]^T$ on the local map, its azimuth angle (Figure 3.1) denoted as $az_{\mathbf{f}_i}$ is calculated as follows:

$$az_{\mathbf{f}_i} = \arctan\left(\left|y_{\tilde{\mathbf{g}}_k} - y_{\mathbf{f}_i}\right| / \left|x_{\tilde{\mathbf{g}}_k} - x_{\mathbf{f}_i}\right|\right) \tag{1}$$

The elevation angle of the point (Figure 3.1), expressed as $el_{\mathbf{f}_i}$ is defined as follows:

$$el_{\mathbf{f}_i} = \arctan\left(\left|z_{\tilde{\mathbf{g}}_k} - z_{\mathbf{f}_i}\right| / \sqrt{\left(x_{\tilde{\mathbf{g}}_k} - x_{\mathbf{f}_i}\right)^2 + \left(y_{\tilde{\mathbf{g}}_k} - y_{\mathbf{f}_i}\right)^2}\right) \tag{2}$$

For all the points within one subregion, their elevation angles are arrayed in ascending order, and the maximum is selected to represent the elevation of that subregion. Subsequently, the mean of all the subregions expressed as $el_{\mathbf{g}_k}$ , is utilized as the MEA for the GNSS-RTK measurement $\mathbf{g}_k$. Given a threshold $el_{th}$, the assessment of GNSS-RTK availability is ultimately determined by the following method. If the MEA is higher than the threshold, the GNSS-RTK solution will be rejected because of low reliability.

$$\mathbf{g}_k: \begin{cases} el_{\mathbf{g}_k} > el_{th}, & \text{unreliable} \\ el_{\mathbf{g}_k} < el_{th} & \text{reliable} \end{cases} \tag{3}$$

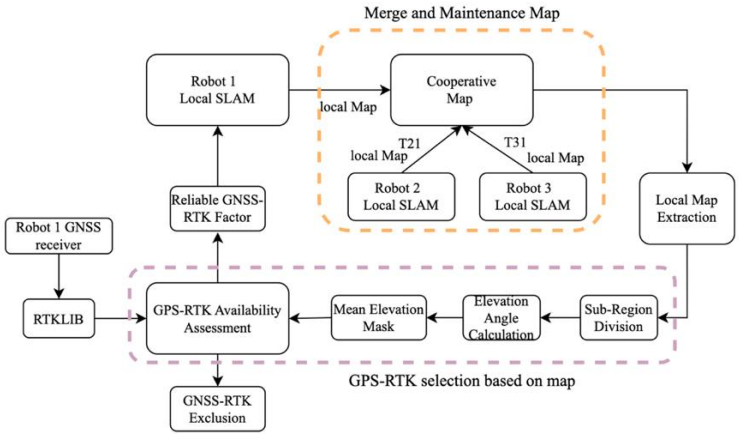



Figure 3.2 GNSS-RTK availability assessment.

### 3.2.3 GNSS-RTK/LIO Fusion

The sensor fusion framework is designed to solve the motion states of the object. Various methodologies for addressing state estimation challenges have been thoroughly explored and can generally be classified into filter-based and non-linear optimization-based approaches. The latter approach is often framed as a non-linear least squares problem. Specifically, the state estimation issue can be formulated as a Maximum A Posteriori (MAP) problem, which is subsequently transformed into a Maximum Likelihood Estimation (MLE) using Bayes' Theorem. Assuming a Gaussian noise model, this is ultimately expressed as a least squares problem through a negative logarithmic transformation.

Each measurement utilized for state estimation can be represented as a function of the states. The term for least squares, often referred to as the residual term, quantifies the Mahalanobis distance between the known measurements and their approximations derived from the states. These residuals, which are expected to be minimal, impose constraints on the states. The objective function in non-linear optimization comprises the sum of all these residuals. A set of states that minimizes this objective function is deemed the optimal solution for the estimator.

The estimated states encompass the 6D poses, captured at the estimation frequency of LIO, and are represented as,

$$\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{w-1}\}$$

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{R}_k & \mathbf{p}_k \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \in \mathrm{SE}(3), k = 0, \cdots, w-1$$

Where $w$ denotes the window size, $\mathbf{R}_k \in$ Special Orthogonal Group (SO(3)) represents the relative rotation, and $\mathbf{p}_k$ denotes the translation. The state estimation framework based on non-linear optimization can be conceptualized as a factor graph, commonly referred to as FGO (Figure 3.3). Within this framework, the states under estimation are represented as vertices. The measurements, which impose constraints on the states, are depicted as edges and are termed factors within the graph. The objective of FGO is to identify a set of states that optimally conform to all the constraints represented by the edges.

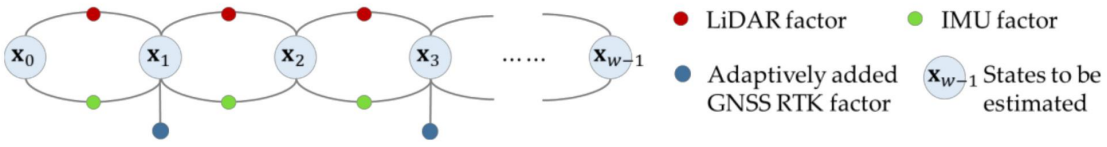



Figure 3.3 The factor graph of the adaptive GNSS/LiDAR/IMU fusion procedure.

The LiDAR factor, developed through scan-matching, serves as relative motion constraints namely known measurements as reference. Whereas those derived from the states under estimation are treated as observed measurements. The FGO is specifically engineered to minimize the discrepancies between these

measurements. The relative motion transformations denoted as $\mathbf{T}_{l_k}$ are calculated through the scan-matching process.

$$\mathbf{r}_{\mathcal{L},k}\left(\mathbf{T}_{l_k}^{l_{k-1}}, \mathbf{X}\right) = \left(\mathbf{T}_{l_k}^{l_{k-1}}\right)^{-1} \cdot \left(\mathbf{x}_{k-1}{}^{-1} \cdot \mathbf{x}_k\right) \tag{4}$$

The IMU factor, originating from pre-integration, is formulated similarly. The motion increment yielded by IMU pre-integration is regarded as the known measurement, while the increment derived from the states under estimation is treated as the observed measurement. The discrepancy between these two is characterized as follows: $\mathbf{Z}_{b_k}^{b_{k-1}} \in \mathrm{SE}(3)$ represents the rotation and translation increments calculated from IMU raw data.

$$\mathbf{r}_{\mathcal{B},k}\left(\mathbf{Z}_{b_k}^{b_{k-1}}, \mathbf{X}\right) = \left(\mathbf{Z}_{b_k}^{b_{k-1}}\right)^{-1} \cdot \left(\mathbf{x}_{k-1}{}^{-1} \cdot \mathbf{x}_k\right) \tag{5}$$

If the GNSS-RTK evaluation module assesses that the GNSS-RTK solution is reliable, GNSS factors are incorporated into the factor graph and constrains the plane coordination of the state. In this formulation, $\mathbf{g}_j$ represents the known position provided by GNSS-RTK. The discrepancy between the estimated position and $\mathbf{g}_j$ is anticipated to be minimal. In scenarios where a commercial level GNSS receiver is utilized to acquire location measurements in urban canyons, only 2D residuals corresponding to the "earth" and "north" directions are integrated into the global optimization process.

$$\mathbf{r}_{\mathcal{G},j}(\mathbf{g}_j, \mathbf{X}) = \mathbf{g}_j - \mathbf{p}_j, j \in [0, w) \quad (6)$$

## 3.3 Experiment setup and datasets

To improve the accuracy of multi-robot SLAM, we propose to integrate the selected GNSS with high fidelity to the multi-sensor fusion framework. With the local map constructed by the lidar , we can infer the degree of openness around it. We consider GNSS in open areas to be exploitable and we use the metric named MEA to measure the openness of a place. The MEA is calculated by two steps. A total of 360 degrees around the GNSS rover is divided into 36 subregions according to the azimuth angle. Considering all the points in one subregion. In each subregion, the max value of elevation angle is taken as the elevation angle of this subregion. Then the MEA of all the subregions can be calculated.

### 3.3.1 Experiment setup

To evaluate the effectiveness of the proposed methodology, an experiment utilizing datasets collected in the urban canyons of Hong Kong was conducted. The equipment setup included a Velodyne HDL-32E and an Xsens Ti-10 IMU for collecting 3D point clouds at a frequency of 10 Hz and acceleration and angular velocity data at 200 Hz. The lidar system was configured with a 360-degree horizontal field of view (HFOV), a vertical field of view (VFOV) ranging from

+10 to -30 degrees, a maximum ranging capability of 80 meters, and an operational frequency of 10Hz. A central 3D lidar, supplemented by two slanted lidars, was utilized for data acquisition. This setup facilitated the construction of a comprehensive point cloud map from multiple lidars. Additionally, a commercial-grade u-blox GNSS receiver was used to collect raw GNSS measurements, which were processed into GNSS-RTK solutions using RTKLIB. For ground truth validation, the NovAtel SPAN-CPT, an integrated GNSS RTK/INS (fiber-optic gyroscopes, FOG) navigation system, was employed. All data were processed and synchronized using ROS and were stored as topics in rosbag. The research methodology utilized this collected data to merge and align timestamps, providing


Figure 3.4 Experimental setup of the LiDAR.

a dataset for the evaluation of multi-vehicle cooperative SLAM.

To calculate the MEA of the surrounding local map, it is needed to extract the local map from the whole map constructed by the LiDAR. The whole map is constructed by the registered LiDAR scan. Each LiDAR scan can be captured by

centrally located LiDAR or the merged scan from the main LiDAR and two slant LiDAR placed on either side of the main LiDAR (Figure 3.4). There is difference between the MEA calculated by Multi LiDAR and Single LiDAR, because the slant LiDAR can expand the VFOV, which enables scanning higher buildings to construct more comprehensive point cloud maps. Because multi-LiDAR systems can cover a larger area, the MEA calculated from point cloud maps constructed using them tends to be larger than those calculated from single lidar systems. To mitigate the interference of roadside trees from adjacent vehicles on computing the MEA, a preprocessing step was implemented to filter out low-height point clouds before calculating the MEA.

### 3.3.2 Datasets

The dataset employed in verifying the proposed method is summarized in Table 3.1. As depicted in Figure 3.5, data collection occurred in Whampoa and Tsim Sha Tsui, an area characterized by dense buildings in Hong Kong.

In the urban-1 scenario, both jackal0 and jackal1 start data collection in relatively open spaces to ensure better initial positioning performance. Jackal0 initially enters an area with moderately tall buildings, followed by a section next to an open space where only one side has buildings, and the right side has a relatively clear sky view. The vehicle then travels along an area near the seashore, with taller buildings only on the right side. Subsequently, the vehicle enters a narrow street

where nearly 80% of the sky view is obstructed by buildings, leading to suboptimal GNSS performance. Finally, the vehicle returns to a wider street environment to meet jackal1. Jackal1 begins in an open-sky environment near the seashore to ensure high-quality ground truth. It then passes through a green space, leaving the right side relatively open. Jackal1 then enters a narrow street with two lanes closely flanked by buildings, resulting in a limited sky view due to surrounding structures. Next, jackal1 enters a densely populated residential area with narrow roads and tall buildings, leading to poor GNSS performance. Eventually, jackal1 returns to a more open area, overlapping the route taken by jackal0. The urban-1 scenario experienced by jackal0 and jackal1 represents a deep urban environment where the poor sky view results in unsatisfactory accuracy.

In the urban-2 scenario, jackal0 starts from a relatively open seashore area and enters a wide street environment with minimal obstructions, providing an open sky view and good GNSS accuracy. The vehicle then moves into an environment with mid-rise buildings (approximately 50 m) on both sides of a street of normal width. Next, the vehicle enters the main road, where the west side features a row of uniformly arranged buildings, while the east side has a relatively clear sky view. Jackal1 begins at jackal0's endpoint, then enters a wide street environment, and finally travels through an open area to reach a residential neighborhood, where one

side is obstructed by taller buildings. Overall, this dataset represents a typical urban scenario with a limited sky view, common in cities worldwide.

Figure 3.5 Dataset Urban-1 and Urban-2 (red:jackal0, orange:jackal1)

Table 3.1 The information of the two datasets used for verification in this thesis.

| Dataset | Length(km) | | Urbanization |
|---|---|---|---|
| | Jackal0 | Jackal1 | |
| Urban-1 | 2.55 | 2.50 | Deep |
| Urban-2 | 3.1 | 2.3 | Medium |

### 3.3.3 Evaluation metrics and method

The research utilizes collected data to merge and align timestamps, providing data for evaluating multi-vehicle cooperative SLAM. The evaluation of cooperative SLAM and the GNSS adaptively integrated with LiDAR/IMU pipeline is based on the Root Mean Square Error (RMSE) of APE regarding translation part to the total

poses. We used the EVO package (Grupp, 2017) ) to calculate the APE and matplotlib library to visualize the estimated trajectory and the ground-truth. The 2D APE implemented in EVO is calculated as follows:

$$e_k = \|\mathbf{p}_{k,\text{ estimation}} - \mathbf{p}_{k,groundtruth}\|_2 \quad (7)$$

In the above equation, the $\mathbf{p}_{k,\text{ estimation}}$ and $\mathbf{p}_{k,groundtruth}$ represents the 2D XY estimated pose and the ground-truth. $\|\cdot\|_2$ demotes the 2-norm. The evaluation of the performance of cooperative SLAM are based on the multi-robot cooperative SLAM framework named DiSCo-SLAM. The local SLAM of individual vehicle of DiSCo-SLAM is based on LIO-SAM, where we can integrate GNSS to the graph optimization. The baseline of this research is the result without any GNSS factor, on the other hand, the LiDAR and IMU message will be used.

## 3.4 Experimental results in urban canyon 1

### 3.4.1 Results of GNSS-RTK availability assessment

The result of the baseline is shown in Figure 3.6 and Table 3.2. In Figure 3.6, the blue line represents the estimated trajectory of jackal0 and the yellow one is of jackal1. The dashed line represents the ground-truth for both. It is shown that the accumulative drift is obvious and the RMSE and the Maximum value of the APE is relatively large. Since neither of the two paths returned to the origin, in other

words, there were no intra loop closure constraints, the deviation from the ground truth increased over time.

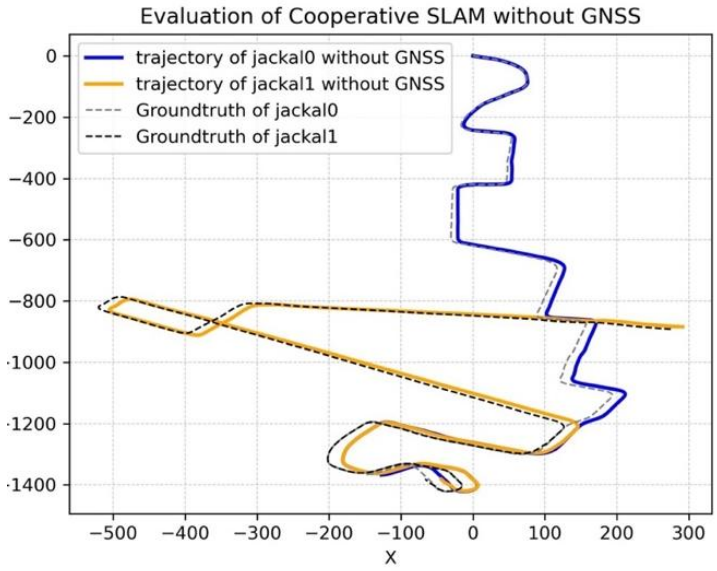
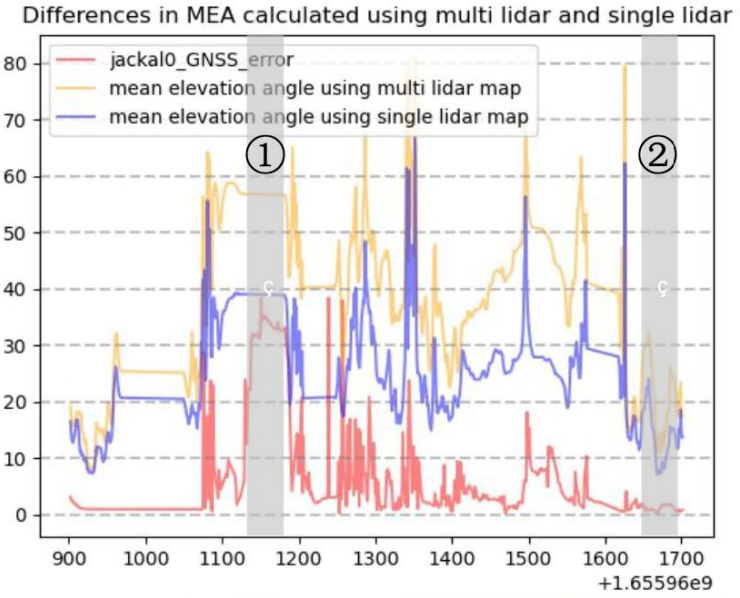



Figure 3.6 Cooperative SLAM without selected GNSS.

Table 3.2 Absolute pose error w.r.t translation part.

| RMSE(m) | Max(m) | Min(m) |
|---|---|---|
| 16.28 | 22.14 | 0.26 |

To investigate the relationship between the availability of GNSS-RTK and MEA, the APE of the u-blox measurement and the MEA of the constructed local




| | |
|---|---|
| MEA = 56.85° | MEA = 11.68° |
| GNSS Error = 35.02(m) | GNSS Error = 0.25(m) |

Figure 3.7 The relationship between the availability of GNSS-RTK and MEA.

LiDAR map is plotted in Figure 3.7 (jackal0) and Figure 3.8 (jackal1). The timestamp and corresponding street map are highlighted. As shown in the figures, the performance of the GNSS-RTK is significantly degraded in urban canyons. We

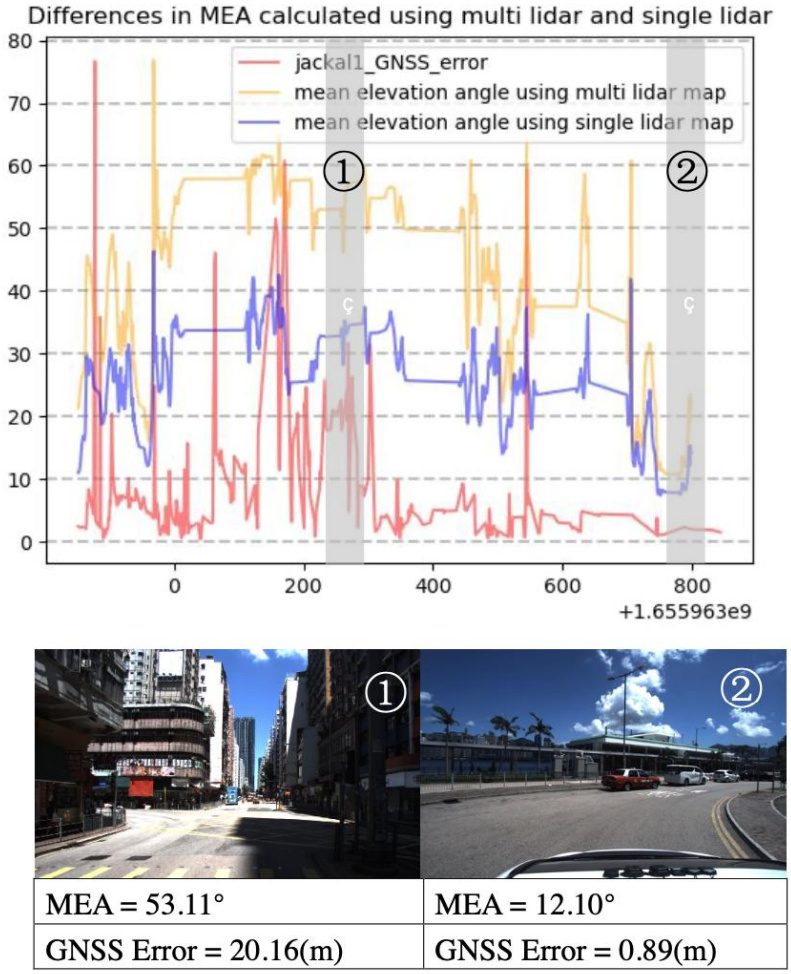



Figure 3.8 The relationship between the availability of GNSS-RTK and MEA.

consider GNSS in open areas to be exploitable and use the metric named MEA to measure the openness of a place. The yellow line represents the MEA calculated from a point cloud map constructed using multiple lidars. The purple line corresponds to the MEA derived from a point cloud map generated by a single lidar. The red line illustrates the estimation error of the GNSS values at the corresponding time. It is evident that when the MEA is relatively small, the error in GNSS measurements is minimized.

### 3.4.2 Positioning results comparison

To verify the contribution of the GNSS integrated method, the trajectories under different threshold as 15°, 25°, 35°, 45° are displayed respectively in Figure 3.9 (jackal0) and Figure 3.10 (jackal1). All RTK solutions with a MEA smaller than the threshold will be incorporated into the FGO. The APE of each pose is mapped on the trajectory refer to the color bar on the right. It is shown that the relatively small threshold will discard those GNSS with large errors and therefore make more accurate estimation.

Table 3.3 shows the APE of the trajectory estimated by different threshold. The APE of the trajectory without any selected GNSS integrated is quite large with an RMSE of 14.68m for jackal0 and 17.99 for jackal1. The trajectory under the threshold as 25° achieves the lowest error with the RMSE of 1.30 for jackal0 and 2.90 for jackal1, which means that when incorporating GNSS where an MEA is less than 25 degrees into optimization, the result is best.

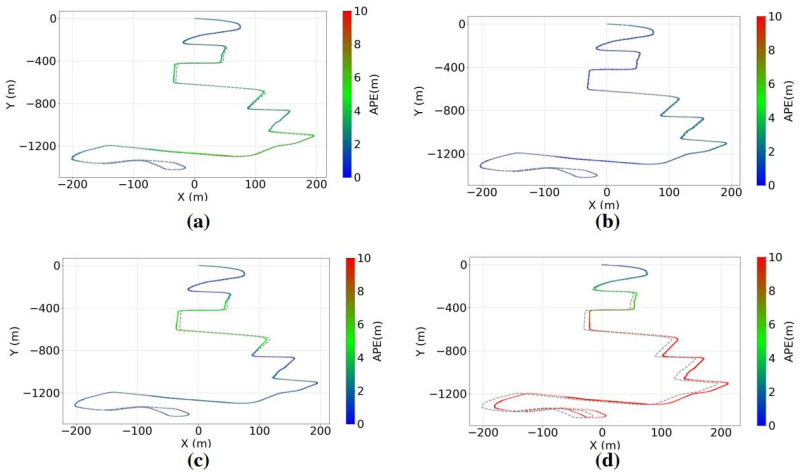



Figure 3.9 Estimated versus ground-truth pose position for jackal0 in urban-1 dataset. The APE of each estimated pose position is represented with the colored line, with corresponding colors indicating the value of APE. (a-c) Trajectories of jackal0 generated by the multi-robot SLAM framework integrated with GNSS-RTK pipelines under different threshold as 15°, 25°, 35° respectively for GNSS-RTK reliability assessment and selection on urban-1. (d) The trajectory generated by the conventional method without any GNSS-RTK factors.

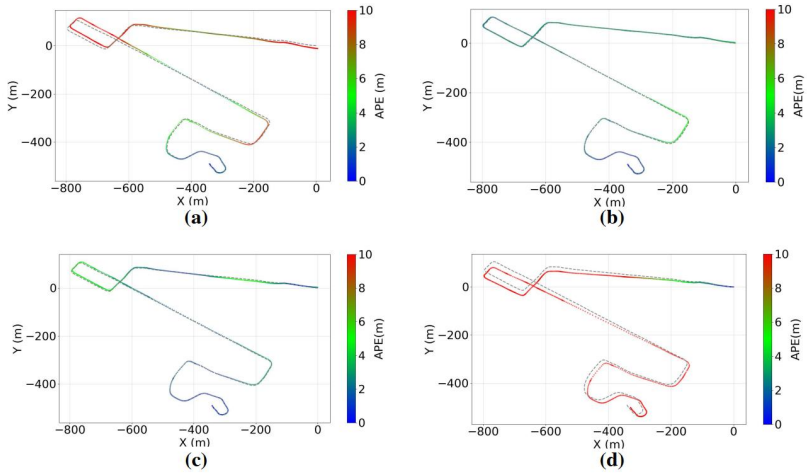



Figure 3.10 Estimated versus ground-truth pose position for jackal1 in urban-1 dataset. The APE of each estimated pose position is represented with the colored line, with corresponding colors indicating the value of APE. (a-c) Trajectories of jackal0 generated by the multi-robot SLAM framework integrated with GNSS-RTK pipelines under different threshold as 15°, 25°, 35° respectively for GNSS-RTK reliability assessment and selection on urban-1. (d) The trajectory generated by the conventional method without any GNSS-RTK factors.

| Dataset | Method | Threshold(°) | APE(m) |
|---|---|---|---|
| | | | RMSE |
| **Jackal0** | Without GNSS | - | 14.68 |
| | Selected GNSS | 15 | 3.62 |
| | | 25 | **1.30** |
| | | 35 | 1.87 |
| **Jackal1** | Without GNSS | - | 17.99 |
| | Selected GNSS | 15 | 8.63 |
| | | 25 | **2.90** |
| | | 35 | 2.95 |

After adaptively integrating GNSS with the LiDAR/IMU, the performance of the cooperative SLAM improves significantly compared to the baseline. The deviation is mostly alleviated by the selected GNSS. As shown in the Figure 3.11, the blue line represents the estimated trajectory of jackal0 and the yellow one is of jackal1. The dashed line represents the ground truth for both. The RMSE of APE is 2.68m, which is one-sixth of the original error (Table 3.4).

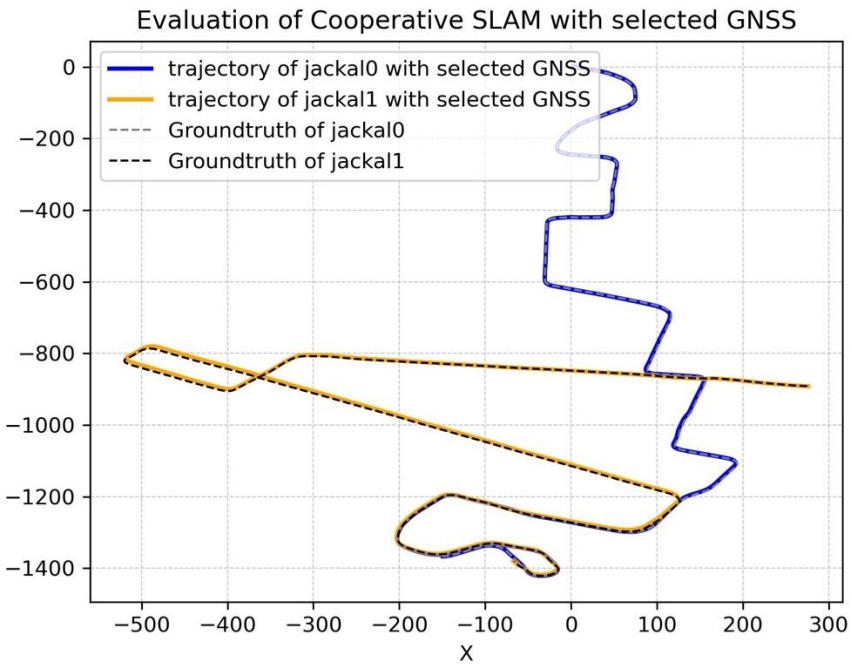



Figure 3.11 Cooperative SLAM with selected GNSS.

Table 3.4 Absolute pose error w.r.t translation part.

| RMSE(m) | Max(m) | Min(m) |
|---|---|---|
| 2.68 | 7.06 | 0.26 |

Furthermore, comparative analysis revealed that utilizing multiple LiDAR to compute MEA for selection achieve better results than using a single LiDAR, as multiple LiDAR can construct more comprehensive LiDAR maps, thereby calculating more accurate MEA to select GNSS. As shown in Figure 3.12, The horizontal axis of the vertical line corresponds to the timestamp of the filtered GNSS. It can be observed that under the threshold of 25 degrees, multi-LiDAR can

filter out more significant points of the real MEA, thereby improving accuracy. The horizontal axis of the blue vertical line on the left represents the timestamp filtered out by single-LiDAR, while the yellow on the right represents the timestamp filtered out by multi-LiDAR. The curve shows the change of APE-Error over time under three conditions: Multi-LiDAR, Single LiDAR and No GNSS. From the results, it is observed that adding GNSS constraints at the beginning and end of the trajectory effectively reduces accumulative drift. This enables trajectories to achieve relatively good performance even without loop closure constraints.

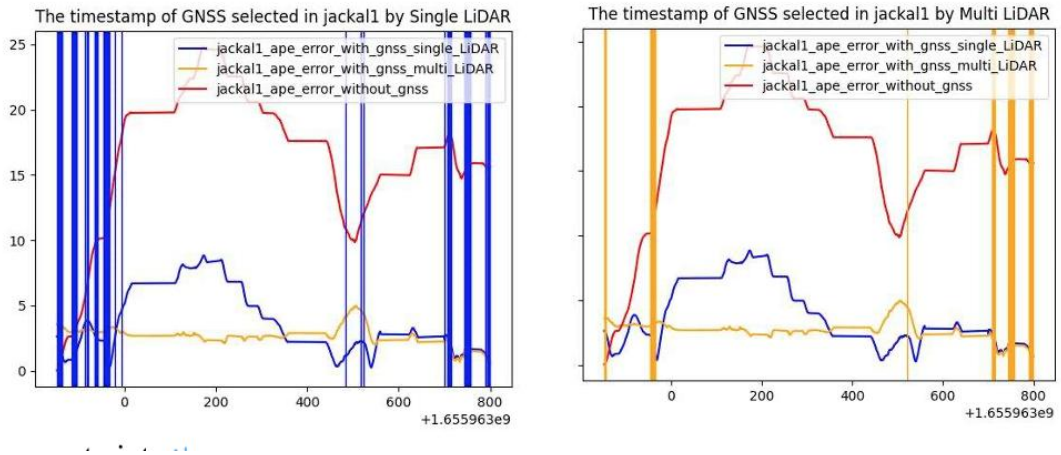



Figure 3.12 The comparison of the selected GNSS with threshold as 25°. The horizontal axis of the blue vertical line on the left represents the timestamp filtered out by single-LiDAR, while the yellow on the right represents the timestamp filtered out by multi-LiDAR.

## 3.5 Experimental results in urban canyon 2

### 3.5.1 Results of GNSS-RTK availability assessment

To verify the performance of the method, another dataset in an urban sicario is tested. The result of the baseline is shown in Figure 3.13 and Table 3.5. In Figure 3.13, the blue line represents the estimated trajectory of jackal0 and the yellow one is of jackal1. The dashed line represents the ground truth for both. It is shown that the accumulative drift is obvious and the RMSE and the Maximum value of the APE is relatively large. Since neither of the two paths returned to the origin, in other words, there were no loop closure constraints, the deviation from the ground truth increased over time.

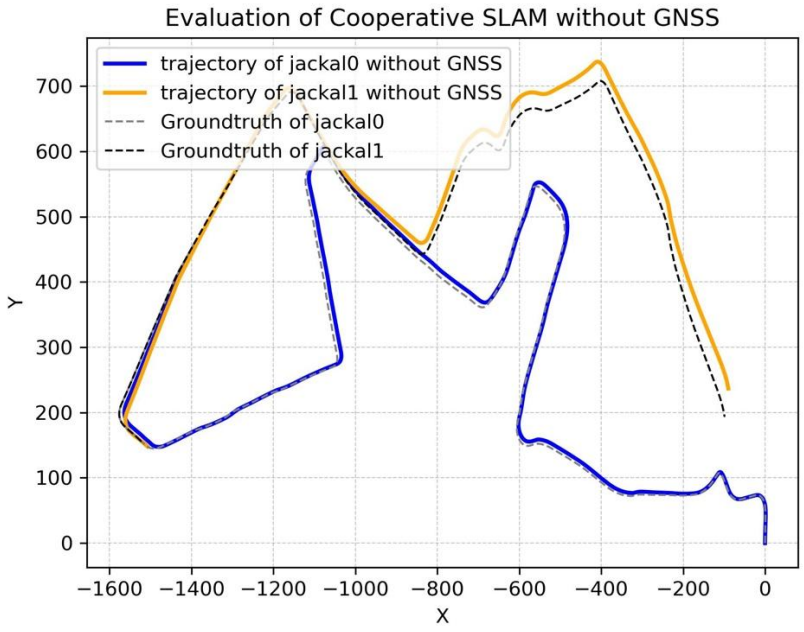



Figure 3.13 Cooperative SLAM without selected GNSS.

Table 3.5 Absolute pose error w.r.t translation part.

| RMSE(m) | Max(m) | Min(m) |
|---|---|---|
| 15.91 | 23.07 | 0.74 |

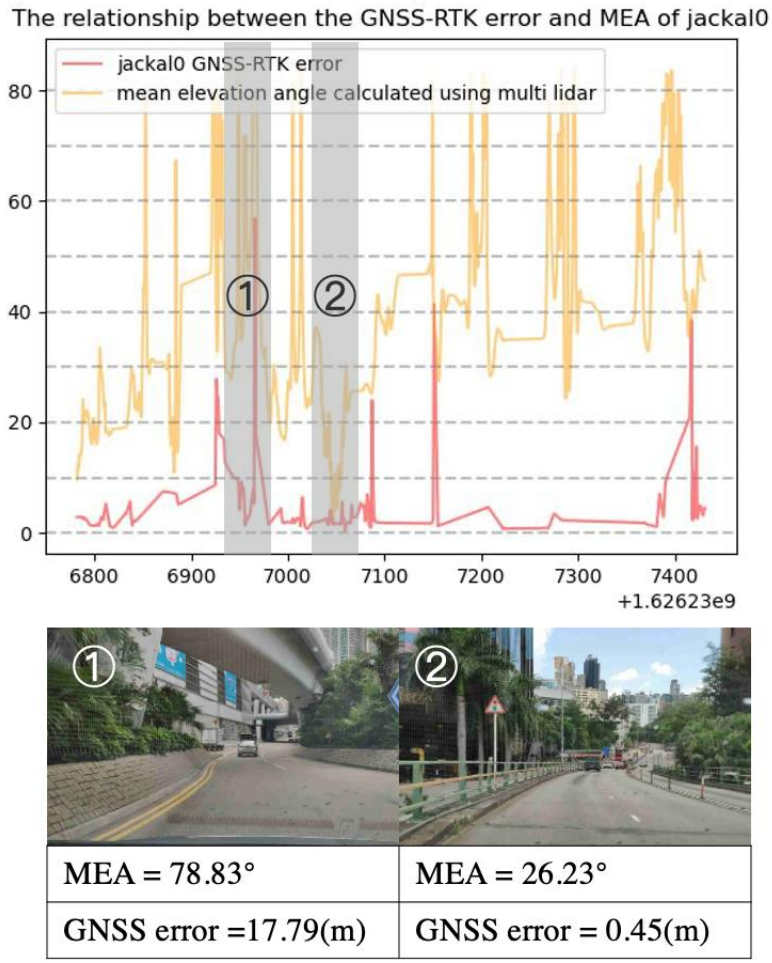



Figure 3.14 The relationship between the error of GNSS-RTK and MEA.

In Figure 3.14 (jackal0) and Figure 3.15 (jackal1), the Absolute Positional Error (APE) of the GNSS-RTK measurement, along with the corresponding MEA derived from the local topographic map, is depicted. The MEA, represented by a

yellow line, is calculated from a point cloud map that has been constructed using multiple lidars. Conversely, the red line depicts the estimation error associated with the GNSS values at the corresponding times. It has been observed that a smaller MEA tends to coincide with GNSS-RTK solutions characterized by reduced APE. This pattern aligns with trends observed in previous evaluations. Points ① and ② serve as representative cases, with their respective streetscapes illustrated accordingly. Specifically, at location ①, structural impediments such as viaducts above the vehicle or skyscrapers along the vehicle lead to inferior GNSS solutions with increased APE. In contrast, location ②, features a relatively open environment with minimal overhead obstructions, resulting in more accurate GNSS solutions with diminished APE. As shown in the figures, the performance of the GNSS-RTK is significantly degraded in urban canyons. Although there is not a strict positive correlation between GNSS error and MEA, GNSS error is relatively accurate when MEA is small. Hence, we consider GNSS in open areas to be exploitable and use the metric named MEA to measure the openness of a place. Unavailable solutions with large error which will destroy the optimization can be excluded by filtering those places with large MEA.

We acknowledge that certain environmental factors can lead to higher MEA values even in relatively open areas. For instance, the presence of utility poles, trees, and other vertical structures can artificially inflate the MEA, despite the

environment being relatively open and the GNSS-RTK solutions having a low APE. This discrepancy is evident in Figure 3.14, where MEAs up to 81.38° are observed. In these cases, the GNSS-RTK solutions still provide accurate positioning data because the obstructions do not significantly impact the signal quality. To address this issue and further enhance the accuracy of our MEA-based filtering, future research will focus on segmenting point clouds using deep learning techniques. This approach will allow us to filter out environmental features such as trees and poles, which can skew the MEA calculations. By more accurately representing the openness of the environment, we can better identify GNSS-RTK solutions with truly low APE, thereby improving the overall accuracy and reliability of our SLAM system.

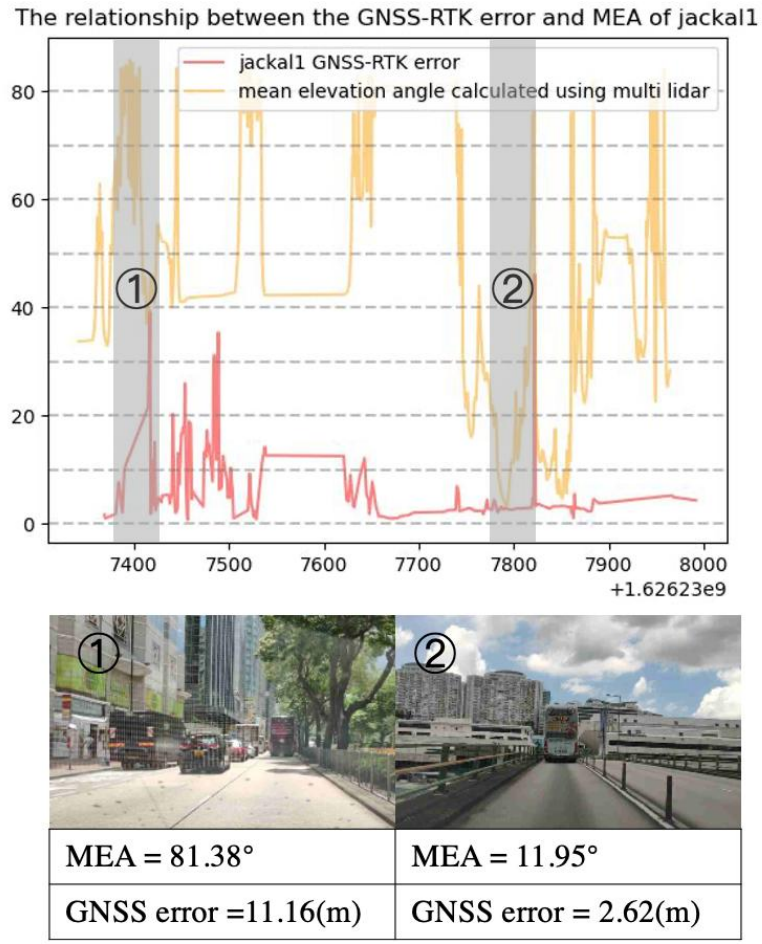



Figure 3.15 The relationship between the error of GNSS-RTK and MEA.

### 3.5.2 Positioning results comparison

The effectiveness of the proposed method for cooperative mapping is further validated on this dataset. The trajectories generated by the adaptively integrated method under different threshold as 15°, 25°, 35° and without GNSS are displayed respectively in Figure 3.16 (jackal0) and Figure 3.17 (jackal1). As the threshold decreasing, the percentage of GNSS-RTK solutions to integrate with LiDAR and IMU measurements decrease. When the threshold is relatively small,

the solutions used to correct for bias are limited hence the estimated trajectory is not optimal. When the threshold is higher, those GNSS-RTK with larger error will have a higher probability of being included due to the larger MEA, which may lead to worse result. As shown in the Table 3.6, our experiments are also consistent with this law. The APE of the trajectory estimated by different threshold is displayed. The APE of the trajectory without any selected GNSS integrated is quite large with the RMSE of 8.92m for jackal0 and 18.01 for jackal1. The trajectory under the threshold as 25° achieves the lowest error with the RMSE of 4.23m for jackal0 and 3.14m for jackal1, which means that when incorporating GNSS where an MEA is less than 25 degrees into optimization, the result is best.

The MEA is used as a criterion to filter GNSS-RTK solutions. This is because areas with higher MEA values typically have more significant obstructions, such as tall buildings or dense foliage, which degrade the quality of GNSS-RTK solutions. In such environments, multipath effects and signal blockages are more prevalent, leading to larger errors in the GNSS-RTK data. When the MEA threshold is set too low (e.g., below 25°), the filtering process rejects a large number of GNSS-RTK solutions. This results in insufficient data to correct the drift in the SLAM system, leading to higher pose estimation errors due to the lack of adequate constraints. At a moderate threshold (e.g., 25°), there is a balance between the quantity and quality of GNSS-RTK solutions. The selected GNSS-

RTK data provide enough constraints to effectively correct the SLAM system's drift, leading to optimal pose estimation accuracy. This is because, at this threshold, the included GNSS-RTK solutions are likely from relatively open areas with fewer obstructions, ensuring higher accuracy. As the threshold increases beyond the optimal point (e.g., 45° or higher), the likelihood of including GNSS-RTK solutions from obstructed areas increases. These solutions are more prone to significant errors due to multipath effects and signal blockages. Consequently, the inclusion of these erroneous solutions compromises the pose estimation accuracy, leading to a higher APE.

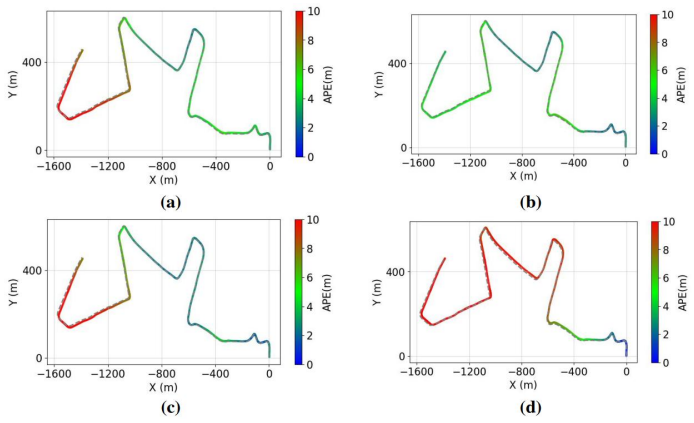



Figure 3.16 Estimated versus ground-truth pose position for jackal0 in urban-2 dataset. The APE of each estimated pose position is represented with the colored line, with corresponding colors indicating the value of APE. (a-c) Trajectories of jackal0 generated by the multi-robot SLAM framework integrated with GNSS-RTK pipelines under different threshold as 15°, 25°, 35° respectively for GNSS-RTK reliability assessment and selection on urban-1. (d) The trajectory generated by the conventional method without any GNSS-RTK factors.

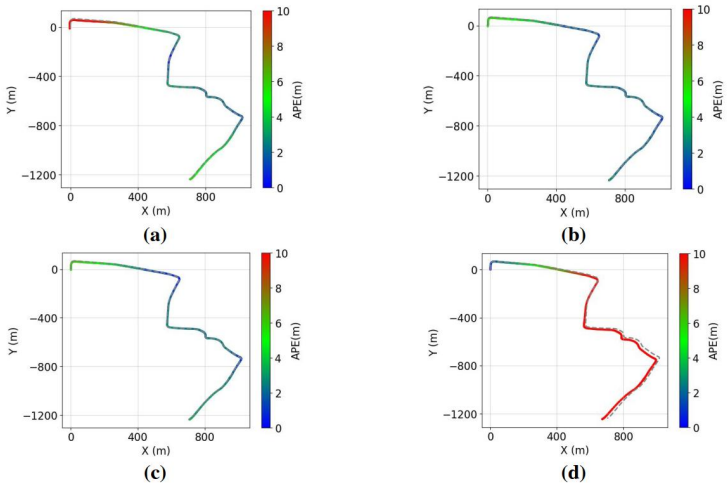



Figure 3.17 Estimated versus ground-truth pose position for jackal1 in urban-2 dataset. The APE of each estimated pose position is represented with the colored line, with corresponding colors indicating the value of APE. (a-c) Trajectories of jackal0 generated by the multi-robot SLAM framework integrated with GNSS-RTK pipelines under different threshold as 15°, 25°, 35° respectively for GNSS-RTK reliability assessment and selection on urban-1. (d) The trajectory generated by the conventional method without any GNSS-RTK factors.

Table 3.6 APE of the trajectories by different threshold and baseline.

| **Dataset** | **Method** | **Threshold(°)** | **APE(m)** |
|---|---|---|---|
| | | | **RMSE** |
| **Jackal0** | Without GNSS | - | 8.92 |
| | Selected GNSS | 15 | 6.59 |
| | | 25 | **4.23** |
| | | 35 | 6.28 |
| **Jackal1** | Without GNSS | - | 18.01 |
| | Selected GNSS | 15 | 6.03 |

| | | |
|---|---|---|
| | 25 | **3.14** |
| | 35 | 3.29 |

After adaptively integrating GNSS with the LiDAR/IMU, the performance of the cooperative SLAM improves significantly compared to the baseline. The deviation is mostly alleviated by the selected GNSS. As shown in the Figure 3.18, the blue line represents the estimated trajectory of jackal0 and the yellow one is of jackal1. The dashed line represents the ground truth for both and the RMSE of APE is 3.76m.

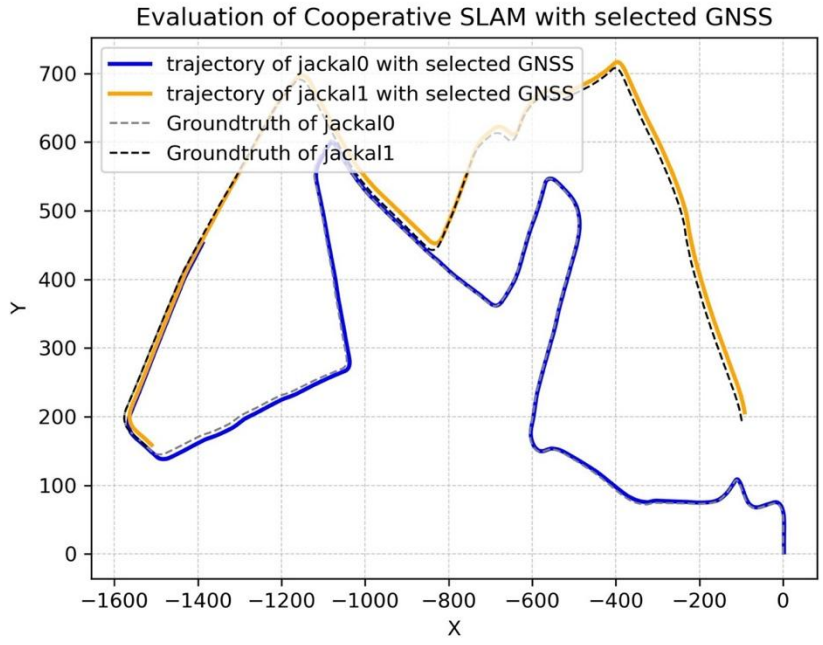



Figure 3.18 Cooperative SLAM with selected GNSS.

Table 3.7 Absolute pose error w.r.t translation part.

| RMSE(m) | Max(m) | Min(m) |
|---|---|---|
| 3.76 | 8.35 | 0.43 |

The Figure 3.19 shows the performance comparison between the baseline and the proposed method. The horizontal axis of the yellow line represents the timestamp of the selected GNSS-RTK solutions which are integrated with LiDAR/IMU. The red line is the APE with selected GNSS-RTK and the blue line is the APE without GNSS-RTK. It can be noticed that the error gradually increases with time when no GNSS is added. In the case of the GNSS scheme, the error decreases at the moment of adding GNSS and the cumulative error over time is curbed.

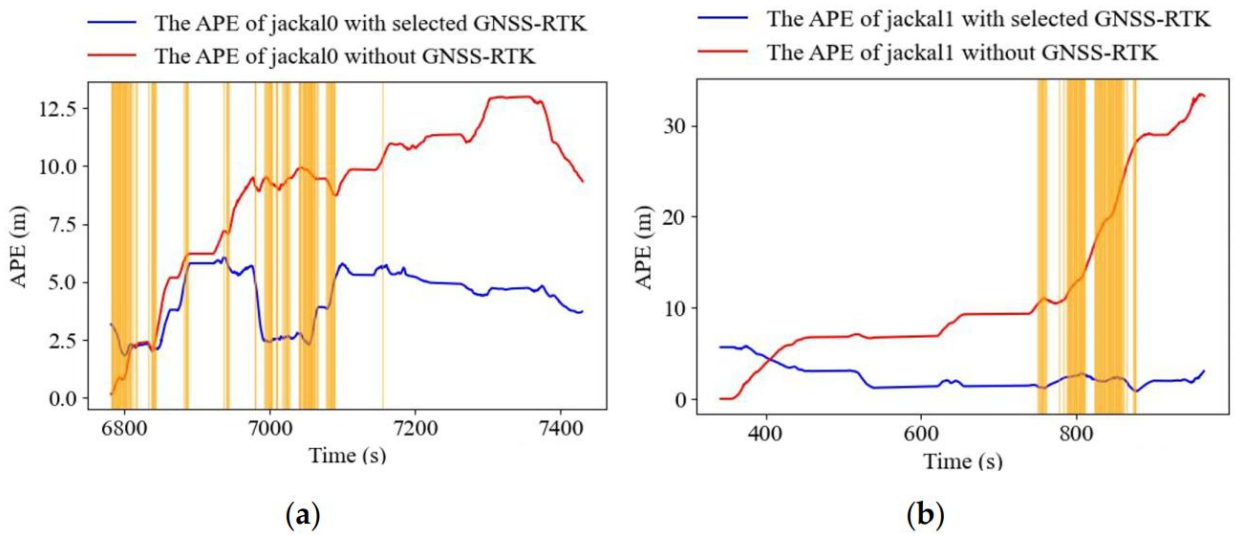



Figure 3.19 (a-b) shows a comparison of the change in APE over time for the two cases with and without the selected GNSS-RTK. The horizontal axis of the yellow line represents the timestamp of the selected GNSS-RTK solutions which are integrated with LiDAR/IMU.

## 3.6 Discussion

By selecting reliable GNSS-RTK solutions under urban scenarios for further integration with LIO, those potential solutions with large errors are excluded which is important for the accurate and convergent optimization. The selection of the reliable GNSS-RTK is based on the metric named MEA of the environment surrounding the solutions. This selection method makes use of the 3D point cloud information provided by LiDAR odometry. When this metric is large, the obstacles and buildings are tall around the GNSS-RTK solution. On the contrary, a small one demonstrates a relatively open space. By analyzing the relationship between the APE of the GNSS-RTK solutions and the MEA, the measurement is relatively reliable when the angle is small. Most cases on the two datasets supports the above law. However, since there is not strict positive relationship between the MEA and the error, some reliable solutions will be missed.

The baseline of cooperative SLAM suffers from the accumulative drift since the LiDAR and IMU are local sensors which are used to provide relative odometry. The maximum errors of both experiments are larger than 10m. after fusing with the global positioning solutions, the drift is dramatically corrected. The RMSE of the APE of the best positioning results with selected GNSS in two datasets are noticeably better than that without GNSS. As shown in the previous tables, the best precision results are improved by more than 50% compared to the baseline. By

looking at the relationship between position result and threshold, the error experiences a tendency to become smaller and then larger as threshold changes from small to large. This is because when the threshold is too small, there are very few GNSS solutions that are incorporated thus resulting in a correction that is not optimal. When the threshold is too large, the greater the likelihood that a GNSS solution with a large error will be included, resulting in a worse result. After experimentation, it was found that the optimization fails to converge when the threshold is greater than 45 degrees, i.e., the optimization fails due to the inclusion of many GNSS solutions with large deviations. In both experiments on both datasets, it was found that the best results were achieved when threshold was equal to 25 degrees. Although the lowest threshold can reduce the possibility of the inclusion of GNSS-RTK with large errors, the number of solutions also plays a great role in alleviating the drift. Also, when the threshold is lower than certain number, the error of the solution is still relatively small. However, the best threshold of different datasets maybe different because of the distribution of the error. In a word, our method still demonstrates the effectiveness of using a lower MEA threshold to filter GNSS-RTK solutions and is able to infer that different datasets will show similar trends and effects. For example, if using less dense dataset, the most precision result will take place on the threshold lower than 25 degrees.

Nevertheless, the best result achieved by the proposed method is still at meter-level due to the limited precision provided by the commercial receivers in urban scenarios.

## 3.7 Summary

This chapter propose a GNSS selection method using the metric named MEA calculated by the local point cloud map. The effectiveness of this method is verified according to the APE of the GNSS solutions. In most cases, a large MEA, where the place is under an urban dense scenario, will result in a relatively large APE of the GNSS solutions. By setting the threshold of the MEA, those GNSS solutions with large errors are more likely to be excluded. The estimation of cooperative SLAM and individual vehicle under different threshold are displayed quantitatively. In both datasets, we found the same trend, i.e., the best results in positioning were achieved when the number as well as the quality of the included GNSS were relatively balanced.

# Chapter 4 Inter-robot constraints in two-stage graph optimization

## 4.1 Introduction

In chapter 4, we investigate the local pose graph optimization phrase in two-stage optimization in cooperative SLAM. This approach aims to mitigate the cumulative drift commonly encountered in multi-robot SLAM systems and improve the overall precision of localization and mapping tasks. By converting these inter-robot constraints into virtual intra-robot constraints, the framework enhances the local pose graph optimization phase, thereby improving the accuracy of individual robot trajectories. To validate the effectiveness of this approach, the experiment section employs datasets collected from urban scenarios. The design of the experiments ensures that the driving paths of multiple robots overlap, generating the necessary separator poses for virtual observations. Performance evaluation is conducted by comparing the APE with and without inter-robot constraints under various GNSS filtering thresholds. The theoretical and experiments shows that the virtual intra-robot loop closures derived from inter-robot loop closures have varying effects on local pose graph optimization depending on scenario. The evaluation shows that the contribution of inter-robot constraints varies, with some datasets reflecting improvements and others showing negligible or negative impacts.

## 4.2 Methodology

### 4.2.1 The global optimization of the transformation

Let $\mathbf{X} = \{\mathbf{x}_0, ..., \mathbf{x}_t\}$ represent a set of 6DoF robot poses spanning from time 0 to time t, where $\mathbf{X} \subset \mathrm{SE}(3)$. The set $\mathcal{C}$ encompasses all constraints between the robot poses. For each pair of poses that define a constraint $\langle i, j \rangle \in \mathcal{C}$, we define the error $\mathbf{e}_{ij}$ as the discrepancy between the observed transformation $\mathbf{z}_{ij} \in \mathrm{SE}(3)$ and the expected transformation denoted as $\hat{\mathbf{z}}_{ij}$

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}) = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j), \tag{8}$$

$$\hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j \tag{9}$$

We specify a set of n robots, denoted as $\mathcal{N} = \{1,2, ..., n\}$. For each robot $\forall \alpha \in \mathcal{N}$, $\mathbf{X}_\alpha$ represents the poses of robot α, and $\mathbb{X} = \{\mathbf{X}_\alpha | \alpha \in \mathcal{N}\}$ constitutes a collection containing the poses of all n robots. The set $\mathbb{C} = \{\langle i, j \rangle | \mathbf{x}_i, \mathbf{x}_j \in \mathbb{X}\}$ comprises all constraints between these poses. In addressing the multi-agent SLAM problem, our objective is to solve the following equation:

$$\mathbb{X}^* = \arg\min_{\mathbb{X}} \{\mathbf{F}_{\text{intra}}(\mathbb{X}) + \mathbf{F}_{\text{inter}}(\mathbb{X})\} \tag{10}$$

The cost function is composed of two distinct components: intra-robot terms and inter-robot terms. The subset $\mathcal{C}_\alpha \subset \mathbb{C}$ encompasses the intra-robot constraints

that exist between the poses of robot α, while the subset $\mathcal{C}_{\alpha\beta} \subset \mathbb{C}$ comprises the inter-robot constraints between the poses of robots α and β.

$$\mathbf{F}_{\text{intra}}(\mathbb{X}) = \sum_{\alpha \in \mathcal{N}} \sum_{\langle i,j \rangle \in \mathcal{C}_\alpha} \mathbf{F}_{ij} \tag{11}$$

$$\mathbf{F}_{\text{inter}}(\mathbb{X}) = \sum_{\alpha,\beta \in \mathcal{N}, \alpha \neq \beta} \sum_{\langle i,j \rangle \in \mathcal{C}_{\alpha\beta}} \mathbf{F}_{ij} \tag{12}$$

Disco-SLAM introduces a two-stage graph optimization approach, integrating both global and local optimization processes. The initial global optimization phase addresses the computation of transformations among multiple robots. For every pair of separator poses $\langle \alpha_i, \beta_j \rangle$ within the set $\mathcal{C}_{\alpha\beta}$, where $\mathbf{x}_{\alpha_i} \in \mathbf{X}_\alpha$ and $\mathbf{x}_{\beta_j} \in \mathbf{X}_\beta$, consider ${}_\alpha\mathbf{X}_{\alpha_i}$ as the pose of robot α at time i in its local coordinate system. Concurrently, ${}_\beta\mathbf{X}_{\alpha_i}$ represents the pose $\mathbf{X}_{\alpha_i}$ in robot $\beta$'s local coordinates. The existence of a transformation $\mathbf{T}_{\beta\alpha,}$, which converts ${}_\alpha\mathbf{X}_{\alpha_i}$ to ${}_\beta\mathbf{X}_{\alpha_i}$, facilitates the following analysis:

$${}_\beta\mathbf{x}_{\alpha_i} = \mathbf{T}_{\beta\alpha} \cdot {}_\alpha\mathbf{x}_{\alpha_i} = \mathbf{z}_{\beta_j\alpha_i} \cdot \mathbf{x}_{\beta_j} \tag{13}$$

$$\mathbf{T}_{\beta\alpha} = \mathbf{z}_{\beta_j\alpha_i} \cdot \mathbf{x}_{\beta_j} \cdot \left( {}_\alpha\mathbf{x}_{\alpha_i} \right)^T \tag{14}$$

Upon the identification of inter-robot loop closures between robot β and robot α, the transformation $T_{\beta\alpha}$ can be ascertained. Define $\mathbb{T}$ as the collection of transformations linking any robot's frame to a global frame:

$$\mathbb{T} = \left\{ \mathbf{T}_{g\alpha} | \forall \alpha \in \mathcal{N}, \alpha \neq g \right\} \tag{15}$$

Our objective is to minimize the cumulative transformation error across local robot frames utilizing the Levenberg-Marquardt algorithm:

$$\mathbb{T}^* = \arg\min_{\mathbb{T}} \sum_{\alpha,\beta \in \mathcal{N}} \mathbf{e}_{\beta\alpha}^T \Omega_{\beta\alpha} \mathbf{e}_{\beta\alpha} \tag{16}$$

$\mathbf{e}_{\beta\alpha}$ represents the error associated with a single transformation, and $\forall \mathbf{T}_{\beta\alpha}^{(i)} \in \mathbb{T}_{\beta\alpha}$

$$\mathbf{e}_{\beta\alpha}\left(\mathbf{T}_{g\beta}, \mathbf{T}_{g\alpha}\right) = \mathbf{T}_{\beta\alpha}^{(i)} - \widehat{\mathbf{T}}_{\beta\alpha}^{(i)}\left(\mathbf{T}_{g\beta}, \mathbf{T}_{g\alpha}\right) \tag{17}$$

Subsequently, all inter-robot constraints are converted into the local coordinate frames of the respective robots, followed by the execution of the local graph optimization step. To facilitate the local optimization for robot α, it is necessary to transform the separator poses of robot β into its local coordinates.

### 4.2.2 local pose graph optimization

Following global optimization, separator poses from other robots are transformed into the local coordinate frame utilizing the most recent coordinate transformation matrices. Assuming the presence of inter-robot constraints between robot α and robot β, it is imperative to conduct local optimization for robot α. This requires the transformation of the separator poses of other robots into the local coordinate system of local robot.

$$ {}_{\alpha}\mathbf{x}_{\beta_j} = \widehat{\mathbf{T}}_{g\alpha}^{T} \cdot \widehat{\mathbf{T}}_{g\beta} \cdot \mathbf{x}_{\beta_j} \tag{18} $$

$$ {}_{\alpha}\mathbf{x}_{\gamma_l} = \widehat{\mathbf{T}}_{g\alpha}^{T} \cdot \widehat{\mathbf{T}}_{g\gamma} \cdot \mathbf{x}_{\gamma_l} \tag{19} $$

Subsequently, a Euclidean distance-based radius search is executed to identify the nearest inter-robot constraint. During the radius search, separator poses that exhibit timestamps closely proximate to the current timestamp are excluded to prevent the optimization of an ill-posed graph. For the local optimization for robot α, the virtual intra-robot loop closure transferred from inter-robot constraints will also be used. For any two distinct sets of separator poses $\langle \alpha_i, \beta_j \rangle, \langle \alpha_k, \gamma_l \rangle$, a virtual intra-robot loop closure ${}_{\alpha}\mathbf{z}_{\beta_j\gamma_l}$ can be computed by following equation:

$$ {}_{\alpha}\mathbf{z}_{\beta_j\gamma_l} = ({}_{\alpha}\mathbf{x}_{\beta_j})^{T} \cdot {}_{\alpha}\mathbf{x}_{\gamma_l} = (\widehat{\mathbf{T}}_{g\alpha}^{T} \cdot \widehat{\mathbf{T}}_{g\beta} \cdot \mathbf{x}_{\beta_j})^{T} \cdot (\widehat{\mathbf{T}}_{g\alpha}^{T} \cdot \widehat{\mathbf{T}}_{g\gamma} \cdot \mathbf{x}_{\gamma_l}) \tag{20} $$

$$ {}_{\alpha}\mathbf{z}_{\alpha_i\alpha_k}(\mathbf{x}_{\alpha_i}, \mathbf{x}_{\alpha_k}) = {}_{\alpha}\mathbf{z}_{\alpha_i\beta_j} \cdot {}_{\alpha}\mathbf{z}_{\beta_j\gamma_l} \cdot ({}_{\alpha}\mathbf{z}_{\alpha_k\gamma_l})^{T} \tag{21} $$

Ultimately, the virtual observations ${}_{\alpha}\mathbf{z}_{\alpha_i\alpha_k}$ are incorporated into the local pose graph of robot α, which is then subject to optimization.

## 4.3 Experiment

### 4.3.1 Datasets and settings

The two-vehicle dataset under urban scenarios in chapter 3 with 10Hz 32-line LiDAR and 200Hz raw inertial measurement unit(IMU) data are employed to investigate the effectiveness of inter-robot constraints in local pose graph

optimization. When designing the driving path of two vehicles, we let two vehicles start from different points and have some overlap. Each trajectories includes separator poses to generate virtual observation. There are no intra-robot constraints within each robot to investigate the inter-robot constraints related to the local robot in optimization. In Figure 4.1, the yellow line represents the ground-truth trajectories of jackal0 and the blue one for the jackal1. The EVO package (Grupp, 2017) is used for the evaluation and comparison of odometry and SLAM.

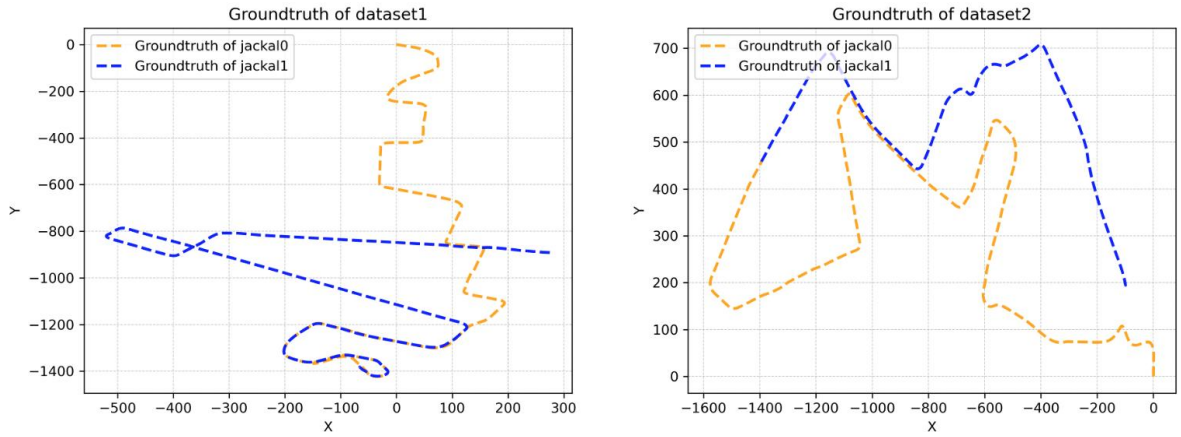



Figure 4.1 Route design of jackal0 and jackal1in two datasets.

Table 4.1 The information of the two datasets used for verification in this chapter.

| Dataset | Length(km) | | Urbanization |
|---|---|---|---|
| | Jackal0 | Jackal1 | |
| Urban-1 | 2.55 | 2.50 | Deep |
| Urban-2 | 3.1 | 2.3 | Medium |

To evaluate the effectiveness of inter-robot constraints in local pose graph optimization, we conduct two experiments. We use the two-vehicle datasets to run the multi-robots SLAM framework named DiSCo-SLAM and save their optimized local pose. In this framework the inter-robots' constraints are converted to virtual intra-robots' measurements to optimize the local pose estimation. Following that, we divided the two-vehicle dataset into two single-vehicle datasets and still run the DiSCo-SLAM for each vehicle. Using single-vehicle data will result in no inter loop constraints in local pose graph optimization. We perform comparisons of pose error between single-robot SLAM and multi-robot SLAM to explore whether the cooperative framework can noticeably improve robot localization performance and the influence of inter-robot constraints in local pose graph. We use the APE w.r.t translation part to evaluate the overall consistency of trajectory.

### 4.3.2 Performance evaluation

To explore the effect of inter-robot constraints in local pose graphs, we tested the comparison of APE with and without inter-robot constraints at the GNSS optimal filtering threshold with 25 degrees for both datasets. Table 4.2 shows the effectiveness of inter-robot constraints in the local pose graph. In general, the performance of both situation is numerically equivalent and very close, and it shows that the contribution of inter-robot constraints is uncertain. For example,

with the inter-robot constraint, the Urban-1 dataset jackal0 is better, but jackal1 is worse. Urban-2 dataset also reflects this pattern.

During the local phase of collaborative SLAM, refinement of the local graph occurs, encompassing: (1) local odometry, (2) intra-robot constraints, and (3) inter-robot constraints pertinent to the specific robot. Conversely, in single SLAM systems, the constraints for local pose graph optimization are confined to the first two categories. The influence of inter-robot constraints on the outcomes can be assessed utilizing the method of control variables. Comparative analysis between collaborative and single SLAM systems (refer to Table 4.2) indicates that the translation of inter-robot constraints into virtual intra-robot constraints marginally enhances the performance of local SLAM system.

However, according to the previous researchers' experiments results, it is said that inter-robot constraints can significantly reduce the drift of local pose estimation when there is sufficient overlap among robots. However, we did not find such a pattern in our experiments.

Table 4.2 the comparison of apes with and without inter-robot constraints.

| | APE(m) | | | |
|---|---|---|---|---|
| | Jackal0 | | Jackal1 | |
| | With | Without | With | Without |
| Urban-1 | **1.21** | 1.46 | 2.95 | **2.59** |

| | | | | |
|---|---|---|---|---|
| Urban-2 | **4.24** | 4.53 | 3.42 | **3.21** |

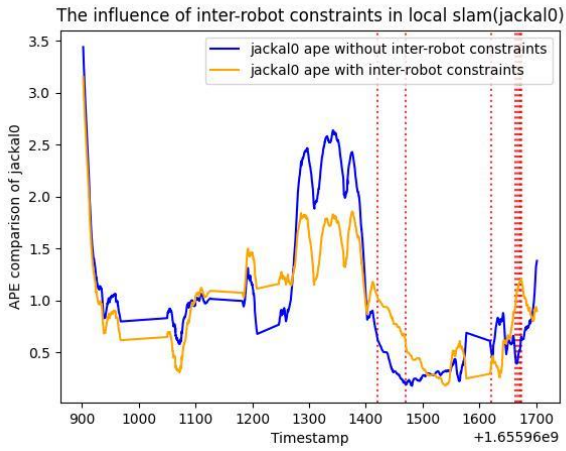



Figure 4.2 The influence of inter-robot constraints in local slam(Urban-1,jackal0). The x-coordinate of the red vertical line represents the timestamp when the inter-robot constraints were added.

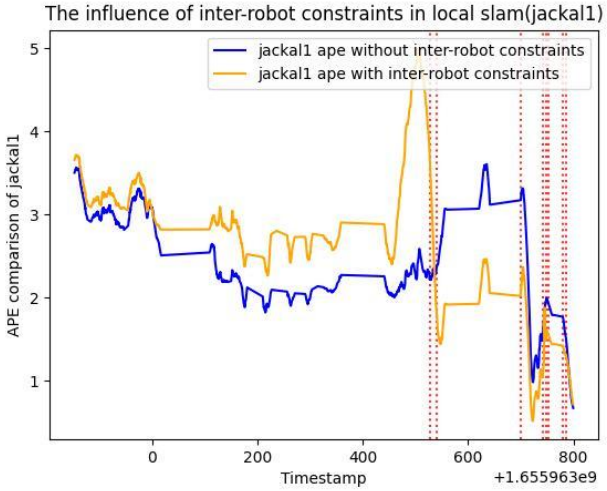



Figure 4.3 The influence of inter-robot constraints in local slam(Urban-1,jackal1). The x-coordinate of the red vertical line represents the timestamp when the inter-robot constraints were added.

The comparison of APE of Urban-1 dataset under two situations are plotted in Figure 4.2 and Figure 4.3. The distance between the horizontal coordinates of the red vertical lines is the range where virtual intra-robot constraints imposed. Unfortunately, it can be concluded from Figure 4.2 that the virtual intra-robot constraints didn't exert the great impact on the range where the constraints impose, and the overall error is very close in both cases. On the contrary, the virtual intra-robot constraints contribute on the decrease of APE on the range in jackal1 from Figure 4.3. However, the constraints make the errors of nearby poses bounce around a lot, which leads to worse estimates overall.

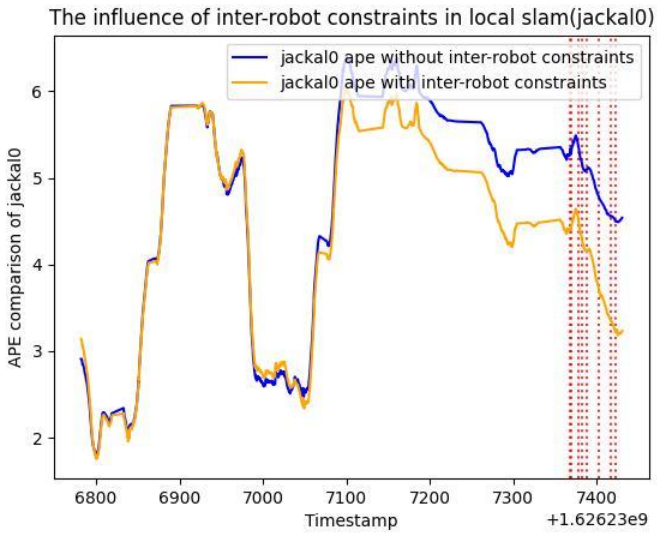



Figure 4.4 The influence of inter-robot constraints in local slam(Urban-2,jackal0). The x-coordinate of the red vertical line represents the timestamp when the inter-robot constraints were added.

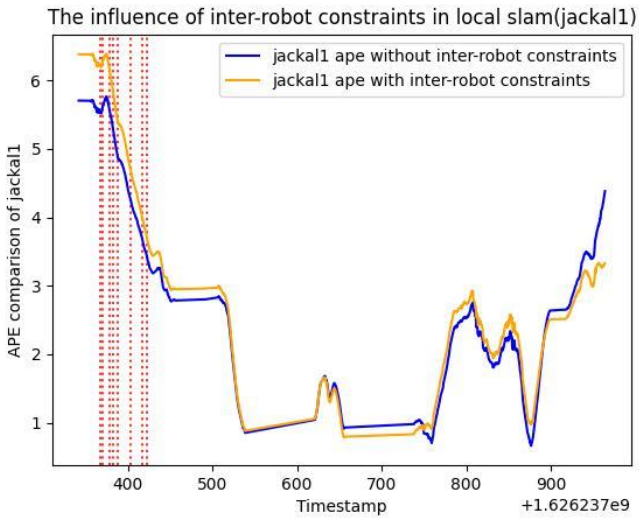



Figure 4.5 The influence of inter-robot constraints in local slam(Urban-2,jackal1). The x-coordinate of the red vertical line represents the timestamp when the inter-robot constraints were added.

The comparison of APE of Urban-2 dataset under two situations are plotted in Figure 4.4 and Figure 4.5. The distance between the horizontal coordinates of the red vertical lines is the range where virtual intra-robot constraints imposed. Constraints resulted in a significant decrease in the second half of the ape and did not bring about a deterioration in the surrounding performance as in urban-1, resulting in an overall better performance than the case without virtual intra-robot constraints. In contrast, the range in jackal1 where constraints were imposed did not improve with restrictions, and the performance of the two with and without constraints was very similar.

| Inter-robot loop closure | |
|---|---|
| jackal0 (index of keyframe) | jackal1 (index of keyframe) |
| : | : |
| 1027 | 926 |
| : | : |
| 1109 | 945 |
| : | : |
| 1253 | 993 |
| : | : |
| 1334 | 1092 |
| : | : |
| 1340 | 1103 |
| : | : |
| 1345 | 1115 |
| : | : |
| 1348 | 1128 |

(a)

| Virtual intra-robot loop closure received by jackal0 | |
|---|---|
| Index from | Index to |
| 1027 | 1109 |
| 1027 | 1253 |
| 1027 | 1334 |
| 1027 | 1340 |
| 1027 | 1345 |
| 1027 | 1348 |

(b)

| Virtual intra-robot loop closure received by jackal0 | |
|---|---|
| Index from | Index to |
| 926 | 945 |
| 926 | 993 |
| 926 | 1092 |
| 926 | 1103 |
| 926 | 1115 |
| 926 | 1128 |

(c)

Figure 4.6 The conversion of inter-robot constraints to intra-robot constraints.

| Inter-robot loop closure | |
|---|---|
| jackal0 (index of keyframe) | jackal1 (index of keyframe) |
| : | : |
| 1192 | 48 |
| : | : |
| 1197 | 61 |
| : | : |
| 1219 | 96 |
| : | : |
| 1230 | 118 |
| : | : |
| 1241 | 146 |
| : | : |
| 1263 | 202 |
| : | : |
| 1284 | 246 |

(a)

| Virtual intra-robot loop closure received by jackal0 | |
|---|---|
| Index from | Index to |
| 1192 | 1197 |
| 1192 | 1219 |
| 1192 | 1230 |
| 1192 | 1241 |
| 1192 | 1263 |
| 1192 | 1284 |

(b)

| Virtual intra-robot loop closure received by jackal0 | |
|---|---|
| Index from | Index to |
| 48 | 61 |
| 48 | 96 |
| 48 | 118 |
| 48 | 146 |
| 48 | 202 |
| 48 | 246 |

(c)

Figure 4.7 The conversion of inter-robot constraints to intra-robot constraints.

Figure 4.6 (Urban-1) and Figure 4.7 (Urban-2) illustrates the correlation between the queue of inter-robot loop closures and the virtual intra-robot loop closures received by individual robots in this experiment. The establishment of a valid inter-robot loop closure queue is achieved through the nearest neighbor search for descriptors, coupled with outlier rejection. Subsequent calculations involve determining the pose transformation from the keyframe of the vehicle in the initially adopted loop to the keyframe of the same vehicle in the subsequent loop at a specified interval, executed sequentially. To prevent close proximity of timestamps between two keyframes, certain loops are excluded. In the computation of virtual intra-robot constraints, separator poses from other robots are converted to the local coordinate frame using the most recent coordinate transformation matrices. A detailed explanation of this transfer process elucidates why inter-robot constraints are not significantly impactful for local SLAM in this study. This is attributed to the dependency of the transformation process on the precision of multiple variables and the observed overlapping of the ranges of keyframes.

## 4.4 Discussion

From the experiments above, it can be found that the contribution of inter-robot constraints to the local pose graph varies in different situations. The conclusion is aligned with the theoretical derivation. For any two distinct sets of separator poses

$\langle \alpha_i, \beta_j \rangle, \langle \alpha_k, \gamma_l \rangle$ , a virtual intra-robot loop closure ${}_{\alpha}\mathbf{z}_{\beta_j\gamma_l}$ can be computed by following equation:

$$ {}_{\alpha}\mathbf{z}_{\beta_j\gamma_l} = ({}_{\alpha}\mathbf{x}_{\beta_j})^T \cdot {}_{\alpha}\mathbf{x}_{\gamma_l} = (\hat{\mathbf{T}}_{g\alpha}^T \cdot \hat{\mathbf{T}}_{g\beta} \cdot \mathbf{x}_{\beta_j})^T \cdot (\hat{\mathbf{T}}_{g\alpha}^T \cdot \hat{\mathbf{T}}_{g\gamma} \cdot \mathbf{x}_{\gamma_l}) \tag{22} $$

$$ {}_{\alpha}\mathbf{z}_{\alpha_i\alpha_k}(\mathbf{x}_{\alpha_i}, \mathbf{x}_{\alpha_k}) = {}_{\alpha}\mathbf{z}_{\alpha_i\beta_j} \cdot {}_{\alpha}\mathbf{z}_{\beta_j\gamma_l} \cdot ({}_{\alpha}\mathbf{z}_{\alpha_k\gamma_l})^T \tag{23} $$

$$ = {}_{\alpha}\mathbf{z}_{\alpha_i\beta_j} (\hat{\mathbf{T}}_{g\alpha}^T \cdot \hat{\mathbf{T}}_{g\beta} \cdot \mathbf{x}_{\beta_j})^T \cdot (\hat{\mathbf{T}}_{g\alpha}^T \cdot \hat{\mathbf{T}}_{g\gamma} \cdot \mathbf{x}_{\gamma_l}) ({}_{\alpha}\mathbf{z}_{\alpha_k\gamma_l})^T \tag{24} $$

Since there is an error in the calculation of coordinate transformation $\hat{\mathbf{T}}_{g\beta}$ and $\hat{\mathbf{T}}_{g\gamma}$ and the error of pose $\mathbf{x}_{\beta_j}, \mathbf{x}_{\gamma_l}$ , the accuracy of the virtual intra-robot loop closure is uncertain. Since the error of transformation between the keyframes may increase as time accumulates, if the error of virtual intra-robot loop closure is lower than the lidar odometry, the performance will be improved and vice versa. Since the inter-robot constraints, which will be converted into virtual intra-robot constraints, are based on the estimation of neighboring robots' poses and the coordinate system transformation between robots, the extent to which inter-robot constraints affect the correction of local SLAM is related to various factors, such as the confidence level of the constraints and the range of keyframes imposed by the constraint. If the pose estimation of neighboring robots is inaccurate, this will directly affect the accuracy of inter-robot constraints. This may lead to map inconsistencies and localization errors. Therefore, accurate pose estimation of neighboring robots is essential for generating reliable inter-robot constraints. At the

same time, by sharing accurate constraints, multi-robot systems can correct each other's localization errors, thereby improving the overall SLAM performance of the system. Inter-robot constraints, when integrated correctly, can potentially improve the local accuracy of individual robots compared to single-robot SLAM without inter-robot constraints. However, this improvement is not guaranteed in every situation and depends on various factors. If the inter-robot constraints are accurate and reliable, they can help correct errors in individual robot's local maps, leading to better local accuracy. On the other hand, if the inter-robot constraints contain significant errors or outliers, they may introduce inconsistencies and negatively impact local SLAM accuracy. Accurate relative pose estimation between robots is critical for establishing useful inter-robot constraints. If relative pose estimation is unreliable or contains significant errors, the local SLAM accuracy may not improve or may even worsen. The effectiveness of inter-robot also depends on the robots' ability to collaborate effectively. Proper coordination, task allocation, and sharing of information among robots are essential for leveraging the benefits of inter-robot constraints. For example. In previous research experiment, the route design and cooperative mechanisms between multiple robots can make use of the more accurate routes to correct for paths with large drift.

## 4.5 Summary

This chapter utilizes two datasets to test the role of inter-robot constraints for local pose graph optimization. Through theoretical derivation and experiments, we find that the virtual inter-robot loop closures measurements transformed from inter-robot constraints have different effects on local pose graph optimization in different situations. Virtual intra-robot constraints are used to optimize the local pose graph. Robot constraints, the accuracy of coordinate transformation estimation, and the accuracy of pose estimation all affect its effectiveness on the overall estimation.

# Chapter 5 Conclusion

## 5.1 Research summary and contributions

This thesis addresses the critical challenges in enhancing the precision and efficiency of multi-robot SLAM systems, particularly within urban environments. Through an extensive study that integrates GNSS-RTK with LIO, we propose novel methodologies to improve the robustness of SLAM frameworks against common issues such as cumulative drift and the inaccuracies inherent in urban canyon scenarios.

In chapter 3, we introduce a method for GNSS-RTK selection by utilizing the 3D point cloud map and consequently fuse reliable GNSS solutions with LiDAR/IMU measurements to accomplish the cooperative mapping and localization. The relationship of the reliability of the GNSS-RTK measurements and the MEA of the extracted 3D map is investigated through the APE of the GNSS-RTK solutions. Typically, a large MEA indicates a sheltered environment, rendering the GNSS-RTK solutions unreliable and increasing the APE of the measurements. By setting the threshold of the MEA, those GNSS solutions with large errors are more likely to be exclude and different percentages of solutions are used. The estimation of cooperative SLAM and individual vehicle under different

threshold are displayed quantitatively. In summary, the performance of cooperative SLAM improves significantly by incorporating filtered GNSS-RTK solutions.

In chapter 4, we investigate the local pose graph optimization phrase in two-stage optimization in cooperative SLAM. The theoretical and experiments shows that the virtual intra-robot loop closures derived from inter-robot loop closures have varying effects on local pose graph optimization depending on scenario. The evaluation shows that the contribution of inter-robot constraints varies, with some datasets reflecting improvements and others showing negligible or negative impacts. The accuracy of coordinate transformations and pose estimations are pivotal in determining the effectiveness of inter-robot constraints.

### 5.1.1 Research scope and previous gaps

(1) **Limited GNSS-RTK Integration:** Many existing SLAM frameworks do not effectively integrate GNSS-RTK data with LiDAR and inertial measurements, leading to significant positional drift over time.

(2) **Reliability Assessment:** There is a lack of reliable methods to assess the quality of GNSS-RTK signals in urban environments, where multipath effects and signal blockages are common.

(3) **Inter-Robot Constraints:** Prior work has not fully explored the potential benefits of incorporating inter-robot constraints into local pose graph

optimizations, which can enhance the accuracy and consistency of multi-robot systems.

### 5.1.2 Contributions of this thesis

(1) **Adaptive GNSS-RTK Integration:** We develop a novel method to assess the reliability of GNSS-RTK data using the MEA derived from LiDAR point clouds. This metric helps filter out unreliable GNSS measurements, which are common in urban canyons. By integrating selected GNSS-RTK data with LiDAR and inertial measurements, our approach significantly reduces positional drift and enhances the accuracy of SLAM systems in urban environments.

(2) **Inter-robot loop closure constraints in local pose optimization :** Extensive experiments using datasets collected in urban scenarios investigate the effectiveness of virtual intra-robot constraints derived from inter-robot loop closures in local pose graph optimization. The evaluation shows that the contribution of inter-robot constraints varies in the GNSS integrated multi-robot SLAM, and the performance is related to the estimation of coordinate transformation, the range where the constraints impose and the accuracy of pose estimation.

## 5.2 Limitations and future work

(1) **GNSS-RTK reliability assessment:** The effectiveness of the proposed GNSS-RTK integration method relies heavily on the quality of GNSS data. In this thesis, the metric named MEA is used to filter reliable solutions. However, this metric is mainly based on the local point cloud map and is not able to accurately quantify the reliability of GNSS-RTK solution. Developing more sophisticated algorithms to access and filter GNSS data reliability using 3D LiDAR map may be needed. Machine learning approaches could be explored to predict GNSS data quality based on environmental features and historical data patterns. Furthermore, the optimal threshold to filter the solutions is not fixed due to the different distribution of error of diverse dataset. More experiments are needed to find the reasonable range of threshold to generalize the findings.

(2) **GNSS-RTK solutions refinement:** The integrated positioning results from the two urban canyons indicate that the highest achieved positioning precisions remain at the meter level. This limitation is attributed to the solutions provided by the commercial-grade GNSS receiver, which are compromised by the dense building structures. Consequently, refining GNSS solutions through measurement model correction during sensor fusion holds promise for improving accuracy.

(3) **Correction of inter-robot constrains:** The evaluation shows that the contribution of inter-robot constraints varies, with some datasets reflecting improvements and others showing negligible or negative impacts. In order to make use of the inter-robot constraints in local pose graph optimization, the correction of inter-robot constraints is needed to ensure the contribution to the optimization. In previous studies, the covariance matrices of these inter-robot constrain exhibit a linear relationship with the timestamp, as the dead reckoning error for each robot increases over time. However, after integrating GNSS-RTK in multi-robot SLAM framework, the error for each robot is no longer linearly related to time. Only by correcting the covariance matrices can we ensure that inter-robot constraints have a positive effect on optimization.

(4) **Camera based method:** For future work, I propose leveraging camera-captured environmental data to measure spatial openness and assess the reliability of GNSS-RTK. By utilizing image processing techniques, such as analyzing histogram properties, we can extract structural information about the environment. Additionally, advanced neural network algorithms can be employed to evaluate the openness of the environment from images. This approach could replace the current metric of mean elevation angle, which calculates the average elevation angle of all points. Using an average value

to measure environmental openness may not be accurate. Therefore, incorporating image-based methods could provide a more precise assessment of spatial openness, enhancing the reliability evaluation of GNSS-RTK.

(5) **Simulation for validation:** To enhance the robustness of the proposed algorithm and its MEA threshold for assessing GNSS-RTK reliability, future research will include extensive simulations of urban environments. We plan to develop a city model that simulates various levels of openness in urban spaces and streets. Additionally, the simulation will account for GNSS-RTK measurement errors resulting from blockage and multipath effects, which lead to non-line-of-sight (NLOS) conditions. By integrating these simulated errors with our MEA-based algorithm, we aim to evaluate its effectiveness in improving GNSS-RTK accuracy across different urban scenarios. This approach will help identify optimal MEA thresholds and their impact on accuracy, providing a more comprehensive validation of the algorithm. Such simulations will significantly bolster the credibility of our findings and offer valuable insights into enhancing GNSS-RTK reliability in complex urban environments.

In conclusion, this thesis provides significant advancements in the field of multi-robot SLAM, offering practical solutions to improve accuracy in urban environments. By addressing the limitations and exploring the proposed future work, further enhancements can be achieved.

## References


Birk, A., & Carpin, S. (2006). Merging occupancy grid maps from multiple robots. *Proceedings of the IEEE*, *94*(7), 1384–1397. https://doi.org/10.1109/JPROC.2006.876965

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., & Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332. https://doi.org/10.1109/TRO.2016.2624754

Cao, Y., & Beltrame, G. (2021). VIR-SLAM: visual, inertial, and ranging SLAM for single and multi-robot systems. *Autonomous Robots*, *45*(6), 905–917. https://doi.org/10.1007/s10514-021-09992-7

Chang, L., Niu, X., Liu, T., Tang, J., & Qian, C. (2019). GNSS/INS/LiDAR-SLAM integrated navigation system based on graph optimization. *Remote Sensing*, *11*(9). https://doi.org/10.3390/rs11091009

Chang, Y., Ebadi, K., Denniston, C. E., Ginting, M. F., Rosinol, A., Reinke, A., Palieri, M., Shi, J., Chatterjee, A., Morrell, B., Agha-Mohammadi, A. A., & Carlone, L. (2022). LAMP 2.0: A Robust Multi-Robot SLAM System for Operation in Challenging Large-Scale Underground Environments. *IEEE Robotics and Automation Letters*, *7*(4), 9175–9182. https://doi.org/10.1109/LRA.2022.3191204

Chiang, K. W., Tsai, G. J., Chu, H. J., & El-Sheimy, N. (2020). Performance Enhancement of INS/GNSS/Refreshed-SLAM Integration for Acceptable Lane-Level Navigation Accuracy. *IEEE Transactions on Vehicular Technology*, *69*(3), 2463–2476. https://doi.org/10.1109/TVT.2020.2966765

Denniston, C. E., Chang, Y., Reinke, A., Ebadi, K., Sukhatme, G. S., Carlone, L., Morrell, B., & Agha-Mohammadi, A. A. (2022). Loop Closure Prioritization for Efficient and Scalable Multi-Robot SLAM. *IEEE Robotics and Automation Letters*, *7*(4), 9651–9658. https://doi.org/10.1109/LRA.2022.3191156

Do, H., Hong, S., & Kim, J. (2020). Robust Loop Closure Method for Multi-Robot Map Fusion by Integration of Consistency and Data Similarity. *IEEE Robotics and Automation Letters*, *5*(4), 5701–5708. https://doi.org/10.1109/LRA.2020.3010731

Dubé, R., Gawel, A., Sommer, H., Nieto, J., Siegwart, R., & Cadena, C. (2017). *An Online Multi-Robot SLAM System for 3D LiDARs*. https://doi.org/10.0/Linux-x86_64

Dubois, R., Eudes, A., & Frémont, V. (2022). Sharing visual-inertial data for collaborative decentralized simultaneous localization and mapping. *Robotics and Autonomous Systems*, *148*. https://doi.org/10.1016/j.robot.2021.103933

Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2015). IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. *Robotics: Science and Systems*, *11*. https://doi.org/10.15607/RSS.2015.XI.006

Gao, J., Sha, J., Wang, Y., Wang, X., & Tan, C. (2024). A fast and stable GNSS-LiDAR-inertial state estimator from coarse to fine by iterated error-state Kalman filter. *Robotics and Autonomous Systems*, *175*. https://doi.org/10.1016/j.robot.2024.104675

Gao, Y., Liu, S., Atia, M. M., & Noureldin, A. (2015). INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm. *Sensors (Switzerland)*, *15*(9), 23286–23302. https://doi.org/10.3390/s150923286

'Grupp, M. (2017). *evo: Python package for the evaluation of odometry and SLAM*.

He, G., Yuan, X., Zhuang, Y., & Hu, H. (2021). An Integrated GNSS/LiDAR-SLAM Pose Estimation Framework for Large-Scale Map Building in Partially GNSS-Denied Environments. *IEEE Transactions on Instrumentation and Measurement*, *70*. https://doi.org/10.1109/TIM.2020.3024405

Hening, S., Ippolito, C., Krishnakumar, K., Stepanyan, V., & Teodorescu, M. (2017). 3D LiDAR SLAM integration with GPS/INS for UAVs in urban GPS-degraded environments. *AIAA Information Systems-AIAA Infotech at Aerospace, 2017*. https://doi.org/10.2514/6.2017-0448

Hsu, L. T. (2018). Analysis and modeling GPS NLOS effect in highly urbanized area. *GPS Solutions*, *22*(1). https://doi.org/10.1007/s10291-017-0667-9

Huang, Y., Shan, T., Chen, F., & Englot, B. (2022). DiSCo-SLAM: Distributed Scan Context-Enabled Multi-Robot LiDAR SLAM with Two-Stage Global-Local Graph Optimization. *IEEE Robotics and Automation Letters*, *7*(2), 1150–1157. https://doi.org/10.1109/LRA.2021.3138156

Khan, M. U., Zaidi, S. A. A., Ishtiaq, A., Bukhari, S. U. R., Samer, S., & Farman, A. (2021, July 15). A Comparative Survey of LiDAR-SLAM and LiDAR based Sensor Technologies. *Proceedings of the 2021 Mohammad Ali Jinnah University International Conference on Computing, MAJICC 2021*. https://doi.org/10.1109/MAJICC53071.2021.9526266

Kim, G., & Kim, A. (2018). Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4802–4809. https://doi.org/10.1109/IROS.2018.8593953

Lajoie, P. Y., & Beltrame, G. (2024). Swarm-SLAM: Sparse Decentralized Collaborative Simultaneous Localization and Mapping Framework for Multi-Robot Systems. *IEEE Robotics and Automation Letters*, *9*(1), 475–482. https://doi.org/10.1109/LRA.2023.3333742

Lajoie, P.-Y., Ramtoula, B., Wu, F., & Beltrame, G. (2021). *Towards Collaborative Simultaneous Localization and Mapping: a Survey of the Current Research Landscape*. https://doi.org/10.55417/fr.2022032

Li, T., Zhang, H., Gao, Z., Chen, Q., & Niu, X. (2018). High-accuracy positioning in urban environments using single-frequency multi-GNSS RTK/MEMSIMU integration. *Remote Sensing*, *10*(2). https://doi.org/10.3390/rs10020205

Li, X., Yu, H., Wang, X., Li, S., Zhou, Y., & Chang, H. (2023). FGO-GIL: Factor Graph Optimization-Based GNSS RTK/INS/LiDAR Tightly Coupled Integration for Precise and Continuous Navigation. *IEEE Sensors Journal*, *23*(13), 14534–14548. https://doi.org/10.1109/JSEN.2023.3278723

Liu, X., Wen, W., & Hsu, L. T. (2024). GLIO: Tightly-Coupled GNSS/LiDAR/IMU Integration for Continuous and Drift-Free State Estimation of Intelligent Vehicles in Urban Areas. *IEEE Transactions on Intelligent Vehicles*, *9*(1), 1412–1422. https://doi.org/10.1109/TIV.2023.3323648

Mangelson, J. G., Dominic, D., Eustice, R. M., & Vasudevan, R. (2018). Pairwise Consistent Measurement Set Maximization for Robust Multi-Robot Map Merging. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2916–2923. https://doi.org/10.1109/ICRA.2018.8460217

Nieto-Granda, C., Rogers, J. G., & Christensen, H. I. (2014). Coordination strategies for multi-robot exploration and mapping. *International Journal of Robotics Research*, *33*(4), 519–533. https://doi.org/10.1177/0278364913515309

Qian, C., Liu, H., Tang, J., Chen, Y., Kaartinen, H., Kukko, A., Zhu, L., Liang, X., Chen, L., & Hyyppä, J. (2017). An integrated GNSS/INS/LiDAR-SLAM

positioning method for highly accurate forest stem mapping. *Remote Sensing*, *9*(1). https://doi.org/10.3390/rs9010003

Rone, W., & Ben-Tzvi, P. (2013). Mapping, localization and motion planning in mobile multi-robotic systems. *Robotica*, *31*(1), 1–23. https://doi.org/10.1017/S0263574712000021

Saeedi, S., Trentini, M., Seto, M., & Li, H. (2016). Multiple-Robot Simultaneous Localization and Mapping: A Review. *Journal of Field Robotics*, *33*(1), 3–46. https://doi.org/10.1002/rob.21620

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., & Rus, D. (2020). *LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping*. http://arxiv.org/abs/2007.00258

Shen, N., Chen, L., Liu, J., Wang, L., Tao, T., Wu, D., & Chen, R. (2019). A review of Global Navigation Satellite System (GNSS)-based dynamic monitoring technologies for structural health monitoring. In *Remote Sensing* (Vol. 11, Issue 9). MDPI AG. https://doi.org/10.3390/rs11091001

Su, Y., Wang, T., Shao, S., Yao, C., & Wang, Z. (2021a). GR-LOAM: LiDAR-based sensor fusion SLAM for ground robots on complex terrain. *Robotics and Autonomous Systems*, *140*. https://doi.org/10.1016/j.robot.2021.103759

Su, Y., Wang, T., Shao, S., Yao, C., & Wang, Z. (2021b). GR-LOAM: LiDAR-based sensor fusion SLAM for ground robots on complex terrain. *Robotics and Autonomous Systems*, *140*. https://doi.org/10.1016/j.robot.2021.103759

Tang, J., Chen, Y., Niu, X., Wang, L., Chen, L., Liu, J., Shi, C., & Hyyppä, J. (2015). LiDAR scan matching aided inertial navigation system in GNSS-denied environments. *Sensors (Switzerland)*, *15*(7), 16710–16728. https://doi.org/10.3390/s150716710

Tian, Y., Chang, Y., Herrera Arias, F., Nieto-Granda, C., How, J. P., & Carlone, L. (2022). Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems. *IEEE Transactions on Robotics*, *38*(4), 2022–2038. https://doi.org/10.1109/TRO.2021.3137751

Transactions On Robotics, I., & September, M. (2008). *iSAM: Incremental Smoothing and Mapping* (Vol. 7). http://ieeexplore.ieee.org,

Wang, Y., Sun, Z., Xu, C.-Z., Sarma, S. E., Yang, J., & Kong, H. (2020). LiDAR Iris for Loop-Closure Detection. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5769–5775. https://doi.org/10.1109/IROS45743.2020.9341010

Wang, Z., Huang, S., & Dissanayake, G. (2007). *Multi-robot simultaneous localization and mapping using D-SLAM framework*.

Wen, W. W., Zhang, G., & Hsu, L. T. (2021). GNSS NLOS Exclusion Based on Dynamic Object Detection Using LiDAR Point Cloud. *IEEE Transactions on Intelligent Transportation Systems*, *22*(2), 853–862. https://doi.org/10.1109/TITS.2019.2961128

Wu, J., Wang, J., & Wei, C. (2022). Robust Map Merging Method for Collaborative LiDAR-based SLAM Using GPS Sensor. *Journal of Physics: Conference Series*, *2402*(1). https://doi.org/10.1088/1742-6596/2402/1/012004

Xu, X., Zhang, L., Yang, J., Cao, C., Wang, W., Ran, Y., Tan, Z., & Luo, M. (2022). A Review of Multi-Sensor Fusion SLAM Systems Based on 3D LIDAR. In *Remote Sensing* (Vol. 14, Issue 12). MDPI. https://doi.org/10.3390/rs14122835

Zangenehnejad, F., & Gao, Y. (2021). GNSS smartphones positioning: advances, challenges, opportunities, and future perspectives. In *Satellite Navigation* (Vol. 2, Issue 1). Springer. https://doi.org/10.1186/s43020-021-00054-y

Zhang, G., & Hsu, L. T. (2018). Intelligent GNSS/INS integrated navigation system for a commercial UAV flight control system. *Aerospace Science and Technology*, *80*, 368–380. https://doi.org/10.1016/j.ast.2018.07.026

Zhang, J., & Singh, S. (n.d.). *LOAM: Lidar Odometry and Mapping in Real-time*.

Zhang, J., Wen, W., Huang, F., Wang, Y., Chen, X., & Hsu, L. T. (2022). GNSS-RTK Adaptively Integrated with LiDAR/IMU Odometry for Continuously Global

Positioning in Urban Canyons. *Applied Sciences (Switzerland)*, *12*(10). https://doi.org/10.3390/app12105193

Zhang, X., Zhang, Z., Wang, Q., & Yang, Y. (2021). Using a two-stage method to reject false loop closures and improve the accuracy of collaborative slam systems. *Electronics (Switzerland)*, *10*(21). https://doi.org/10.3390/electronics10212638

Zhong, S., Qi, Y., Chen, Z., Wu, J., Chen, H., & Liu, M. (2022). *DCL-SLAM: A Distributed Collaborative LiDAR SLAM Framework for a Robotic Swarm*. http://arxiv.org/abs/2210.11978

Zhou, B., He, Y., Qian, K., Ma, X., & Li, X. (2021). S4-SLAM: A real-time 3D LIDAR SLAM system for ground/watersurface multi-scene outdoor applications. *Autonomous Robots*, *45*(1), 77–98. https://doi.org/10.1007/s10514-020-09948-3

Zhou, H., Yao, Z., Zhang, Z., Liu, P., & Lu, M. (2022). An Online Multi-Robot SLAM System Based on Lidar/UWB Fusion. *IEEE Sensors Journal*, *22*(3), 2530–2542. https://doi.org/10.1109/JSEN.2021.3136929