DYNAMIC OBJECT-AWARE LIDAR ODOMETRY IN URBAN AREAS: FROM SINGLE TO COOPERATIVE NAVIGATION

FENG HUANG

PhD

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University

Department of Aeronautical and Aviation Engineering

Dynamic Object-Aware LiDAR Odometry in Urban Areas: From Single to Cooperative Navigation

Feng Huang

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

Doctor of Philosophy

November 2024

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.

_______(Signature)

Feng Huang (Name of student)

Abstract

Robust and precise positioning is critical for the autonomous system with navigation requirements. In recent years, Light detection and ranging (LiDAR) odometry have been extensively studied to achieve this goal. Satisfactory performance of LiDAR odometry (LO) can be achieved in sub-urban areas with abundant environmental features and limited moving objects. However, the performance is significantly degraded in challenging urban canyons with numerous moving objects. Moreover, the LO is subjected to drift over time. Global Navigation Satellite Systems (GNSS) can provide reliable absolute positioning in open areas and serve as a complement to LO. However, GNSS performance is often degraded in urban areas due to signal reflections caused by surrounding structures.

In this thesis, we developed new methods to mitigate the impact of outliers in LiDAR odometry, enhancing positioning performance for autonomous driving in urban environments. First, we evaluated several popular and widely studied LO pipelines using datasets collected from urban canyons in Hong Kong, presenting the results in terms of both positioning accuracy and computational efficiency. We concluded three key factors that affect LO performance in urban canyons: ego-vehicle dynamics, moving objects, and the degree of urbanization. Second, we conducted an in-depth study on how to improve LO performance with the existence of large amounts of dynamic objects using deep learning-based techniques and point-wise discrepancy images. LO performance was further improved by applying object reweighting in highly dynamic scenarios. Third, we proposed a LiDAR-aided cycle slip detection method for GNSS-RTK, which effectively identifies cycle slips in carrier-phase measurements by leveraging consecutive relative pose estimates provided by LO.

Furthermore, we present roadside infrastructure-assisted navigation in urban areas. First, we explore the use of roadside LiDAR to provide accurate states that serve as the global constraint in the LiDAR/Inertial odometry (LIO) graph-based optimization. Second, we present the use of consistent roadside double-differenced (DD) constraints provided by roadside GNSS are jointly optimized into the factor graph optimization. Third, we introduce an error-map-aided multi-sensor integrated system that utilizes error information collected by a sensor-rich autonomous vehicle. This error data are then uploaded to the roadside infrastructure, where it is subsequently distributed to other vehicles, which benefits the navigation performance of other vehicles.

Numerous experiments are conducted using the onboard sensor platform and vehicle-infrastructure platform to validate the performance of the proposed method. The proposed dynamic object-aware LO significantly enhances positioning accuracy, achieving decimeter-level precision compared to the traditional meter-level accuracy in high-dynamic environments. With the assistance of roadside sensors, the proposed method achieved a 36.6% improvement in terms of absolute positioning accuracy compared to the state-of-the-art GNSS/LiDAR/INS integrated method. The evaluation results demonstrate that our proposed methods outperform the conventional positioning methods, providing accurate and reliable positioning and mapping for autonomous driving in urban canyons.

Acknowledgements

I extend my deepest gratitude to my supervisors, Dr. Li-Ta Hsu and Dr. Weisong Wen, for giving me the opportunity to engage in exciting research. Their encouragement and praise have sustained me through the challenges of my Ph.D. journey. This thesis would not have been possible without their brilliant ideas and exceptional dedication to research. Dr. Li-Ta Hsu and Dr. Weisong Wen are not just my Ph.D. supervisors but also my mentors in life. I am also grateful for the guidance and valuable suggestions from Prof. Danwei Wang and Dr. Yuanzhe Wang during my attachment program at Nanyang Technological University.

I would like to extend my gratitude to my examiners, Associate Professor Hui Kong from the University of Macau and Professor Shengfeng Gu from Wuhan University, for their constructive comments and insightful advice. I am deeply appreciative of the valuable time they generously provided to review and examine my thesis. Additionally, I would like to extend my heartfelt thanks to Associate Professor Wei Liu for his kind support as BoE during my Ph.D. confirmation and oral examination.

I am grateful to The Hong Kong Polytechnic University for providing the resources necessary to carry out my research. I wish to thank Dr. Guohao Zhang, Dr. Bing Xu, and Dr. Xiwei Bai for their valuable discussions and advice; Dr. Jiachen Zhang, Hoi-Fung Ng, Yihan Zhong, Xikun Liu, Donghui Shen for their assistance and cooperation in completing my thesis; Jiachong Chang, Runzhi Hu, Penggao Yan for their help and collaboration in research; Yin Chiu KAN, Samantha SIN, Dr. Yiran

Luo, Dr. Liang Zhang, Penghui Xu, Dr. Ronghe Jin, Liyuan Zhang, Xi Zheng, Xiao Xia, Dr. Jian Liu, Dr. Shiyu Bai, Dr. Shaoming Xin, Xiaoyu Luo and all the IPNL members for their support in my research; Peiyu Chen, Weipeng Guan and Dr. Peng Lu from HKU for their insights in collaborative research; colleagues from NTU for their constructive advice on my research and life during my attachment; colleagues from the Hong Kong Applied Science and Technology Research Institute for their invaluable industrial support for my research; my friends Huang Kaicheng, Zheng Yibin, Liao Shengda, Wu Di, Li Dongfang for enriching my life in Hong Kong. Furthermore, I deeply appreciate everyone who offered emotional and practical support during my studies in Hong Kong.

Last but not least, I wish to express my gratitude to my family. I extend special thanks to my loving wife, Mengting, and my child, Xinhe, for their endless love and support throughout the challenges and difficulties of my Ph.D. journey. I am also grateful for the support of my wife's family.

List of Publications

Journal

[1] F. Huang, W. Wen, J. Zhang, and L.-T. Hsu, “Point Wise or Feature Wise? A Benchmark Comparison of Publicly Available LiDAR Odometry Algorithms in Urban Canyons,” *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 155–173, 2022

[2] F. Huang, W. Wen, J. Zhang, C. Wang, and L.-T. Hsu, “Dynamic object-aware LiDAR odometry aided by joint weightings estimation in urban areas,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, pp. 3345–3359, 2024 (Project)

[3] F. Huang, W. Wen, Guohao Zhang, Dongzhe Su, Yulong Huang, and Li-Ta Hsu, Continuous Error Map Aided Adaptive Multi-Sensor Integrated Positioning for Connected Autonomous Vehicles in Urban Scenarios, *IEEE Transactions on Instrumentation and Measurement* (submitted)

[4] L.-T. Hsu, F. Huang, H.-F. Ng, G. Zhang, Y. Zhong, X. Bai, and W. Wen, “Hong Kong UrbanNav: An open-source multisensory dataset for benchmarking urban navigation algorithms,” *NAVIGATION: Journal of the Institute of Navigation*, vol. 70, no. 4, 2023 (Project)

[5] P. Chen, W. Guan, F. Huang, Y. Zhong, W. Wen, L.-T. Hsu, and P. Lu, “Ecmd: An event-centric multisensory driving dataset for slam,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 407–416, 2024 (Co-first author worked

with HKU team) (Project)

[6] J. Chang, Y. Zhang, S. Fan, F. Huang, D. Xu, and L.-T. Hsu, “An anti-spoofing model based on mvm and mccm for a loosely-coupled gnss/ins/lidar kalman filter,” *IEEE Transactions on Intelligent Vehicles*, 2023

[7] J. Chang, R. Hu, F. Huang, D. Xu, and L.-T. Hsu, “Lidar-based ndt matching performance evaluation for positioning in adverse weather conditions,” *IEEE Sensors Journal*, 2023 (Corresponding author)

[8] J. Chang, F. Huang, L. Zhang, D. Xu, and L.-T. Hsu, “Selection of areas for effective gnss spoofing attacks to a vehicle-mounted msf system based on scenario classification models,” *IEEE Transactions on Vehicular Technology*, vol. 72, no. 11, pp. 14 645–14 655, 2023

[9] Y. Zhong, F. Huang, J. Zhang, W. Wen, and L.-T. Hsu, “Low-cost solid-state LiDAR/inertial-based localization with prior map for autonomous systems in urban scenarios,” *IET Intelligent Transport Systems*, vol. 17, no. 3, pp. 474–486, 2023

[10] J. Zhang, W. Wen, F. Huang, Y. Wang, X. Chen, and L.-T. Hsu, “Gnss-rtk adaptively integrated with lidar/imu odometry for continuously global positioning in urban canyons,” *Applied Sciences*, vol. 12, no. 10, p. 5193, 2022

[11] J. Chang, L. Zhang, L.-T. Hsu, B. Xu, F. Huang, and D. Xu, “Analytic models of a loosely coupled gnss/ins/lidar kalman filter considering update frequency under a spoofing attack,” *IEEE Sensors Journal*, vol. 22, no. 23, pp. 23 341–23 355, 2022

[12] J. Zhang, W. Wen, F. Huang, X. Chen, and L.-T. Hsu, “Coarse-to-fine loosely-coupled lidar-inertial odometry for urban positioning and mapping,” *Remote Sensing*, vol. 13, no. 12, p. 2371, 2021

[13] P. Yan, Z. Li, F. Huang, W. Wen, and L.-T. Hsu, “Fault detection algorithm for gaussian mixture noises: An application in lidar/imu integrated localization

systems," *NAVIGATION: Journal of the Institute of Navigation*, vol. 72, no. 1, 2025

Conference

[1] F. Huang, H. Chen, A. Urtay, D. Su, W. Wen, and L.-T. Hsu, "Roadside infrastructure assisted lidar/inertial-based mapping for intelligent vehicles in urban areas," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 5831–5837 (Project)

[2] F. Huang, W. Wen, G. Zhang, D. Su, and L.-T. Hsu, "Adaptive multi-sensor integrated navigation system aided by continuous error map from rsu for autonomous vehicles in urban areas," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 5895–5902(Project)

[3] Huang, F., Zhong, Y., Chen, H., Su, D., Wu, J., Wen, W., and Huang, Y.. Roadside GNSS Aided Multi-Sensor Integrated Positioning for Vehicle Positioning in Urban Areas (submitted).

[4] F. Huang, W. Wen, H.-F. Ng, and L.-T. Hsu, "Lidar aided cycle slip detection for gnss real-time kinematic positioning in urban environments," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 1572–1578

[5] F. Huang, D. Shen, W. Wen, J. Zhang, and L.-T. Hsu, "A coarse-to-fine lidar-based slam with dynamic object removal in dense urban areas," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 3162–3172

Symbols And Acronyms

Symbols and Acronyms

Symbols

$\mathbb{R}^n$	The n-dimensional Euclidean space
$\\| \cdot \\|$	The 2-norm of a vector or matrix in Euclidean space
$\{\cdot\}^L$	The value expressed in LiDAR body frame
$\{\cdot\}^G$	The value expressed in GNSS frame, which refers to the Earth-centered, Earth-fixed (ECEF) frame
$\{\cdot\}^{EN}$	The value expressed in East-North-Up (ENU) frame. This frame is commonly used to represent the state on the Earth's surface relative to a local reference point
$\{\cdot\}^C$	The value expressed in camera frame
$\{\cdot\}^b$	The value expressed in IMU frame
$\{\cdot\}^W$	The world frame which aligned with the initial position of the vehicle
$\mathbf{x}_i$	The ground-true state at timestamp i
$\hat{\mathbf{x}}_i$	The estimated state at timestamp i
$\mathbf{T}_i$	The transformation in 6 degrees of freedom (DoF) at timestamp i, it can be divided into translational vector $\mathbf{t}_i$ and rotational matrix $\mathbf{R}_i$
$\mathcal{P}_i$	Point cloud collected at timestamp i, which is the point set of $\{\mathbf{p}_{i,1}, \mathbf{p}_{i,2}, \ldots, \mathbf{p}_{i,n}\}$. $\mathbf{p}_n = \{x_n, y_n, z_n\}$ is the 3D observation of a point n in Cartesian coordinates

Acronyms

SLAM	Simultaneous Localization And Mapping
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
ROS	Robot Operating System
RTK	Real-Time Kinematic
ECEF	Earth-Centered, Earth-Fixed
ENU	East-North-Up
NLOS	Non-line-of-sight
LiDAR	Light Detection and Ranging
LO	LiDAR Odometry
IMU	Inertial Measurement Unit
AHRS	Attitude and Heading Reference System
GT	Ground Truth
LIO	LiDAR-Inertial Odometry
VIO	Visual-Inertial Odometry
ICP	Iterative Closet Point
G-ICP	Generalized-ICP
NDT	Normal Distribution Transform
EKF	Extended Kalman Filter
LOAM	LiDAR Odometry and Mapping
FGO	Factor Graph Optimization
WLS	Weighted Least Squares
DD	Double-differenced

V2X	Vehicle-to-Everything
C-V2X	Cellular Vehicle-to-Everything
RSU	Roadside Unit
OBU	Onboard Unit
RSI	Roadside Infrastructure
CAV	Connected Autonomous Vehicle
ITS	Intelligent Transportation System
FOV	Field of View
DNN	Deep Neural Network
RMSE	Root Mean Square Error
RPE	Relative Pose Error
ATE	Absolute Translation Error
STD	Standard Deviation

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Positioning remains one of the key challenges hindering the deployment of fully autonomous mobile systems, including autonomous vehicles, autonomous mobile service robots [19], and autonomous aerial robots [20]. LiDAR is widely used to provide accurate and reliable 3D point clouds of the environment, making it a valuable tool for positioning and mapping in autonomous systems [21; 22]. Satisfactory accuracy of LO can be achieved [23] in sub-urban areas with rich environmental features and limited moving objects. However, the performance is significantly degraded [1] in challenging urban canyons with numerous moving objects. GNSS is popular for providing globally referenced positioning services. In open areas, GNSS RTK can achieve centimeter-level accuracy [24]. However, in urban canyons, positioning accuracy is severely compromised by NLOS signals and multipath effects caused by tall buildings [25; 26; 27]. To address these issues, several approaches have been proposed to mitigate [28] and correct [29] affected GNSS raw measurements to enhance positioning accuracy in urban canyons. However, the achieved accuracy still remains insufficient for the navigation requirements of fully autonomous vehicles.

With the rapid development of RSUs and 5G V2X [30; 31], vehicle-to-infrastructure cooperation has gained significant attention for its potential to enhance sensing capabilities, supporting various downstream applications in autonomous driving. Roadside infrastructure is a key component of smart cities and will be deployed at scale to enable V2X cooperation, as outlined in the smart mobility roadmap for Hong Kong [32]. However, it is still limited for the existing research on leveraging vehicle-infrastructure cooperation to improve navigation performance in challenging urban environments.

To enhance positioning performance for autonomous driving in urban canyons, the study first conducted a benchmark evaluation of publicly available LO and concluded three dominant factors that degrade the performance of LO in urban canyons (Chapter 3). The impact of dynamic objects was then addressed using both deep learning and optimization-based methods (Chapter 4.2). Further improvements in LO performance were achieved by object reweighting in highly dynamic scenarios (Chapter 4.3). However, the LO still suffers from accumulated drift after long-term mapping. Interestingly, GNSS-RTK offers meter-level accurate absolute positioning solutions that can correct this drift in urban environments. The study further utilizes consecutive relative pose estimates from LiDAR sensors to detect potential cycle slips in carrier measurements of GNSS-RTK, thereby increasing its fixing rate and accuracy (Chapter 5).

Limited by the level of intelligence of a single intelligent vehicle, the C-V2X opened a new window for the realization of fully autonomous driving. Therefore, the study proposed the use of roadside LiDAR to improve the onboard LIO positioning and mapping in GNSS-degraded urban areas (Chapter 6.2). Furthermore, a roadside GNSS-assisted integrated navigation system is proposed to mitigate GNSS error caused by NLOS and multipath (Chapter 6.3). Finally, an adaptive multi-sensor integrated solution is developed by using continuous error maps for various sensors under different conditions to enhance the positioning accuracy of connected autonomous vehicles in complex urban (Chapter 6.4). Extensive experiments are conducted both in simulation and Hong Kong C-V2X testbed to validate the effectiveness of the proposed methods.

1.2 Research Objectives

The primary objective of this study is to investigate the dynamic object-aware LiDAR odometry to enhance positioning performance for autonomous driving in urban areas. Three objectives were stated and achieved as follows:

Objective 1: This objective aims to improve the LO positioning performance by mitigating the impact of dynamic objects. To achieve this, a coarse-to-fine LiDAR-based solution for dynamic object removal is developed, utilizing both instance-level deep neural networks and point-wise discrepancy images to handle dynamic points. Furthermore, this thesis conducts a numerical analysis to assess the effects of dynamic objects on LO degradation. Subsequently, an adaptive weighting-based LO approach is proposed, where contributions from available features are comprehensively evaluated. This ensures that both static and dynamic features are reasonably utilized. Unhealthy features are automatically assigned lower weights, while healthy ones receive higher weights, preventing the false exclusion of feature subsets due to inaccurate detection of dynamic objects.

Objective 2: The objective aims to leverage the complementary strengths of GNSS-RTK and LiDAR. A common issue in GNSS-RTK positioning is cycle slips in carrier-phase measurements, which degrade performance in urban canyons due to excessive signal reflections from buildings. LiDAR sensors, however, can provide accurate relative state estimation in such environments. To address this, a LiDAR-aided cycle slip detection method for GNSS-RTK is developed, utilizing the consecutive relative poses estimated by LO. By incorporating the relative pose information from LO, the performance of GNSS-RTK is significantly enhanced, improving both the fixing rate and positioning accuracy.

Objective 3: The objective aims to explore vehicle-infrastructure cooperation to enhance the navigation performance of CAVs in urban areas. To alleviate the drift of LIO-based odometry and mapping in urban areas, the research proposes an RSI-assisted LIO for reliable odometry and mapping, which benefits from the global constraint provided by RSI LiDAR. Furthermore, integration of roadside GNSS-assisted sensor fusion is proposed to achieve reliable odometry and mapping, utilizing the high-quality DD measurements provided by nearby roadside GNSS, effectively mitigating shared random errors such as multipath and NLOS. To this end, the research introduces an adaptive multi-sensor integrated solution, employing continuous error maps for various sensors under different conditions broadcasted by RSUs to significantly enhance the positioning accuracy of CAV in complex urban.

1.3 Thesis Outline

The outline of this thesis is as follows:

Chapter 2 provides a comprehensive review of existing algorithms related to integrated navigation systems. It begins with an overview of numerous LO methods, followed by an analysis of positioning results that illustrate the current LO performance and limitations in urban areas. Subsequently, advanced sensor-integrated positioning methods are introduced to achieve reliable sensor fusion. Finally, various vehicle-to-infrastructure cooperation methods are reviewed, including cooperative navigation and cooperative perception.

Chapter 3 provides a benchmark detailed evaluation of the performance of publicly available LO pipelines using the challenging datasets collected in typical Hong Kong urban environments. Three dominant factors that degrade the performance of LO algorithms are concluded: motion difference, dynamic objects, and degree of urbanization.

Chapter 4 develops a coarse-to-fine LiDAR-based method that incorporates dynamic object removal. This approach utilizes both instance-level DNN and pointwise discrepancy images to accurately identify and mitigate the impacts of dynamic points. Furthermore, a dynamic object-aware LO is proposed to mitigate the effects of dynamic objects. It evaluates the impact of dynamic objects and adjusts the weighting of both dynamic and static objects in highly dynamic scenarios by joint weighting estimation in urban areas. This approach helps to alleviate the false detection problems raised in parameter-based and learning-based methods. Finally, the performance of both methods is evaluated through real-world experiments conducted in urban areas.

Chapter 5 presents a LiDAR-aided cycle slip detection method for GNSS-RTK, which benefits from the consecutive relative pose estimated by LO. Integer ambiguity resolution is resolved after detecting the potential cycle slips. The corresponding performance is evaluated and compared with the conventional GNSS-RTK positioning through real-world experiments conducted in a typical urban environment.

Chapter 6 introduces roadside infrastructure-assisted navigation in urban environments. Due to the limitations in the intelligence of individual autonomous vehicles, C-V2X technology opens new opportunities for achieving fully autonomous driving. The chapter begins by exploring the use of roadside LiDAR to provide precise state information, which acts as a global constraint in LIO graph-based optimization. It then integrates accurate DD observations from roadside GNSS, which are jointly optimized within a factor graph framework. Additionally, an error-map-aided multi-sensor integrated system is presented, utilizing error data gathered by sensor-equipped autonomous vehicles. This error data is uploaded to roadside infrastructure and redistributed to autonomous vehicles. The effectiveness of the proposed methods is validated through real-world experiments conducted at the Hong Kong

C-V2X testbed.

Chapter 7 includes the conclusion and future work.

1.4 Research Contributions

The research topic of this thesis is organized into five phases. The studies in each phase are summarized and published as academic articles. The flow chart for each chapter is shown in Fig. 1.1.

Chapter 3 comprehensive performance evaluation and analysis of publicly available LO pipelines using challenging datasets collected in typical urban canyons of Hong Kong and concluded three major factors dominating the performance of LO in urban canyons. The work is published in:

- F. Huang, W. Wen, J. Zhang and L. -T. Hsu, ”Point Wise or Feature Wise? A Benchmark Comparison of Publicly Available Lidar Odometry Algorithms in Urban Canyons,” in *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 155-173, Nov.-Dec. 2022. [1]

Chapter 4 develops a coarse-to-fine LiDAR-based method that includes dynamic object removal. Furthermore, a dynamic object-aware LO is proposed that adaptively assigns weightings to dynamic features, enhancing accuracy by mitigating the impacts of dynamic objects. The work is published in:

- F. Huang, D. Shen, W. Wen, J. Zhang, and L.-T. Hsu, ”A Coarse-to-Fine LiDAR-Based SLAM with Dynamic Object Removal in Dense Urban Areas,” in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, pp. 3162-3172.[16]

- F. Huang, W. Wen, J. Zhang, C. Wang and L. -T. Hsu, ”Dynamic Object-Aware LiDAR Odometry Aided by Joint Weightings Estimation in Urban Areas," in *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, pp. 3345-3359, Feb. 2024. [2]

Chapter 5 proposes to detect the cycle slip using the time-differenced carrier-phase with the help of the consecutive relative pose estimated by LO. The work is published in:

- F. Huang, W. Wen, H. -F. Ng and L. -T. Hsu, "LiDAR Aided Cycle Slip Detection for GNSS Real-Time Kinematic Positioning in Urban Environments," *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, Macau, China, 2022, pp. 1572-1578.[15]

Chapter 6 presents roadside infrastructure-assisted navigation in urban areas, which is aimed at overcoming the limitations of a single intelligent vehicle. The proposed integrated navigation solutions were validated through extensive simulations and real-world experiments. The work is published in:

- F. Huang, H. Chen, A. Urtay, D. Su, W. Wen and L. -T. Hsu, "Roadside Infrastructure assisted LiDAR/Inertial-based Mapping for Intelligent Vehicles in Urban Areas," *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, Bilbao, Spain, 2023, pp. 5831-5837.[13]

- F. Huang, W. Wen, G. Zhang, D. Su and L. -T. Hsu, "Adaptive Multi-Sensor Integrated Navigation System Aided by Continuous Error Map from RSU for Autonomous Vehicles in Urban Areas," *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, Bilbao, Spain, 2023, pp. 5895-5902.[14]

In addition to the above contributions, which address the research challenge in this thesis, I have made contributions to co-authored publications related to my thesis but will not be included.

- Hsu, L.-T., Huang, F., Ng, H.-F., Zhang, G., Zhong, Y., Bai, X., & Wen, W. (2023). Hong Kong UrbanNav: An open-source multisensory dataset for benchmarking urban navigation algorithms. *NAVIGATION*, 70(4). Project

- Chen, P., Guan, W., Huang, F., Zhong, Y., Wen, W., Hsu, L. T., & Lu, P. (2023). ECMD: An Event-Centric Multisensory Driving Dataset for SLAM. *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 407-416, 2023. Project

- Chang, J., Hu, R., Huang, F.*, Xu, D., and Hsu, L. T. (2023). LiDAR-based NDT matching performance evaluation for positioning in adverse weather conditions. *IEEE Sensors Journal*, vol. 23, no. 20, pp. 25346-25355, Oct. 15, 2023. (*corresponding author)

- Yan, P., Li, Z., Huang, F., Wen, W., and Hsu, L. T. (2025). Fault Detection Algorithm for Gaussian Mixture Noises: An Application in Lidar/IMU Integrated Localization Systems. *NAVIGATION: Journal of the Institute of Navigation*, 72(1).

- Chang, J., Zhang, Y., Fan, S., Huang, F., Xu, D., and Hsu, L. T. (2023). An Anti-spoofing Model based on MVM and MCCM for a Loosely-coupled GNSS/INS/LiDAR Kalman Filter. *IEEE Transactions on Intelligent Vehicles*, 9(1), 1744-1755.

- Chang, J., Huang, F., Zhang, L., Xu, D., and Hsu, L. T. (2023). Selection of areas for effective GNSS spoofing attacks to a vehicle-mounted MSF system based on scenario classification models. *IEEE Transactions on Vehicular Technology*, 72(11), 14645-14655.

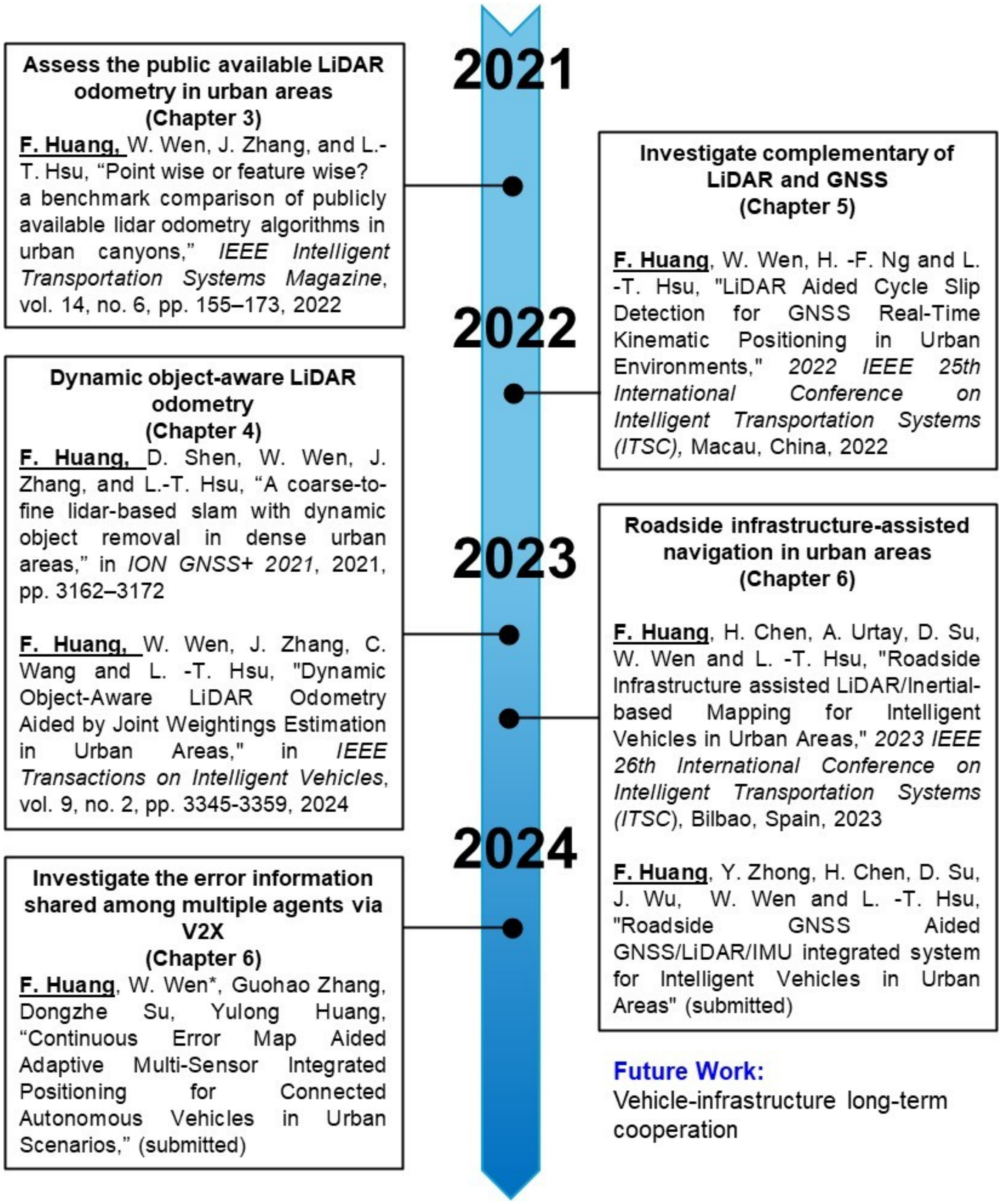
- Zhong, Y., Huang, F., Zhang, J., Wen, W., Hsu, L.-T.: Low-cost solid-state

LiDAR/inertial-based localization with prior map for autonomous systems in urban scenarios. *IET Intelligent Transport Systems*, 00, 1–13 (2022).

- 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*, 12(10):5193.

- Chang, J., Zhang, L., Hsu, L.-T., Xu, B., Huang, F., & Xu, D. (2022). Analytic Models of a Loosely-coupled GNSS/INS/LiDAR Kalman Filter considering Update Frequency under a Spoofing Attack. *IEEE Sensors Journal*, 2022.

- Zhang, J., Wen, W., Huang, F., Chen, X., Hsu, L.-T. (2021). Coarse-to-Fine Loosely-Coupled LiDAR-Inertial Odometry for Urban Positioning and Mapping. *Remote Sensing*, 13(12):2371.



Assess the public available LiDAR odometry in urban areas (Chapter 3)

F. Huang, W. Wen, J. Zhang, and L.-T. Hsu, "Point wise or feature wise? a benchmark comparison of publicly available lidar odometry algorithms in urban canyons," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 155–173, 2022

Investigate complementary of LiDAR and GNSS (Chapter 5)

F. Huang, W. Wen, H. -F. Ng and L. -T. Hsu, "LiDAR Aided Cycle Slip Detection for GNSS Real-Time Kinematic Positioning in Urban Environments," *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC),* Macau, China, 2022

Dynamic object-aware LiDAR odometry (Chapter 4)

F. Huang, D. Shen, W. Wen, J. Zhang, and L.-T. Hsu, "A coarse-to-fine lidar-based slam with dynamic object removal in dense urban areas," in *ION GNSS+ 2021*, 2021, pp. 3162–3172

F. Huang, W. Wen, J. Zhang, C. Wang and L. -T. Hsu, "Dynamic Object-Aware LiDAR Odometry Aided by Joint Weightings Estimation in Urban Areas," in *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, pp. 3345-3359, 2024

Roadside infrastructure-assisted navigation in urban areas (Chapter 6)

F. Huang, H. Chen, A. Urtay, D. Su, W. Wen and L. -T. Hsu, "Roadside Infrastructure assisted LiDAR/Inertial-based Mapping for Intelligent Vehicles in Urban Areas," *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC*), Bilbao, Spain, 2023

F. Huang, Y. Zhong, H. Chen, D. Su, J. Wu, W. Wen and L. -T. Hsu, "Roadside GNSS Aided GNSS/LiDAR/IMU integrated system for Intelligent Vehicles in Urban Areas" (submitted)

Investigate the error information shared among multiple agents via V2X (Chapter 6)

F. Huang, W. Wen*, Guohao Zhang, Dongzhe Su, Yulong Huang, "Continuous Error Map Aided Adaptive Multi-Sensor Integrated Positioning for Connected Autonomous Vehicles in Urban Scenarios," (submitted)

Future Work:
Vehicle-infrastructure long-term cooperation

Figure 1.1: Overall flowchart of each chapter of this thesis.

Chapter 2

Literature Review

Reliable localization and mapping are crucial for achieving fully autonomous driving. As outlined in Chapter 1, the research aims to investigate integrated navigation systems that enhance positioning performance for autonomous vehicles operating in urban areas. This chapter begins by introducing existing LO methods, focusing specifically on how raw point clouds are modeled and processed. We will explore various techniques for feature extraction and point cloud registration, highlighting the strengths and weaknesses of current approaches. The second part of this chapter delves into works related to sensor integration, where we examine how data from multiple sensors, such as GNSS, LiDAR, IMU, and cameras can be fused to improve overall positioning accuracy. We will discuss various algorithms and frameworks that leverage the complementary strengths of these sensors to mitigate the limitations of single sensors. Finally, we present existing research on vehicle-infrastructure cooperation, highlighting the importance of collaboration between intelligent vehicles and roadside infrastructure.

2.1 LiDAR Odometry

LO methods can be categorized into two groups based on how raw point clouds are modeled: point-wise and feature-wise methods. Point-wise methods estimate the relative transformation directly from the raw points, while feature-wise methods, such as those in [18], extract representative edge and planar features from the raw point clouds. In other words, the distinction between point-wise and feature-wise methods is based on whether all raw points are directly utilized for further data association. A summary of publicly available LO methods and their key properties is provided in Table 2.1.

One of the most widely recognized point-wise LO methods is the ICP algorithm [33]. ICP is a modular, straightforward algorithm that aligns two point cloud frames by identifying point-to-point correspondences. However, a significant drawback of ICP is its high computational cost, particularly when registering dense point clouds. The performance of ICP is highly dependent on the initial alignment guess due to the non-convex nature of its optimization [42]. Outliers from moving objects can further exacerbate this issue by introducing additional non-convexity. To address these issues, several variants of ICP have been developed to improve both efficiency and accuracy, such as Trimmed ICP [43] and Normal ICP [44]. Among these ICP variants, the G-ICP [34] is one of the most popular due to its notable accuracy. Unlike standard ICP, which directly optimizes point-wise correspondences, G-ICP leverages both standard ICP and the point-to-plane method introduced by Chen and Medioni [45], optimizes the transformation via a distribution-to-distribution fashion. This allows G-ICP to account for geometric correlations between points, making it less sensitive to the initial guess compared to conventional ICP. Despite the high accuracy of G-ICP, its optimization process suffers from the same limitation as ICP and its variants, namely the reliance on the time-intensive nearest neighbor search (NNS). As stated in [35], NNS is the dominant factor affecting the computational efficiency of G-ICP.

Rather than depending on the time-intensive NNS process, NDT [36], another prominent point-wise registration method, uses a voxel-based correspondence asso-

Table 2.1: Overview of the selected publicly available LO methods.

Style	**Method**	**Modeling**[1]	**Optimization**[2]	**Year**
Point-wise	Besl et al, ICP [33]	minimize the point-to-point distance	s2s, SVD (PCL version)	1992
	Segal et al, G-ICP [34]	surface-based distribution-to-distribution	s2s, BFGS and FLANN (PCL version)	2009
	Koide et al, FastGICP [35]	surface-based distribution-to-distribution	s2s, Gauss-Newton, multi-threaded	2020
	Koide et al, VG-ICP [35]	voxel-based distribution-to-multi-distribution	s2s, Gauss-Newton, multi-threaded	2020
	Koide et al, FastVG-ICPCuda [35]	voxel-based distribution-to-multi-distribution	s2s, Gauss-Newton, CUDA-optimized	2020
	Biber et al, NDT [36]	voxel-based point-to-distribution	s2s, Newton method	2003
	Koide et al, NDT-OMP [37]	voxel-based point-to-distribution	s2s, Newton method	2019
Feature-wise	Zhang et al, LOAM [18]	minimize the distance of feature points	s2s and s2m, LM	2014
	Qin et al, A-LOAM [38]	minimize the distance of feature points	s2s and s2m, ceres-solver	2018
	Shan et al, LeGO-LOAM [39]	minimize the distance of feature points, add ground-optimization	s2s and s2m, LM	2018
	Ye et al, LIO-Mapping [40]	minimize the distance of feature points	s2s and s2m, LM	2019
	Han et al, Fast LOAM [41]	minimize the distance of feature points	s2m, ceres-solver	2020

[1] Modeling represents how to model the transformation function.
[2] Optimization: s2s refers to scan-to-scan, while s2m refers to scan-to-map (also known as scan-to-model in some references). PCL stands for Point Cloud Library. The abbreviations SVD, BFGS, FLANN, and LM correspond to Singular Value Decomposition, Broyden–Fletcher–Goldfarb–Shanno, Fast Library for Approximate Nearest Neighbors, and Levenberg–Marquardt, respectively.

ciation to estimate transformations. NDT divides raw and discrete point clouds into multiple voxels, modeling each voxel with a Gaussian distribution. The transformation is then estimated by mapping other point clouds onto the voxelized point cloud. Overall, NDT is significantly faster than typical ICP. In a performance comparison [46], NDT demonstrated superior accuracy over ICP on the evaluated dataset. However, the performance of the NDT is sensitive to the selection of the resolution of voxels. In other words, careful tuning of the voxel size is necessary based on sensor and environmental conditions. Meanwhile, although NDT avoids the time-consuming NNS, it still struggles to achieve real-time performance when dealing with large, dense point clouds. To address this, researchers at Toyohashi University of Technology have worked on improving the computational efficiency of both G-ICP and NDT. The study in [37] introduced NDT-OMP, a multi-threaded accelerated version of NDT, achieving efficient point cloud registration with real-time performance. Additionally, the follow-up work in [35] proposed the Voxelized Generalized Iterative Closest Point (VGICP) algorithm, built upon conventional G-ICP, which also achieves real-time performance. VGICP's novel voxelization method aggregates the distribution of all points in a voxel, enabling parallel implementation. According to the evaluation in [35], VGICP demonstrated similar or even superior performance compared to G-ICP.

In short, G-ICP is a highly popular and accurate variant of ICP. VGICP addresses the computational challenges of G-ICP by reducing the load associated with nearest neighbor search (NNS). NDT is another commonly employed point-wise method, and the development of NDT-OMP allows for real-time execution of the standard NDT approach.

Instead of estimating the transformation directly from all raw points, feature-based LO methods [18; 39] focus on extracting representative features from the raw point cloud. The Fast Point Feature Histogram (FPFH) [47] was introduced to extract and describe the features within the point cloud. FPFH enables the exploration of local geometry, with the transformation optimized by matching FPFH-based correspondences. A similar feature descriptor, histograms of OrienTations (SHOT), is studied in [48] for feature extraction. Other works in [49; 50] optimized the ground surface features using multiple roadside LiDARs. Feature-based LO methods extract a smaller subset of features compared to the full raw point cloud, resulting in a considerably reduced computational load for matching, especially when compared to the optimization processes of G-ICP and NDT. Additionally, feature-based methods are less sensitive to the initial guess than point-wise methods. However, their performance relies heavily on accurate feature detection, and any misdetection can result in incorrect feature associations. Theoretically, because feature-based methods only utilize a subset of the raw points, their convergence accuracy [35] tends to be lower than that of point-wise LO methods.

LOAM [18], a prominent feature-based LO method, was first introduced by the Carnegie Mellon University team in 2014. While LOAM theoretically incorporates aspects of both point-wise and feature-wise methods, we classify it as feature-wise in this section. To reduce the computational load typical of ICP, LOAM extracts two types of features: edge and planar. These features are extracted based on the smoothness of a small region near a given feature point. Different from FPFH or SHOT, which provide multiple feature categories based on descriptors, LOAM uses only two feature types. Nevertheless, LOAM retains the advantage of feature-based LO methods by avoiding the use of all raw points, enabling more efficient registration. LOAM matches extracted edge features to a previously maintained dense edge feature map, and planar features to a corresponding planar feature map, resulting in scan-to-map matching, which differs from the scan-to-scan matching typical of ICP. Interestingly, LOAM evaluates matches based on the Euclidean distance between features rather than descriptor smoothness, as is the case with point-wise registration. This can significantly decrease the mis-association of features and increase the robustness of the matching. As a result, LOAM combines the low computational load of feature-wise methods with the robustness of point-wise methods. Due to its strengths, LOAM is ranked 3rd on the KITTI benchmark as of September 2024. Although the performance evaluation among the KITTI dataset is prominent, the accuracy is not guaranteed in the diverse operation scenarios [39; 51]. The work in [39] argues that can exhibit significant drift in the altitude direction due to limited features. To address this, LeGO-LOAM was proposed to optimize the altitude state based on detected ground points, resulting in reduced drift in the evaluated dataset. Several variants of LOAM have been introduced to improve computational efficiency and accuracy. A-LOAM [38] accelerates LOAM by replacing its complex Jacobian derivation with the state-of-the-art non-linear optimization solver, Ceres-solver [52].Fast-LOAM [41], on the other hand, is a faster implementation that removes the odometry process and relies solely on the mapping process to estimate transformation. LIO-Mapping [40] presents another LOAM implementation with carefully tuned parameters for feature selection and extraction. However, while these variants have improved certain aspects of LOAM, they have not fundamentally addressed its core limitations. As mentioned above, LOAM remains dependent on the availability of edge and planar features, utilizing only a portion of the raw point clouds. The authors in [35] suggest that combining feature-based LO methods (e.g., LOAM) with point-based LO methods (e.g., VGICP) offers a promising approach to ensure both accuracy and robustness, where a coarse feature-based registration is followed by a fine point-based registration. In short, LOAM has dominated LO methods on the KITTI dataset since 2015 due to three key factors: (1) The extraction and matching of edge and planar features ensure both real-time performance and accuracy; (2) the scan-to-map matching between keyframe features and the historical dense feature map enhances the precision of correspondence identification. Table 2.3 presents the positioning error of LOAM using the Hong Kong UrbanNav [3] dataset. An RPE RMSE of 0.35 m and 0.80 m was achieved using LOAM in middle-class urban areas and deep urban areas, respectively. Interestingly, a reduction in RPE RMSE was observed in harsh urban scenarios. A possible explanation for this is that taller buildings in harsh urban environments may provide denser features for the LiDAR. However, accumulated drift is observed in all urban areas, leading to significant absolute translation errors. It highlights the need for advanced algorithms to improve LO-based methods in urban environments. Moreover, LO failed on the tunnel dataset due to the limited features available in tunnel scenes. Therefore, it is essential to enhance LO-based methods in urban areas by developing advanced algorithms.

Table 2.3: Positioning error of the LOAM [18] using UrbanNav [3] datasets.

	Relative Pose Error				**Absolute Translation Error**			
Dataset	**RMSE (m)**	**Mean (m)**	**STD (m)**	**Max (m)**	**RMSE (m)**	**Mean (m)**	**STD (m)**	**Max (m)**
Middle-class Urban	0.35	0.17	0.30	5.33	20.07	16.49	11.45	46.31
Deep Urban	0.80	0.17	0.78	9.98	110.29	101.33	43.55	134.65
Harsh Urban	0.17	0.07	0.16	3.09	22.88	22.19	5.59	32.95
Tunnel	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail

Recent advancements in Neural Networks [53] offer a promising approach to enhancing the performance of LO methods by more deeply exploring features within point clouds. To address dynamic objects in scenes, LO-Net [54] was introduced, incorporating a mask-weighted geometric constraint loss, achieving results comparable to LOAM. Another recent work, LiDAR Odometry and Mapping using Neural Implicit Representation [55] was proposed in 2023. For the broader generality of learning-based methods, an unsupervised LO method [56] based on the geometric consistency of point clouds was introduced by the team from KAIST.

2.2 Sensor Integration

Significant efforts have been dedicated to achieving reliable sensor fusion. [57; 58; 59; 60]. A common approach is integrating inertial sensors with LiDAR [61] to provide an initial estimation for scan matching. VIO [62] combines high-frequency inertial sensor data for robust pose estimation. However, both LIO and VIO are prone to accumulated errors in degraded environments [9; 63]. GNSS positioning can mitigate drift accumulation [9], but its accuracy is significantly degraded in deep urban canyons [25]. Each sensor type has specific advantages and drawbacks depending on the environment, and the challenge remains to effectively fuse these sensors for autonomous vehicles in urban environments. The integration of multiple sensors strengthens the system's resilience in handling difficult scenarios [63], especially in cases of partial sensor failure or performance degradation. The Multi-State Constraint Kalman Filter (MSCKF) [64] is a filter-based VIO framework designed for fast pose estimation, though it is sensitive to time synchronization. Optimization-based methods such as VINS [62] can address synchronization issues and optimize historical frames using local maps or sliding windows through local maps or sliding windows to obtain a globally optimized solution.

To address the accumulated error in LIO, integrating GNSS offers a promising solution. Loosely coupled GNSS-LIO methods independently estimate GNSS, LiDAR, and IMU data before fusing these estimates within the positioning domain. The study in [61] proposed a LiDAR/inertial system that incorporates GNSS data characterized by low positioning uncertainty. Reference [9] presented an adaptive integration strategy for GNSS-RTK and LIDAR/IMU, which selects reliable GNSS-RTK solutions by employing an elevation angle mask derived from the surrounding point cloud data. Raw measurements from GNSS, LiDAR, and IMU sensors are integrated into the tightly coupled approaches, which enables separate mitigation of unhealthy GNSS measurements while maintaining healthy measurements in one single epoch. The research in [65] presented a tightly coupled PPP-GNSS/INS/LiDAR method by accurate outlier exclusion based on spatiotemporal consistency checks of sensor data. In [66], a GNSS/INS/LiDAR integration is proposed by considering the cycle slip and outlier detection with the precise orbit and clock products. Nonetheless, the performance of such methods depends on outlier detection which requires a good initial guess which is not always guaranteed in complex urban environments.

Recent studies have demonstrated the potential of deep learning-based methods [53] for efficient weighting estimation in multi-sensor integration. DeLS-3D [67] is a deep-learning framework that integrates data from cameras, GPS/IMU, and semantic maps to enhance robustness and accuracy. An unsupervised method [68] was proposed to estimate camera motion using stereo images and IMU data. Lvio-Fusion [69] applied reinforcement learning to optimize sensor weighting by training on ground truth positioning data. However, the generalization of deep learning-based methods for estimating sensor weighting coefficients in various challenging scenarios remains an open issue. Therefore, it is necessary to improve the multi-sensor integrated positioning performance in urban areas by estimating the weighting of different measurements.

2.3 Vehicle-Infrastructure Cooperation

Recent advancements in V2X technology have greatly enhanced the potential of smart mobility. The study in [30] explored various V2X applications and their requirements for next-generation vehicular mobility. In [70], a method was proposed that combines roadside-assisted received signal strength (RSS) with GNSS signals to improve V2I-based localization on a simulated platform. The study in [71] introduced a background filtering technique integrated with 3D object detectors for vehicle detection using roadside LiDAR in mining environments. In [72], a V2X perception framework utilizing a visual transformer was developed and tested through the CARLA simulator [73]. Another study [74] employed roadside LiDAR to detect and track both pedestrians and vehicles. In 2022, Tsinghua University released a real-world V2X collaborative autonomous driving dataset, and in 2023, the TUM traffic dataset [75] was open-sourced, featuring labeled LiDAR and camera data from two roadside units at an intersection. The work [76] aligned point clouds between ego vehicles and roadside infrastructure using semantic traffic elements in campus scenarios. A follow-up study [77] in 2023 proposes using roadside LiDAR to enhance on-vehicle HD mapping in both simulation and campus environments. Previous research [78] demonstrated that sensor measurement sharing between multiple agents can enhance positioning performance in dense urban environments. However, there is limited research has explored using vehicle infrastructure cooperation to improve LIO positioning in challenging urban areas such as Hong Kong.

In urban environments, the GNSS measurements in the same region share similar random errors [79] such as NLOS and multipath. The study in [80] employed a single roadside GNSS station to enhance DGNSS and GNSS RTK performance via DSRC communication. Similarly, the work in [81] proposed a GPS/IMU/V2X observation fusion framework using an LSTM network, simulating multiple V2X nodes. Another research presented a cooperative positioning solution to enhance the accuracy of low-cost GNSS receivers in obstructed areas by using multi-attribute decision-making (MADM) methodology to rank neighboring vehicles. A novel GNSS-based collaborative positioning method, enhanced by 3D mapping and factor graph optimization, is proposed [79] to improve positioning accuracy in urban areas. Despite these advancements, there is still limited research on leveraging multiple roadside GNSS to enhance positioning performance in sensor integration, especially in challenging urban environments such as Hong Kong.

Chapter 3

Benchmark Comparison of Publicly Available Lidar Odometry Algorithms in Urban Canyons

3.1 Introduction

Both point-wise and feature-wise registration methods have their advantages and limitations. The LO methods listed in Table 2.1 will be theoretically compared and experimentally evaluated in the following sections.

3.2 Method

3.2.1 ICP

The core idea behind ICP [33] is to determine the transformation between a source and target point cloud by minimizing the distance error between corresponding points. Fig. 3.1 illustrates an example of point cloud registration between two successive frames. The cost function of ICP is as follows,

$$\mathbf{T}_{k+1}^{k} = \arg\min \frac{1}{2} \sum_{i=1}^{N} \left\| \mathbf{p}_{k,i} - \left(\mathbf{R}_{k+1}^{k} \mathbf{p}_{k+1,i} + \mathbf{t}_{k+1}^{k} \right) \right\|^2 \tag{3.1}$$

Firstly, the ICP calculates the centroid of the point cloud $\bar{\mathcal{P}}_k$ and $\bar{\mathcal{P}}_{k+1}$.

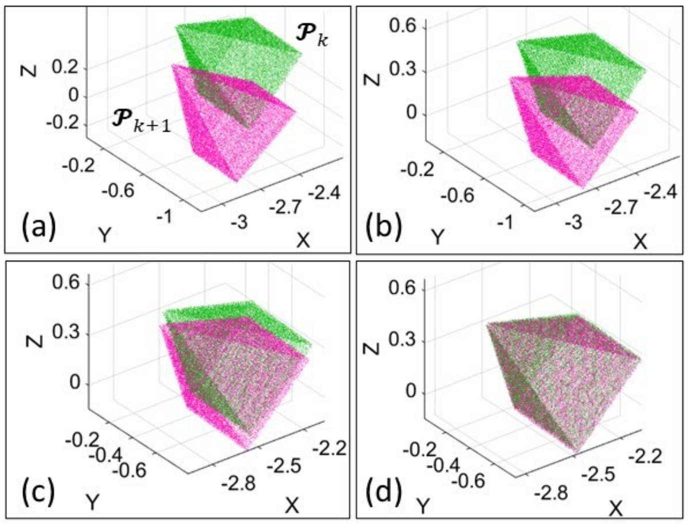


Figure 3.1: Point cloud $\mathcal{P}_k$ and $\mathcal{P}_{k+1}$ captured from two successive scans. (a) represents the initial scan, while (b) and (c) depict the transformations after multiple iterations. (d) presents the final result of the point set alignment.

$$\bar{\mathbf{p}}_k = \frac{1}{n}\sum_{i=1}^{n}\mathbf{p}_{k,i} \tag{3.2}$$

Then subtract the center of mass from consecutive point sets to obtain the decentering point set $\mathbf{p}'_{k,i} \in \mathcal{P}'_k$ and $\mathbf{p}'_{k+1,i} \in \mathcal{P}'_{k+1}$ corresponding to $\mathcal{P}_k$ and $\mathcal{P}_{k+1}$, respectively [33],

$$\mathbf{p}'_{k,i} = \mathbf{p}_{k,i} - \bar{\mathbf{p}}_k, \tag{3.3}$$

Modifying Equation 3.1 based on Equation 3.2 and Equation 3.3, we can have,

$$\begin{aligned} &\frac{1}{2}\sum_{i=1}^{N}\left\|\mathbf{p}_{k,i} - \left(\mathbf{R}^k_{k+1}\mathbf{p}_{k+1,i} + \mathbf{t}^k_{k+1}\right)\right\|^2 \\ = &\frac{1}{2}\sum_{i=1}^{N}\left\|\mathbf{p}_{k,i} - \left(\mathbf{R}^k_{k+1}\mathbf{p}_{k+1,i} + \mathbf{t}^k_{k+1}\right) - \bar{\mathbf{p}}_k + \mathbf{R}^k_{k+1}\bar{\mathbf{p}}_{k+1} + \bar{\mathbf{p}}_k - \mathbf{R}^k_{k+1}\bar{\mathbf{p}}_{k+1}\right\|^2 \end{aligned} \tag{3.4}$$

Then, the right side of Equation 3.4 can be simplified to,

$$\frac{1}{2}\sum_{i=1}^{N}\left\|(\mathbf{p}_{k,i}-\bar{\mathbf{p}}_k)-\left(\mathbf{R}_{k+1}\left(\mathbf{p}_{k+1,i}-\bar{\mathbf{p}}_{k+1}\right)\right)\right\|^2+\left\|\bar{\mathbf{p}}_k-\left(\mathbf{R}_{k+1}\bar{\mathbf{p}}_{k+1}+\mathbf{t}_{k+1}^{k}\right)\right\|^2 \tag{3.5}$$

Derive the revised cost function by recalling Equation 3.4 and Equation 3.5,

$$\mathbf{T}_{k+1}^{k}=\arg\min\frac{1}{2}\sum_{i=1}^{N}\left\|\mathbf{p}'_{k,i}-\mathbf{R}\bar{\mathbf{p}}_{k+1,i}\right\|^2+\left\|\bar{\mathbf{p}}_k-\mathbf{R}\bar{\mathbf{p}}_{k+1}-\mathbf{t}\right\|^2 \tag{3.6}$$

Finally, the commonly used singular value decomposition (SVD) [82] is used to compute the estimate translation and rotation iteratively.

3.2.2 Generalized-ICP

G-ICP [34] enhances the traditional ICP by integrating both standard ICP and 'point-to-plane ICP' into a distribution-to-distribution matching framework. G-ICP models each measured point $\hat{\mathbf{p}}_{k,i}$ in $\mathcal{P}_k$ and $\hat{\mathbf{p}}_{k+1,i}$ in $\mathcal{P}_{k+1}$ as being sampled from Gaussian distributions: $\mathbf{p}_k \sim \mathcal{N}(\hat{\mathbf{p}}_{k,i}, \mathbf{C}_{k,i})$ and $\mathbf{p}_{k+1} \sim \mathcal{N}(\hat{\mathbf{p}}_{k+1,i}, \mathbf{C}_{k+1,i})$, where $\mathbf{C}_{k,i}$ can be calculated from the covariance of the points.

$$\begin{aligned}\mathbf{C}_{k,i}&=\begin{bmatrix}\operatorname{cov}(x,x) & \operatorname{cov}(x,y) & \operatorname{cov}(x,z)\\ \operatorname{cov}(y,x) & \operatorname{cov}(y,y) & \operatorname{cov}(y,z)\\ \operatorname{cov}(z,x) & \operatorname{cov}(z,y) & \operatorname{cov}(z,z)\end{bmatrix}\\ &=\begin{bmatrix}\mathbb{E}(x^2)-\mu_x^2 & \mathbb{E}(xy)-\mu_x\mu_y & \mathbb{E}(xz)-\mu_x\mu_z\\ \mathbb{E}(y,x)-\mu_x\mu_y & \mathbb{E}(y^2)-\mu_y^2 & \mathbb{E}(yz)-\mu_y\mu_z\\ \mathbb{E}(x,z)-\mu_x\mu_z & \mathbb{E}(y,z)-\mu_y\mu_z & \mathbb{E}(z^2)-\mu_z^2\end{bmatrix}\end{aligned} \tag{3.7}$$

μ_x, μ_y, and μ_z are the expected values of $\mathbf{p}_{k,i}$, roughly equal to the average of its k neighboring points (e.g., $k = 20$ by default in PCL's k-d tree search). The operator $\mathbb{E}(*)$ is used to compute the expected value of a given component.

The residual for G-ICP can be expressed as,

$$\hat{\mathbf{d}}_i = \hat{\mathbf{p}}_{k,i} - \mathbf{T}_{k+1}\hat{\mathbf{p}}_{k+1,i} \tag{3.8}$$

The residual is assumed to follow a Gaussian distribution, reflecting the geometric relationship between the target and its neighboring points. If $\mathbf{p}_{k,i}$ and $\mathbf{p}_{k+1,i}$ are considered independent and Gaussian distributed, the distribution of $\mathbf{d}_i$ can be expressed as,

$$\begin{aligned}\mathbf{d}_i &\sim \mathcal{N}\left(\hat{\mathbf{x}}_{k,i} - \mathbf{T}_{k+1}\hat{\mathbf{p}}_{k+1,i}, \mathbf{C}_{(k,i)} + \mathbf{T}_{k+1}\mathbf{C}_{k+1,i}{\mathbf{T}_{k+1}}^T\right) \\ &= \mathcal{N}\left(0, \mathbf{C}_{k,i} + \mathbf{T}_{k+1}\mathbf{C}_{k+1,i}{\mathbf{T}_{k+1}}^T\right)\end{aligned} \tag{3.9}$$

Then, we use a maximum likelihood estimation (MLE) to compute the transformation $\mathbf{T}_{k+1}^k$ iteratively,

$$\mathbf{T}_{k+1}^k = \arg\max \sum_{i=1}^{N} \log\left(p(\mathbf{d}_i)\right) \tag{3.10}$$

$$= \arg\min \sum_{i=1}^{N} \mathbf{d}_i^T \left(\mathbf{C}_{k,i} + \mathbf{T}_{k+1}^k \mathbf{C}_{k+1,i} \left(\mathbf{T}_{k+1}^k\right)^T\right)^{-1} \mathbf{d}_i \tag{3.11}$$

3.2.3 VGICP

Voxelized GICP (VGICP) [35] enhances G-ICP by incorporating voxelization, which significantly reduces processing time while maintaining accuracy. First, Equation 3.13 is extended to compute residual d_i' between $\mathbf{p}_{k+1,i}$ and its neighbor points within distance r in $\mathbf{p}_k$,

$$\{\mathbf{p}_{k,j} \mid \|\mathbf{p}_{k,j} - \mathbf{p}_{k+1,i}\| < r\} \tag{3.12}$$

$$\hat{d}'_i = \sum_{j=1}^{N_i} \left(\hat{\mathbf{p}}_{k,j} - \mathbf{T}_{k+1}\hat{\mathbf{p}}_{k+1,i}\right) \tag{3.13}$$

Similar to Equations 3.10 and 3.11, the distribution of d'_i can be represented as [35],

$$\mathbf{d}'_i \sim \mathcal{N}\left(0, \sum_{j=1}^{N_i} \left(\mathbf{C}_{k,j} + \mathbf{T}_{k+1}\mathbf{C}_{k+1,i}\mathbf{T}_{k+1}{}^{T}\right)\right) \tag{3.14}$$

where the calculation of the covariance is similar to G-ICP, the cost function can be expressed as follows,

$$\begin{aligned}\mathbf{T}_{k+1}^{k} = \arg\min \sum_{i=1}^{N} N_i \left(\left(\frac{\sum \mathbf{p}_{k,j}}{N_i} - \mathbf{T}_{k+1}\mathbf{p}_{k+1,i}\right)^{T} \frac{\sum \mathbf{C}_{k,j}}{N_i}\right. \\ \left. + \mathbf{T}_{k+1}\mathbf{C}_{k+1,i}\mathbf{T}_{k+1}{}^{T}\right)^{-1} \left(\frac{\sum \mathbf{p}_{k,i}}{N_i} - \mathbf{T}_{k+1}\mathbf{p}_{k+1,i}\right)\end{aligned} \tag{3.15}$$

The equation above can be further converted into the voxel-based calculation by substituting $\bar{\mathbf{p}}_{k,i}^{\text{voxel}} = \frac{\sum \mathbf{p}_{k,i}}{N_i}$ and $\mathbf{C}_{k,i}^{\text{voxel}} = \frac{\sum \mathbf{C}_{k,j}}{N_i}$ respectively in each voxel. Finally, the MLE is used to compute the transformation $\mathbf{T}_{k+1}^{k}$ iteratively as Equations 3.10 and 3.11.

Compared to point correspondence models in ICP or GICP, VGICP optimizes data association by leveraging voxel association, which is significantly faster than the KD-tree-based nearest neighbor search used in ICP. Specifically, if a voxel contains ten points, the computational load for voxel correspondences can be reduced by a factor of ten compared to ICP or GICP. As a result, VGICP is much more efficient than traditional point-wise methods [34]. However, VGICP requires additional time for KD-tree-based nearest neighbor search to estimate the covariance matrix for each point [35].

3.2.4 NDT

Unlike ICP, which performs point-to-point registration, NDT [36] begins by dividing the point cloud into voxels. For each voxel, a probability density function (PDF) is generated to represent the likelihood that a point $\mathbf{p}_{k,i}$ is contained within a voxel in $\mathcal{P}_k$. The covariance matrix of the points inside a cell of $\mathcal{P}_k$ is then calculated to model the distribution of the points.

$$\mathbf{C}_{k,i}^{\text{ndt\_voxel}} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{p}_{k,i} - \bar{\mathbf{p}}_{k,i}^{\text{voxel}}\right)\left(\mathbf{p}_{k,i} - \bar{\mathbf{p}}_{k,i}^{\text{voxel}}\right)^T \tag{3.16}$$

The calculation of the covariance $\mathbf{C}_{k,i}^{\text{ndt\_cell}}$ in NDT differs from $\mathbf{C}_{k,i}^{\text{voxel}}$ in VGICP. In NDT, $C_{k,i}^{\text{ndt\_cell}}$ considers only the points within the voxel, requiring at least four points per cell. This results in a point-to-distribution correspondence, where each point is associated with the distribution constructed in the target frame, as shown in Table 2.1. In contrast, VGICP employs a single-to-multiple distribution approach by using the average covariance of points within the same voxel.

The PDF can be expressed as,

$$p(\mathbf{p}_{k,i}) = \frac{1}{c}\exp\left(-\frac{1}{2}\left(\mathbf{p}_{k,i} - \bar{\mathbf{p}}_{k,i}^{\text{voxel}}\right)^T\left(\mathbf{C}_{k,i}^{\text{ndt\_voxel}}\right)^{-1}\left(\mathbf{p}_{k,i} - \bar{\mathbf{p}}_{k,i}^{\text{voxel}}\right)\right) \tag{3.17}$$

where c is a constant and can be set to one [36]. An example of 2D-NDT is shown in Fig. 3.2.

The NDT of the scan $\mathcal{P}_k$ is constructed. The maximum matching score function for all points in $\tilde{\mathcal{P}}_{k+1}$ is then reformulated to minimize the negative sum of the scores,

$$\mathbf{T}_{k+1}^{k} = \arg\min\frac{1}{2}\sum_{i=1}^{N}\left(\tilde{\mathbf{p}}_{k+1,i} - \bar{\mathbf{p}}_{k,i}^{\text{voxel}}\right)^T\left(\mathbf{C}_{k,i}^{\text{ndt\_cell}}\right)^{-1}\left(\tilde{\mathbf{p}}_{k+1,i} - \bar{\mathbf{p}}_{k,i}^{\text{voxel}}\right) \tag{3.18}$$

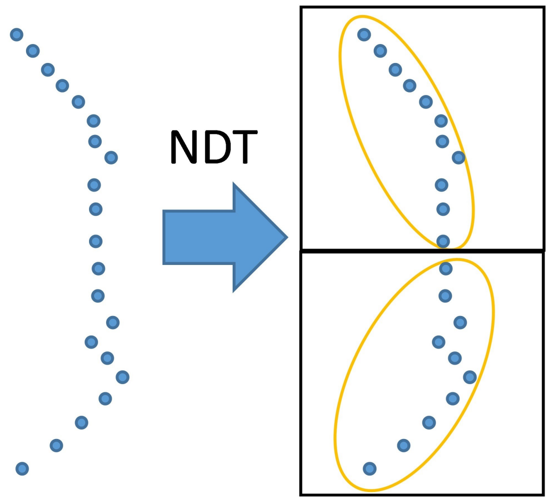


Figure 3.2: Points are divided into multiple cells and their distribution in a voxel in 2D-NDT.

The $\mathbf{T}_{k+1}^{k}$ is updated iteratively using the Newton method to minimize the score. Different from the ICP, NDT models the geometry of 3D point clouds using multiple Gaussian components, with each component representing a voxel. However, the performance of NDT is sensitive to the choice of voxel size.

NDT-OMP is a modified implementation of NDT in PCL, utilizing OpenMP for parallel processing with SSE optimization and multi-threading [37]. This enhancement makes NDT-OMP up to 10 times faster than the original PCL version.

3.2.5 LOAM

LOAM [18] primarily consists of three key processes: feature extraction, lidar odometry, and lidar mapping process.

Feature Extraction

Let m indicate the ring number of point cloud $\mathcal{P}_k$. $\mathbf{S}_{k,i}^{m}$ represent a set of continuous neighboring points of $\mathbf{p}_{k,i}$ within scan ring m. Typically, points in a ring are ordered either clockwise or counterclockwise based on their receiving time during a scan period (usually 0.1 seconds). Feature points are extracted based on the curvature c_i

of point $\mathbf{p}_{k,i}$ and its neighboring points [18].

$$c_i = \frac{1}{N_s * \|\mathbf{p}_{k,i}\|} \left\| \sum_{j \in \mathbf{S}^m_{k,i}, j \neq i} (\mathbf{p}_{k,i} - \mathbf{p}_{k,j}) \right\| \tag{3.19}$$

the $\mathbf{p}_k$ denotes the consecutive point of $\mathbf{p}_{k,i}$ within subset $\mathbf{S}^m_{k,i}$. N_s represents the number of points in $\mathbf{S}^m_{k,i}$, including $\mathbf{p}_{k,i}$ and ten consecutive points. A point is selected as the edge point if its curvature value is larger than a pre-determined threshold c_{e-th}, or classified as a planar point by a smaller curvature in $\mathcal{P}_k$. For points $\mathbf{p}^m_{k,i}$ within ring $\mathcal{P}^m_{k,i}$ in each scan, we normally divide the ring into four to eight subregions $\mathcal{P}^{s,m}_{k,i}$ and each subregion selects two edge points $\mathbf{p}^e_{k,i}$ from $\mathcal{P}^{e,m}_{k,i}$ and four planar points $\mathbf{p}^p_{k,i}$ within $\mathcal{P}^{p,m}_{k,i}$ for odometry process [18], as shown in Equation 3.20. Ten times edge points and enormous planar features are utilized in mapping process to achieve better accuracy but bring more computational cost. An example of feature extraction results is shown in Fig. 3.3.

$$\mathbf{p}^{s,m}_{k,i} = \begin{cases} \mathbf{p}^e_{k,i} \in \mathcal{P}^{e,m}_{k,i}, & c_i > c_{e-\text{th}} \text{ and } N^e_{k,i} \leqslant 2 \\ \mathbf{p}^p_{k,i} \in \mathcal{P}^{p,m}_{k,i}, & c_i < c_{e-\text{th}} \text{ and } N^p_{k,i} \leqslant 4 \end{cases} \tag{3.20}$$

Lidar Odometry

In general, odometry is estimated by accumulating the transformations between consecutive point cloud frames. In LOAM, the role of LiDAR odometry is to estimate the motion between two successive sweeps. The estimated transformation $\hat{\mathbf{T}}_{k+1}$ is used to correct the distortion of points in $\mathcal{P}_{k+1}$ and provide the initial guess for projecting $\mathcal{P}_k$ as $\hat{\mathcal{P}}_k$. In the next frame, corresponding features are found between $\hat{\mathcal{P}}_k$ and $\mathcal{P}_{k+1}$.

For each edge point $\mathbf{p}^e_{k+1,i}$, its nearest neighbors in $\hat{\mathcal{P}}_k$ are searched to fit a line

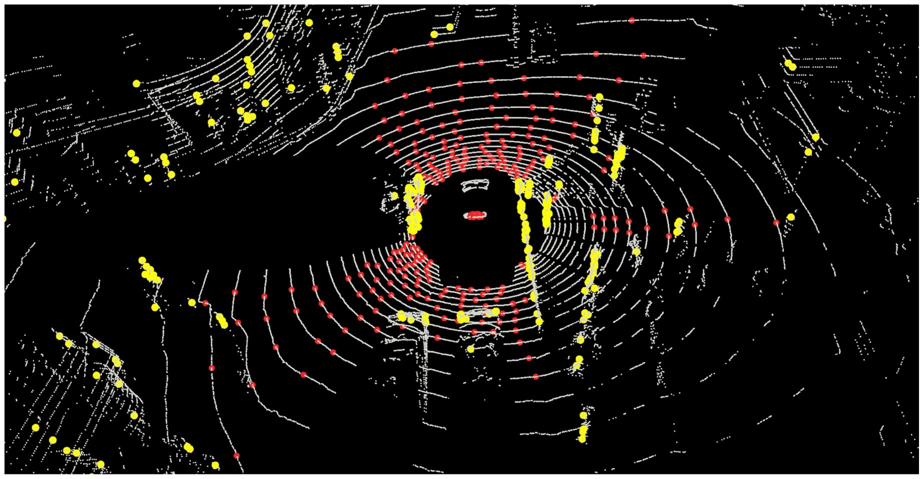


Figure 3.3: Illustration of extracted edge (yellow) and planar points (red) from a LiDAR point cloud frame (grey) in a road intersection, used for scan-to-map matching in LOAM.

through $\mathbf{p}^e_{k,j}$ and $\mathbf{p}^e_{k,l} \in \hat{\mathcal{P}}_k$, representing the corresponding edge. The distance $d^e_{k+1,i}$ between the edge point $\mathbf{p}^e_{k+1,i}$ and the fitted line is the residual of the edge feature, which is minimized and can be expressed as:

$$d^e_{k+1,i} = \frac{\left|\left(\mathbf{p}^e_{k+1,i} - \hat{\mathbf{p}}^e_{k,j}\right) \times \left(\mathbf{p}^e_{k+1,i} - \hat{\mathbf{p}}^e_{k,l}\right)\right|}{\left|\hat{\mathbf{p}}^e_{k,j} - \hat{\mathbf{p}}^e_{(k,l)}\right|} \tag{3.21}$$

Similarly, for each plane point $\mathbf{p}^p_{k+1,i}$ in $\mathcal{P}_{k+1}$, the distance $d^p_{k+1,i}$ between the point and the fitted plane in $\hat{\mathcal{P}}_k$, is the residual of the plane feature to be minimized, which can be represented as:

$$d^p_{k+1,i} = \frac{\left|\left(\mathbf{p}^p_{k+1,i} - \hat{\mathbf{p}}^p_{k,j}\right) \cdot \left(\hat{\mathbf{p}}^p_{k,j} - \hat{\mathbf{p}}^p_{k,l}\right) \times \left(\hat{\mathbf{p}}^p_{k,j} - \hat{\mathbf{p}}^p_{k,m}\right)\right|}{\left|\left(\hat{\mathbf{p}}^p_{k,j} - \hat{\mathbf{p}}^p_{k,l}\right) \times \left(\hat{\mathbf{p}}^p_{k,j} - \hat{\mathbf{p}}^p_{k,m}\right)\right|} \tag{3.22}$$

where $\hat{\mathbf{p}}^p_{k,j}$, $\hat{\mathbf{p}}^p_{k,l}$, $\hat{\mathbf{p}}^p_{k,m}$ are three nearest points of $\mathbf{p}^p_{k+1,i}$ among planar points in $\hat{\mathcal{P}}_k$ using kd-tree search.

A geometric relationship between edge and plane feature point in $\mathcal{P}_{k+1}$ and the corresponding in $\hat{\mathcal{P}}_k$ can be estimated as

$$f\left(\mathbf{p}^{e}_{k+1,i}, \mathbf{T}_{k+1}\right) = d^{e}_{k+1,i} \tag{3.23}$$

$$f\left(\mathbf{p}^{p}_{k+1,i}, \mathbf{T}_{k+1}\right) = d^{p}_{k+1,i} \tag{3.24}$$

Therefore, the transformation $\mathbf{T}^{k}_{k+1}$ can be calculated by minimizing the cost function based on Equations 3.23 and 3.24,

$$\mathbf{T}^{k}_{k+1} = \arg\min \frac{1}{2}\left\{\sum_{i=1}^{N^{\text{edge}}} \left\|\omega_{k+1,i} f\left(\mathbf{p}^{e}_{k+1,i}, \mathbf{T}_{k+1}\right)\right\|^2 + \sum_{i=1}^{N^{\text{planar}}} \left\|\omega_{k+1,i} f\left(\mathbf{p}^{p}_{k+1,i}, \mathbf{T}_{k+1}\right)\right\|^2\right\} \tag{3.25}$$

N^{edge} and N^{planar} are the numbers of edges and planar points obtained in $\mathcal{P}_{k+1}$. Smaller weight $\omega_{k+1,i}$ is assigned to feature points with a large residual. The optimization can be solved using the Levenberg-Marquardt [83] method by minimizing the distance of the feature.

Lidar Mapping

The mapping algorithm matches the $\mathcal{P}_{k+1}$ and the point cloud map $\mathbf{M}_k$ to mitigate the error estimation arising from lidar odometry. Let $\mathbf{T}_{k,k+1}$ be the transformation of lidar odometry between k and $k+1$, the initial guess $\mathbf{T}^{W}_{k+1}$ can be represented as:

$$\mathbf{T}^{W}_{k+1} = \mathbf{T}^{W}_{k}\mathbf{T}_{k,k+1} \tag{3.26}$$

Then $\mathbf{T}^{W}_{k+1}$ can be used to transform $\mathcal{P}_{k+1}$ into world coordinates, denoted as $\mathcal{P}^{W}_{k+1}$. Similar to the process of lidar odometry described earlier, feature-to-feature correspondence can be determined by conducting a k-d tree search between $\mathcal{P}^{W}_{k+1}$ and $\mathbf{M}_k$. The $\mathbf{T}^{W}_{k+1}$ is then optimized by minimizing the residuals, as outlined in Equations

3.21, 3.22, and 3.25. Later the $\mathbf{T}_{k+1}^{W}$ is incorporated into $\mathbf{M}_k$ to generate $\mathbf{M}_{k+1}$ for the next scan-to-map iteration.

Compared with the point-wise theoretically, the curvature is exploited by LOAM for the edges and planar points classification based on Equations 3.19 and 3.20. Then the feature points are utilized to compute the inter-frame motion by minimizing the residuals in Equations 3.21 and 3.22. Finally, precise state estimation can be obtained with the scan-to-mapping strategy based on the approximate initial guess using Equation 3.26. Compared with the traditional point-wise LiDAR odometry pipelines, the LOAM extracts features that can significantly decrease the number of points involved in the data association. The number of correspondences involved in the optimization is significantly reduced when compared with the point-wise pipeline, leading to improved accuracy and efficiency.

3.2.6 LeGO-LOAM

LeGO-LOAM [39] is a lightweight, ground-optimized LOAM. First, *LeGO-LOAM* projects a point cloud onto a range image with a resolution determined by the horizontal and vertical angular resolution of the LiDAR scanner. Each valid point in the range image is assigned a pixel value based on the Euclidean distance from the point to the sensor. Segmentation is then applied to the range image to classify ground and large objects. In the odometry module, $[t_z, \theta_{\text{roll}}, \theta_{\text{pitch}}]$ is obtained by finding correspondences of planar features from ground points, while $[t_x, t_y, \theta_{\text{yaw}}]$ is estimated by matching edge features from segmented clusters.

$$[t_z, R_y, R_z] = \arg\min \frac{1}{2} \sum_{i=1}^{N^{pg}} \left\| \omega_{k+1,i} f\left(\mathbf{p}_{k+1,i}^{p}, \mathbf{T}_{k+1}\right) \right\|^2 \tag{3.27}$$

$$[t_x, t_y, R_z] = \arg\min \frac{1}{2} \sum_{i=1}^{N^{es}} \left\| \omega_{k+1,i} f\left(\mathbf{p}_{k+1,i}^{e}, \mathbf{T}_{k+1}\right) \right\|^2 \tag{3.28}$$

where N^{pg} and N^{es} are the number of planar points within the ground area and edge points from large objects on range images in $\mathcal{P}_{k+1}$. According to the evaluation in [39], the drift in the vertical direction of the *LeGO-LOAM* is significantly smaller than the conventional LOAM since the $[t_z, \theta_{\text{roll}}, \theta_{\text{pitch}}]$ is innovatively estimated by the ground points. We believe this is one of the major contributions of LeGO-LOAM. Moreover, compared with the conventional LOAM, the LeGO-LOAM also performs a loop closure detection [84] to eliminate drift in the lidar mapping process.

3.2.7 Summary

The models of point clouds differ among the various methods mentioned above. An overview of the fundamental differences in the selected LO (Localization and Odometry) methods is shown in Table 2.1. K-d trees and voxelization are commonly used for point modeling and correspondence searching in these LO methods. G-ICP adopts a Gaussian-based point representation, providing advantages in accuracy and robustness compared to ICP, though at the cost of higher computational complexity due to point modeling. NDT, on the other hand, is a computationally efficient LO method, as voxelization is applied for both point modeling and correspondence searching. However, its performance is sensitive to voxel cell size, particularly in diverse environments. VGICP combines the strengths of both G-ICP and NDT, achieving fast and accurate point registration.

Feature-based LO algorithms, such as LOAM and its variants, employ feature extraction before correspondence searching, which reduces computational load and allows real-time execution in most cases. However, because feature-based methods use only a subset of the point cloud, they can suffer from degeneration in environments with limited features. Next, we evaluate both point-based and feature-based LO methods in the context of the challenges presented by urban canyons.

3.3 Performance Evaluation

3.3.1 Experiment Setup

This section does not include the KITTI dataset, as it has already been extensively evaluated in numerous studies, with results summarized in [23]. The performance of the LO methods is assessed using our recently published UrbanNav dataset [85], which contains data collected from various urban environments in Hong Kong and Tokyo. The dataset includes measurements from GNSS, IMU, camera, and LiDAR sensors. Additionally, ground truth data is provided by the NovAtel SPANCPT system, which integrates GNSS RTK with a fiber-optic gyroscope-grade IMU. This section focuses on two competitive datasets from the UrbanNav collection: HK-Data20200314 (Data1) and HK-Data20190428 (Data2). Data2 presents a more challenging environment compared to Data1. Typical scenes from both datasets are shown in Fig. 3.4. The entire dataset is publicly available to the research community for further evaluation and algorithm development.

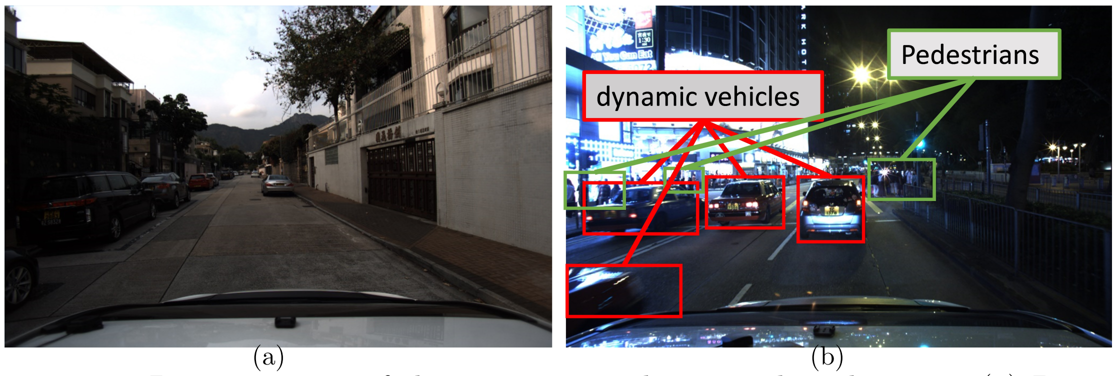


Figure 3.4: Demonstration of the scenarios in the two urban datasets. (a) Data1: Low-rising buildings and multiple right-turning areas. (b) Data2: Variety of dynamic vehicles and numerous high-rising buildings.

We set up Ubuntu 18.04 with ROS Melodic as the platform baseline for the majority of the LO pipelines. However, the original publicly available LOAM implementation is restricted to Ubuntu 16.04 with ROS Kinetic due to software incompatibilities with ROS Melodic. The recommended parameters provided by the original authors were retained across all experiments. We substituted the package configuration for the Velodyne HDL32 sensor. The detailed configuration and the rosbag playback rates are presented in Table 3.1. For FastGICP and Fast LOAM, the playback rate was set to 0.5, as the processing time for some frames was slower than 10 Hz. All experiments were conducted using only the point cloud data published by the sensor. The estimated trajectories from all methods are shown in Fig. 3.5. The effectiveness of the LO methods was evaluated using the widely adopted EVO tools [86], a popular Python package for evaluating and comparing odometry and SLAM algorithms. RPE compares the estimated relative pose with the reference pose in a fixed time interval. An epoch represents a single time step in the sequence of measurements or observations. Given a sequence of poses from the estimated trajectory $\mathbf{T}_{\text{est}} = \{\mathbf{T}_{\text{est},1}, \mathbf{T}_{\text{est},2}, \ldots, \mathbf{T}_{\text{est},N_{\text{epochs}}}\}$ and ground truth $\mathbf{T}_{\text{gt}} = \{\mathbf{T}_{\text{gt},1}, \mathbf{T}_{\text{gt},2}, \ldots, \mathbf{T}_{\text{gt},N_{\text{epochs}}}\}$, N_{epochs} represents the number of states for evaluation. The relative pose error $\text{RPE}_{i,j}$ between timestamp t_i and t_j can be defined as,

$$\text{RPE}_{i,j} = \left(\mathbf{T}_{\text{gt},i}^{-1}\mathbf{T}_{\text{gt},j}\right)^{-1}\left(\mathbf{T}_{\text{est},i}^{-1}\mathbf{T}_{\text{est},j}\right) \tag{3.29}$$

$\text{RPE}_{i,j}$ belongs to the Special Euclidean Group, $SE(3)$.

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{epochs}}}\sum_{i=1}^{N_{\text{epochs}}} \text{RPE}_i^2} \tag{3.30}$$

N_{epochs} represents the number of epochs to be evaluated. The RMSE can be calculated based on the overall RPE.

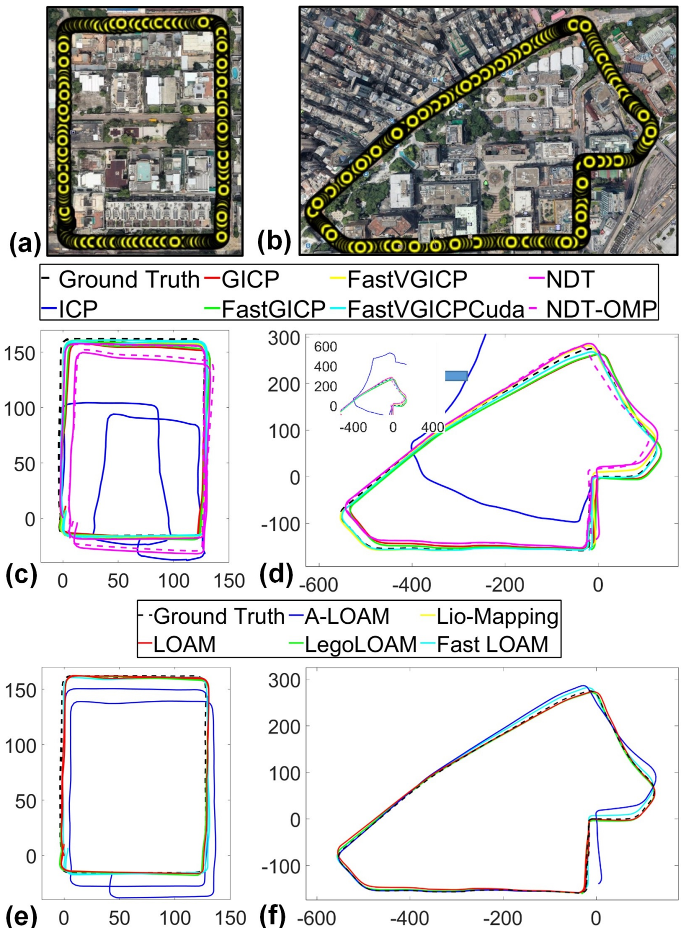


Figure 3.5: (a) Satellite image showing the ground truth for Data1. (b) Satellite image showing the ground truth for Data2. Trajectories in (c) and (d) are evaluated using point-wise LO methods for Data1 and Data2, respectively, while trajectories in (e) and (f) are evaluated using feature-wise LO methods for the same datasets.

3.3.2 Computational Cost

The processing time for per-frame (PTPF) is used to evaluate the computational cost of LO methods, as shown in Table 3.2. PTPF is measured from the moment a new point cloud is received until the LO odometry result is generated. For point registration-based methods, only the processing time for odometry is recorded as PTPF. For LOAM-related methods, the processing time is collected at the end of both the odometry and mapping processes. Specifically, LOAM-related methods evaluate the PTPF for odometry and mapping separately, with LOAM achieving the fastest odometry time at a mean value of 9.41 ms, while LIO-Mapping was the quickest in the mapping process, with a mean value of 105.54 ms. Fast LOAM demonstrated a mean PTPF of 60.78 ms by merging the odometry and mapping processes of conventional LOAM.

Among point registration methods, FastVGICPCuda achieved the fastest processing time, while LOAM excelled in the odometry process. LeGO-LOAM demonstrated the fastest mapping process among LOAM series methods. However, the processing times of FastGICP, FastVGICP, and FastVGICPCuda revealed a potential issue when handling large datasets (marked in red in Table 3.2). The maximum processing speed was significantly faster than both the mean processing speed and the published rate in Table 3.1. This processing delay may result in significant errors in some frames.

3.3.3 Detailed Analysis of Accuracy

Translation error is used as the criterion for accuracy evaluation, as shown in Table 3.3. For the Data1 dataset, G-ICP achieved the most accurate results among all evaluated point-wise registration methods, while LOAM had the best performance across all methods in terms of RMSE. For the Data2 dataset, G-ICP again delivered the best results, while Fast LOAM outperformed standard LOAM.

A-LOAM's performance in Table 3.3 and Fig. 3.6 (a) of Data1 is much worse than other LOAM-related methods, with an error of up to 0.8 meters in Data1. This can be attributed to A-LOAM's lack of outlier removal during the feature extraction process, a step included in the original LOAM [18]. To verify this, we conducted an experiment where the feature extraction process from "LIO-Mapping" was applied to the A-LOAM odometry and mapping processes. The accuracy of LiDAR odometry significantly improved, as shown in Fig. 3.6 (b), where the trajectory almost perfectly overlapped when returning to the starting point.

Based on the theoretical comparisons in the previous section and the experimental results, we argue that three major factors are affecting the performance of LO methods, including (1) the motion difference between two consecutive epochs, which has an impact on the initial guess that project $\mathcal{P}_k$ to $\tilde{\mathcal{P}}_k$; (2) the density of dynamic objects, which impacts the assumption of a static environment and affects geometric correspondences., and (3) the degree of urbanization of the evaluated environment, which explores the accuracy of LOs for those highly urbanized areas that GNSS positioning has degenerated [87].

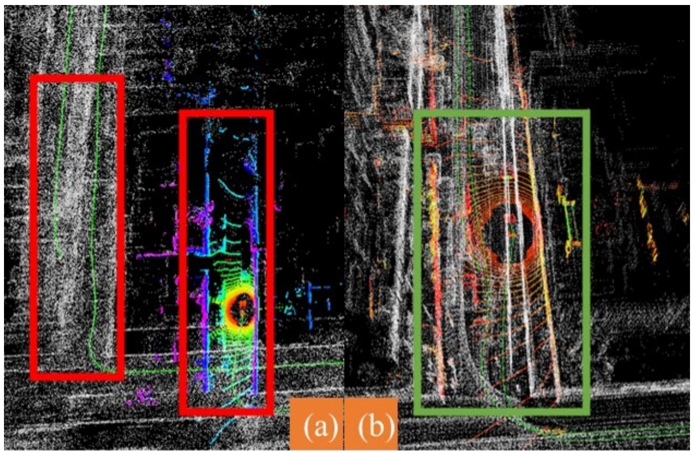


Figure 3.6: (a) The mapping process of A-LOAM, where the repeated route lacks the overlap, is highlighted in red. (b) The feature extraction process of A-LOAM is replaced with that from LIO-mapping, resulting in the pose closely aligning with the previous estimate, indicated in green.

The first factor we examined is the motion difference between the two epochs. To compute the motion difference between the timestamp t_i to t_{i+1}, the Euclidean norm of the vector ξ_{k+1} in Lie algebra $\mathfrak{se}(3)$ is utilized to represent its corresponding rigid body motion $\mathbf{T}^k_{k+1} \in \mathrm{SE}(3)$. The formula is represented as,

$$\Delta d_{i+1} = \left\| \log\left(\mathbf{T}^k_{k+1}\right)^{\vee} \right\|, \quad \mathbf{T}^k_{k+1} \in \mathbb{R}^{4\times4} \tag{3.31}$$

where the superscript $\vee$ denotes the operator that maps elements from SE(3) to their vector representation in $\mathfrak{se}(3)$.

The second factor involves the number of dynamic objects contained in each scan. Dynamic objects, such as cars and buses, were annotated in two datasets using SUSTechPOINTS [88], a semi-automatic labeling tool. The density of dynamic objects is represented as,

$$c = \left(\frac{N_{\text{car}} + N_{\text{bus}}}{N_{\text{total}}} \right) \times 100\% \tag{3.32}$$

which N_{car} and N_{bus} denote the number of LiDAR points corresponding to cars and buses, respectively, in the current scan. N_{total} represents the total number of points in existing epoch.

To evaluate the degree of urbanization of the evaluated scene, our previous work in [87] proposed to adopt the 3D building models to further estimate the skymask (GNSS skyplot with building boundaries). The mean mask elevation angle μ_{MEA} is defined to quantitatively represent the degree of urbanization as follows:

$$\mu_{\text{MEA}} = \frac{\sum_{\alpha=1}^{N} \theta_\alpha}{N} \tag{3.33}$$

where θ_α represents the elevation angle corresponding to the building height at a given azimuth angle α. N refers to the number of evenly spaced azimuth angles from the skymask, typically 360 for a resolution of 1 degree. Locations surrounded by tall buildings will have a high μ_{MEA}, while rural areas will result in a lower μ_{MEA}.

a) Verification using Data1

Data1 - Factor 1 - Motion Difference (MD): As shown in Fig. 3.7 (A), the motion difference increases from 0 to 5.0 m, leading to a translation error of 0.7 m for both G-ICP and LOAM. It is a typical scene of motion changing when a vehicle starts from the roadside. Both methods demonstrated accurate performance between

Fig. 3.7 (A) and (B) as there was a minor change in motion. In Fig. 3.7 (B), as the ego-vehicle slows down before passing an intersection, a significant motion offset is produced. As shown in Fig. 3.7 (C), the error increases again when the vehicle makes a right turn and eventually returns to the starting point, stopping at the end.

Data1 - Factor 2 - Dynamic Objects (DO): Data1 represents a low-traffic area with no other vehicles, except in (D) in Fig. 3.7. The results indicate that LOAM is more sensitive to dynamic vehicles compared to G-ICP in this dataset.

Data1 - Factor 3 - Degree of the Urbanization (Skymask): Fig. 3.7 (E) shows the G-ICP error peaking as the skymask changes. Similarly, in Fig. 3.7 (F), the LOAM error increases when the density of surrounding buildings drastically decreases.

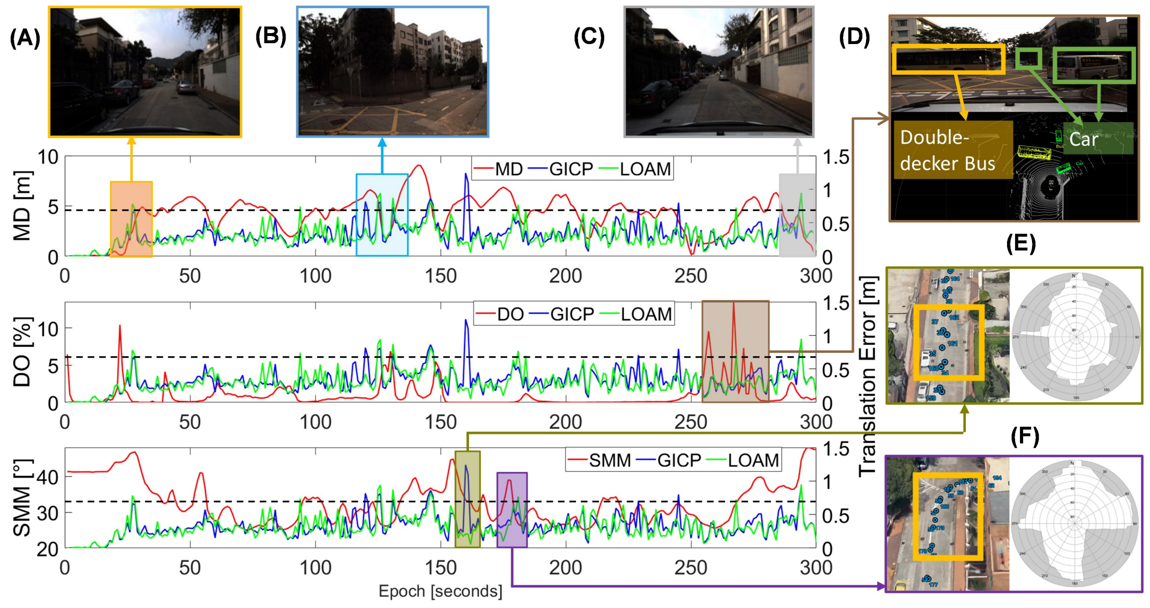


Figure 3.7: The comparison of motion difference (MD), dynamic objects (DO), and sky mask mean (SMM) against translation error in G-ICP and LOAM using Data1. The dash-dot line represents the outlier threshold. (A), (B), and (C) mark typical challenging scenarios for the LO methods.

b) Verification using Data2

Data2 - Factor 1 - Motion Difference (MD): As shown in Fig. 3.8 (A), the translation error exceeded 1.0 m for both GICP and Fast LOAM. This occurred as the ego-vehicle resumed movement when the traffic light turned green, with the motion difference increasing from 0 to 10 meters. Both methods showed superior performance around frame 300, as depicted in Fig. 3.8 (A) and (B), where the vehicle was stationary, and the motion was effectively zero. In Fig. 3.8 (B), the ego-vehicle made a right turn at a busy intersection, and the error increased to 1.5 meters, attributed to both the motion difference and the presence of dynamic vehicles.

Data2 - Factor 2 - Dynamic Objects(DO): Fig. 3.8 (C)(D) depicts scenes of heavy traffic and numerous vehicles, demonstrating that dynamic objects significantly impacted the error.

Data2 - Factor 3 - Degree of the Urbanization (Skymask): As shown in Fig. 3.8 (E)(F), the rapid changes in the skymask led to a loss of accuracy in both LO methods.

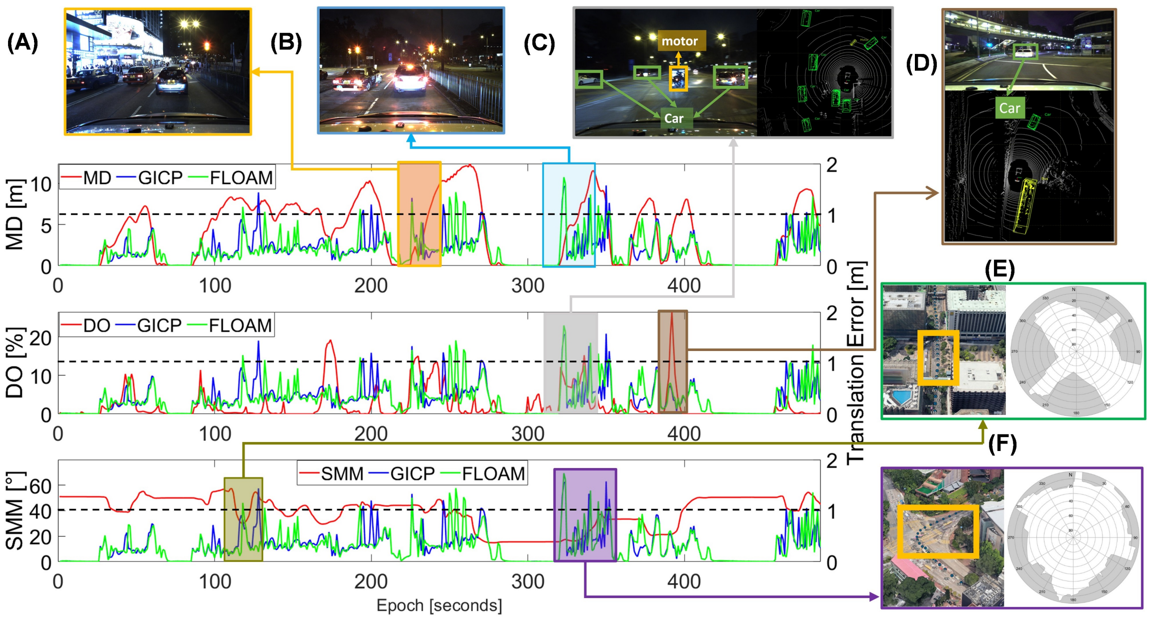


Figure 3.8: The comparison of motion difference (MD), dynamic objects (DO), and sky mask mean (SMM) against translation error in G-ICP and LOAM using Data2. The dash-dot line represents the outlier threshold. (A), (B), and (C) mark typical challenging scenarios for the LO

c) Short discussion

As shown in Fig. 3.7 and Fig. 3.8, MD and translation error exhibit the strongest correlation, as indicated by their synchronized peaks. Therefore, integrating with inertial sensors is good to compensate the initial guess for registration. DO and translation error also show a noticeable correlation, which we will discuss in the following chapters. SMM and translation error demonstrate a moderate correlation, as urban occlusion effects contribute to localization errors.

3.4 Summary

In this chapter, we presented a benchmark comparison and error analysis of publicly available LO methods using two challenging datasets collected in the urban canyons of Hong Kong. Feature-based methods demonstrated both accuracy and cost-efficiency by leveraging extracted feature points. However, the performance of both methods was impacted by three dominant factors. According to our experiments, we suggest that combining both the feature-wise and point-wise methods can be a promising solution, which will be discussed in Chapter 4. The feature-based method can efficiently provide a coarse odometry estimate, which can then be used as an initial guess for point-based point cloud registration. Since point-based methods are highly dependent on the initial guess, this approach results in a robust coarse-to-fine LiDAR odometry pipeline.

Table 3.1: Code and Parameters of the publicly available LO methods using UrbanNav Dataset

Method	**Code Repository**	**Recommended Parameters**	**Rosbag Rate**
ICP [33]	`github.com/koide3/hdl_graph_slam`	Default parameters in `hdl_graph_slam.launch`	0.1
G-ICP [34]	`github.com/koide3/hdl_graph_slam`	Default parameters in `hdl_graph_slam.launch`	0.1
FastGICP [35]	`github.com/SMRT-AIST/fast_gicp`	Integrated into `hdl_graph_slam`. Default parameters are adopted in `hdl_graph_slam.launch`	0.5
FastVGICP [35]	`github.com/SMRT-AIST/fast_gicp`	Default parameters are adopted in `hdl_graph_slam.launch`	1.0
VGICPCuda [35]	`github.com/SMRT-AIST/fast_gicp`	Default parameters are adopted in `hdl_graph_slam.launch`	1.0
NDT [36]	`github.com/koide3/hdl_graph_slam`	`ndt_resolution` was set to 3.0 for outdoor environment in `hdl_graph_slam.launch`	0.1
NDT-OMP [36]	`github.com/koide3/ndt_omp/commit/c799f459b4838dd9c65698573370b16c4e7ce7d9`	Default parameters in `hdl_graph_slam.launch`	1.0
LOAM [18]	`github.com/laboshinl/loam_velodyne`	Changing `VLP-16` to `HDL-32` in `loam_velodyne.launch`	1.0
A-LOAM [38]	`github.com/HKUST-Aerial-Robotics/A-LOAM`	Default parameters are adopted in `aloam_velodyne_HDL_32.launch`	1.0
LeGO-LOAM [39]	`github.com/RobustFieldAutonomyLab/LeGO-LOAM`	Changing `VLP-16` config to `HDL-32E` in `utility.h`	1.0
LIO-Mapping [40]	`github.com/hyye/lio-mapping`	Updating `sensor_type = 32, no_deskew = false` in `64_scans_test.launch` as 32 scans	1.0
Fast LOAM [41]	`github.com/wh200720041/floam`	Changing `scan_line = 64` to `32` in `floam.launch`	0.5

Table 3.2: The processing time for per-frame (PTPF) evaluation results of Data1 and Data2. Top performance of the registration-based methods is highlighted with **bold**, as well as feature-based methods marked with blue. Red fonts denote the maximum PTPF values obviously shifted compared to the mean values.

Dataset	**Method**	**Odometry PTPF (ms)**			**Mapping PTPF (ms)**		
		Max	**Min**	**Mean**	**Max**	**Min**	**Mean**
Data1	ICP	1156.76	29.99	91.47	N/A		
	G-ICP	1476.64	92.79	232.89	N/A		
	FastGICP	375.21	24.50	75.02	N/A		
	FastVGICP	786.19	22.39	42.23	N/A		
	FastVGICPCuda	1182.24	**12.92**	**29.00**	N/A		
	NDT	1681.75	156.59	472.97	N/A		
	NDT-OMP	**301.57**	9.35	51.67	N/A		
	LOAM	49.04	**3.98**	**9.41**	345.32	**16.32**	138.81
	A-LOAM	40.74	10.73	15.93	612.79	61.27	198.51
	LeGO-LOAM	**31.58**	4.64	9.73	**252.27**	30.32	122.93
	LIO-Mapping	79.88	7.88	25.22	274.58	28.94	**105.54**
	Fast LOAM	N/A			124.02	22.85	60.78
Data2	ICP	988.06	34.42	105.38	N/A		
	G-ICP	520.10	106.35	219.64	N/A		
	FastGICP	1465.45	21.69	87.14	N/A		
	FastVGICP	339.66	25.77	47.16	N/A		
	FastVGICPCuda	1626.14	**16.03**	**38.08**	N/A		
	NDT	1442.36	195.41	488.43	N/A		
	NDT-OMP	**142.85**	13.87	39.18	N/A		
	LOAM	56.85	4.7	**10.73**	322.08	**17.50**	125.80
	A-LOAM	38.06	11.96	15.2	588.59	76.02	209.23
	LeGO-LOAM	**33.70**	**3.35**	11.16	**248.47**	26.49	**115.19**
	LIO-Mapping	86.57	9.13	28.16	328.82	37.40	119.38
	Fast LOAM	N/A			164.99	27.23	90.52

Table 3.3: Evaluation results on the two urban datasets. The best performance among registration-based methods in terms of accuracy is highlighted in bold, while the best feature-based results are marked in blue.

Dataset	**Description**	**Trajectory Length**	**Method**	**Relative Translation Error (m)**		**Relative Rotation Error (deg)**	
				RMSE	**Mean**	**RMSE**	**Mean**
Data1	Low-urbanization, Small Loop	1.21 km	ICP	1.857	1.529	2.073	1.533
			G-ICP	**0.371**	0.326	1.912	1.268
			FastGICP	0.383	0.330	1.814	1.238
			FastVGICP	0.372	**0.321**	**1.670**	**1.113**
			FastVGICP-Cuda	0.627	0.389	1.773	1.184
			NDT	0.405	0.323	1.938	1.301
			NDT-OMP	0.510	0.397	1.840	1.197
			LOAM	**0.354**	**0.311**	2.113	1.379
			A-LOAM	0.803	0.476	1.870	1.232
			LeGO-LOAM	0.374	0.324	1.972	1.236
			LIO-Mapping	0.479	0.337	**1.201**	**0.803**
			Fast LOAM	0.376	0.322	1.661	1.094
Data2	Heavy Traffic, Tall Buildings	2.01 km	ICP	1.684	1.213	1.494	0.902
			G-ICP	**0.417**	**0.296**	1.129	0.662
			FastGICP	0.543	0.301	1.303	**0.472**
			FastVGICP	0.711	0.367	**1.093**	0.618
			FastVGICP-Cuda	0.874	0.411	1.734	0.732
			NDT	0.816	0.422	1.106	0.657
			NDT-OMP	0.788	0.388	1.149	0.661
			LOAM	0.450	0.321	1.383	0.799
			A-LOAM	0.478	0.331	1.234	0.695
			LeGO-LOAM	0.462	0.333	1.226	0.694
			LIO-Mapping	0.664	0.379	**0.886**	**0.471**
			Fast LOAM	**0.423**	**0.294**	1.141	0.619

Chapter 4

Dynamic Object-aware LiDAR Odometry in Urban Areas

4.1 Introduction

As highlighted in the benchmark comparison of publicly available LO in Chapter 3, existing LO methods still struggle to provide satisfactory positioning accuracy in urban areas due to three key factors: ego-vehicle dynamics, dynamic objects, and degree of urbanization. Among these, dynamic objects are particularly challenging and must be effectively managed to enhance positioning performance.

In this chapter, we develop a coarse-to-fine LiDAR-based method that incorporates dynamic object removal. This approach utilizes both instance-level DNN and point-wise discrepancy images to accurately identify and mitigate dynamic points. Furthermore, we propose a dynamic object-aware LiDAR odometry method that adaptively assigns weightings to dynamic features, improving accuracy by reducing the impact of these transient elements.

4.2 A Coarse-to-Fine LiDAR-Based SLAM with Dynamic Object Removal in Dense Urban Areas

To mitigate the impacts of the errors caused by dynamic objects, numerous studies [89; 90] have been presented to address localization and mapping problems in high-dynamic environments. A random sampling consensus (RANSAC) method [91] was proposed to eliminate mismatches by treating moving objects as outliers. However, its performance significantly degrades when dealing with a large number of dynamic points. The clustering method [89] divides the point cloud into organized groups, efficiently extracting dynamic vehicles based on classification modeling. Nevertheless, parameter-based approaches are vulnerable to unknown object classes or threshold limitations.

In recent years, deep learning-based methods have been widely developed to address the effects of moving objects, achieving impressive results on the KITTI dataset [23]. LO-Net [54] was introduced to tackle dynamic objects using a mask-weighted geometric constraint loss, achieving similar results to LOAM. Recent work [92] improves SLAM accuracy by predicting point-wise semantic labels using RangeNet++ [93] with range images. For broader applicability, an unsupervised dynamic awareness LO method [94] was proposed by a team from ETH, where dynamic objects are automatically labeled using an occupancy grid-based approach. DUFOMap [95] is a dynamic awareness mapping framework developed for efficient online processing, maintaining robust performance across diverse scenarios with consistent parameter settings. However, detecting and estimating motion in highly urbanized environments, such as Hong Kong, remains a challenging research problem.

Change detection [96] is another effective approach for detecting objects by comparing the current scan with a pre-generated map based on prior pose estimation.

Yoon et al. [97] proposed a ray-tracing method with a false-positive filter to improve detection accuracy. Another study [98] introduced a voxel ray-casting-based method to build a static map, but this process is time-consuming as it requires traversing the voxel grid. To reduce computational load, a range image-based visibility check was proposed in [99] to directly remove dynamic points from the map. However, this method often includes static areas in its detections due to the field of view (FOV) limitations.

The objective of this work is to provide a pipeline to optimize LiDAR-based SLAM performance by detecting and removing dynamic objects. Our approach leverages the capabilities of DNN and point-wise discrepancy comparison. First, a custom-trained DNN [100] is employed to obtain precise feature representations and classifications in highly urbanized areas, as illustrated in Fig.4.1 (b). Second, an existing LO method [18] is used to generate coarse poses from scans with dynamic object removal. These poses are then used to construct a submap, which helps further refine the detection of dynamic objects by comparing range image-based discrepancies between the scan and the submap. The initial odometry guess is provided during this coarse process, and the refined scans are subsequently processed by the LiDAR odometry to produce more accurate poses. This coarse-to-fine approach improves relative translation accuracy by 19.1% through the filtering of dynamic objects. As a result, the generated point cloud map contains significantly fewer non-static points, as shown in Fig. 4.1 (d). The key contributions of this work are summarized as follows:

(1) We presented a Coarse-to-Fine LiDAR-based pipeline that integrates dynamic object removal and improves odometry performance in dense urban environments.

(2) We constructed a more accurate point cloud map to better represent the real-

world conditions in urban canyons.

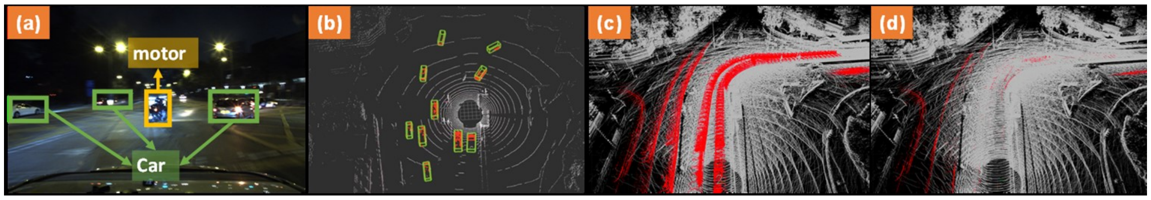


Figure 4.1: (a) Numerous dynamic objects at a crossing road; (b) Example of object detection using a custom DNN model; (c) The raw point cloud map generated by LiDAR SLAM, with static points marked in greyscale and dynamic points labeled in red; (d) The refined point cloud map produced using the proposed method.

4.2.1 Method

We present a coarse-to-fine LO approach with dynamic object removal. First, a 3D DNN is trained offline to support object detection in urban canyons. Removing dynamic objects from the point clouds allows for more precise odometry. A range image-based scan-to-submap process is then performed to further refine the point clouds. Finally, the refined point cloud map and accurate poses are generated through LOAM in the fine stage, utilizing the point-wise labeled point clouds and the initial guess from the coarse stage. The complete pipeline is illustrated in Fig. 4.2.

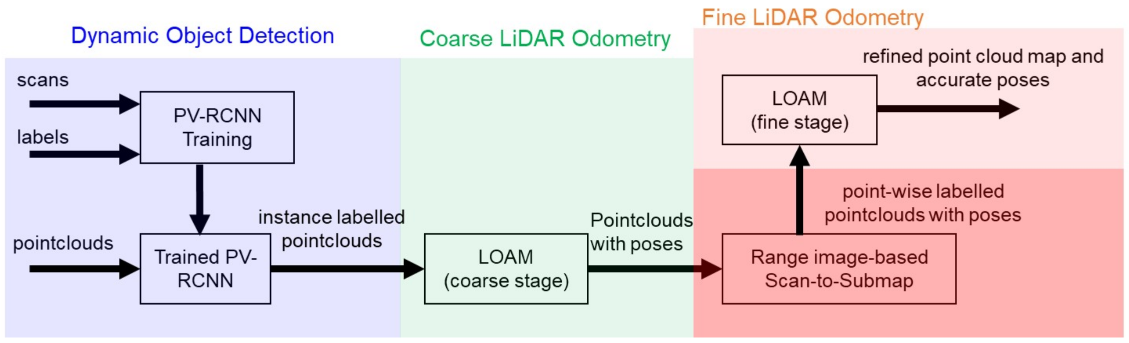


Figure 4.2: Overview of the Proposed Pipeline for coarse-to-fine LiDAR-Base SLAM with dynamic object removal.The LiDAR Odometry with coarse process will be evaluated as LOAM-C while the LiDAR odometry with coarse and fine process will be evaluation as LOAM-CF in the evaluation section.

PV-RCNN Training

PV-RCNN, proposed by Shi et al. [100], combines both point-based and voxel-based convolutional networks for feature learning. Specifically, raw point cloud data is first voxelized and downsampled by 3D voxel CNNs at multiple scales (1x, 2x, 4x, 8x) to capture multi-scale semantic features and generate 3D object proposals. The learned voxel-wise feature volumes at various neural layers are then condensed into a small set of key points through the novel voxel set abstraction module. These keypoint features are subsequently aggregated to the RoI-grid points to learn proposal-specific features for fine-grained proposal refinement and confidence prediction. Experimental results demonstrate that this method achieved superior performance compared to other approaches in the KITTI 3D object detection challenge. Let $\mathbf{F}_k$ be a set of 3D bounding boxes detected at timestamp k from point cloud $\mathcal{P}_k$. The object points $\mathbf{p}_{k,i}^{\text{obj}}$ are extracted within the 3D boxes, while the remaining points are classified as clean points $\mathbf{p}_{k,i}^{\text{clean}}$.

$$\mathbf{p}_{k,i} = \begin{cases} \mathbf{p}_{k,i}^{obj}, & \text{if } \mathbf{p}_{k,i} \text{ is inside any of } \mathbf{F}_k \\ \mathbf{p}_{k,i}^{clean}, & \text{classified as clean if outside the boxes} \end{cases} \tag{4.1}$$

To implement PV-RCNN in customized datasets, we annotated 3D dynamic objects with SUSTechPoints [88]. The FOV is fine-tuned to support 360 degrees. After modifying the data type to suit the network, all data frames are classified into training, validation, and testing datasets and trained the network.

LOAM

LOAM [18] was introduced by the team from Carnegie Mellon University in 2014. For the details of feature extraction and scan matching can refer to Chapter 3.2.5.

Range imaged-based Scan-to-Submap

Inspired by the study in [99], dynamic objects are refined by calculating the difference between the range image of the current scan and a submap within a 50-meter radius. The current scan $\mathcal{P}_k$ and the surrounding map $\mathcal{P}_k^M$ are projected into fix-size range image, $\mathbf{I}_k$ and $\mathbf{I}_k^M$, respectively. For a LiDAR sensor with a 40° vertical FOV and 360° horizontal FOV, the resulting range image has dimensions of 360x40 pixels, where each pixel corresponds to 1 degree in both horizontal and vertical FOV, as illustrated on the left side of Fig. 4.3. Then the visibility check of the map points is calculated via pixel-wise subtraction,

$$\mathbf{I}_k^{Diff} = \mathbf{I}_k - \mathbf{I}_k^M \tag{4.2}$$

We assign a point as dynamic on the map if its corresponding pixel value of $\mathbf{I}_k^{Diff}$is larger than a certain threshold. Finally, we can search the corresponding position of the dynamic points in the current scan to further refine the object points. However, the method might contain several false positive points like trees and ground which cannot visible correctly by the current scan. Such that we apply a clustering and drivable area validation to filter the actual static points, an example of the refined object points is shown on the right of Fig. 4.3.

4.2.2 Performance Evaluation

The performance of the proposed method is evaluated using our UrbanNav dataset [85], which includes data collected from urban environments in Hong Kong and Tokyo. The dataset comprises measurements from GNSS, IMU, camera, and LiDAR sensors. Additionally, ground truth data is recorded using the NovAtel SPAN-CPT system, which integrates GNSS RTK with a fiber-optic gyroscope-level IMU for precise positioning.

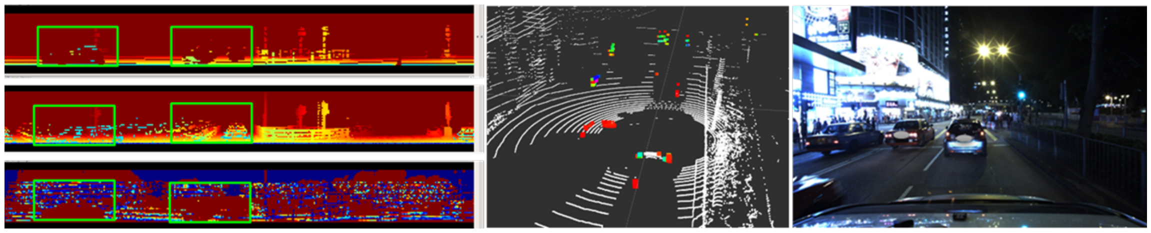


Figure 4.3: Left: The range image, where each row represents $\mathbf{I}_k$, $\mathbf{I}_k^M$, and $\mathbf{I}_k^{\text{Diff}}$, respectively. In the color map, blue indicates closer distances, while red represents farther distances. Therefore, the red pixels in the bottom range image $\mathbf{I}_k^{Diff}$ highlight a high discrepancy between the scan and submap, classifying them as dynamic points. Right: The detected dynamic objects alongside the captured image of the corresponding scenario.

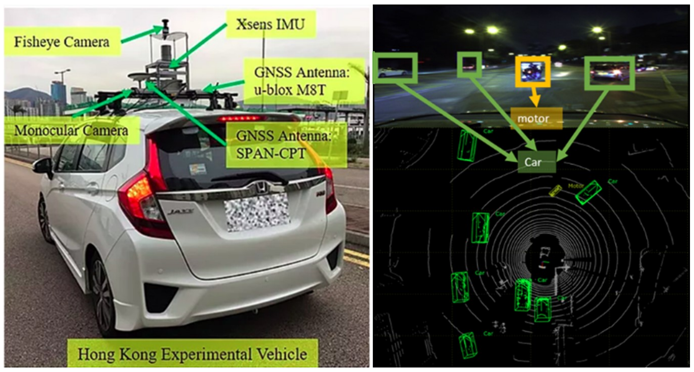


Figure 4.4: Left: The sensors and vehicles used for data collection; Right: A variety of dynamic vehicles captured in the Hong Kong dataset.

PV-RCNN Object Training and Verification Results

We achieve an IoU of 79.02% for the validation data in terms of dynamic object removal of dataset HK-Data20190428. In this experiment, 487 annotated frames are separated into 292 frames are utilized as the training set, 97 frames are performed as the validation set and testing set, respectively. LiDAR scans and ground truth labels of the training set are taken as the input to the network for training. The evaluation results are shown in Table 4.1.

IoU generally means the overlapping level of prediction and ground truth bounding boxes.

$$IoU = \frac{A_{intersection}}{A_{union}} \tag{4.3}$$

$A_{intersection}$is the area of intersection between the predicted box and ground truth box; A_{union} is the area of all two boxes associated.

Table 4.1: 3D object detection results in the validation set.

Type	**AP_0.5**
Bev	79.02%
3D	67.35%

In Table 4.1, we present two different types of Average Precision. The bird's-eye view (Bev) is calculated based on 2D overhead projections, which loses precision in the Z-axis. In contrast, the 3D Average Precision is computed in 3D space, offering a more comprehensive evaluation of performance. Most ground truth objects are successfully predicted with satisfactory accuracy. Examples of detections can be seen in Figs. 4.5 and 4.1 (b). However, currently, only the car class is predicted due to limitations in training labels and available data

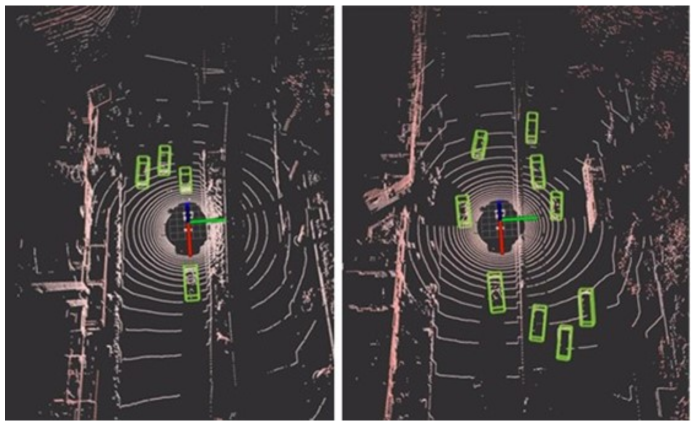


Figure 4.5: Detection of the dynamic object.

LiDAR Odometry

To verify the performance of each component and the entire process, we separate the evaluation based on LOAM, LOAM-C, and LOAM-CF respectively.

- **LOAM**: the original LO
- **LOAM-C**: LO with the coarse process, the DNN is utilized to detect the dynamic objects.
- **LOAM-CF**: LO with the coarse-to-fine process, the scan is further refined by the discrepancy of scan-to-submap.

We labeled the dynamic objects such as cars and buses in the datasets to explore how the dynamic objects affect the position error. The density of dynamic objects factor is defined as,

$$c = \left(\frac{N_{car} + N_{bus}}{N_{total}} \right) \times 100\% \tag{4.4}$$

which N_{car} and N_{bus} represent the number of LiDAR points of cars and buses separately in the current scan. The total number of points in the current frame is denoted by N_{total}. The performance of the listed methods was evaluated using relative pose error (RPE) via the popular EVO tools [86], a widely-used Python package for evaluating and comparing odometry or SLAM algorithms. The overall results are presented in Table 4.2 and Fig. 4.4. Compared to standard LOAM, the proposed pipeline, LOAM-C, achieved a 19.1% reduction in drift, as demonstrated by the RMSE of the translation error. The mean error of LOAM-C decreased from 0.321 m to 0.258 m. Fig. 4.6 shows that LOAM-C can slightly mitigate the error caused by dynamic points. To further evaluate the proposed method in highly dynamic urban canyon scenarios, an epoch-wise evaluation based on the scenarios (A), (B), and (C) in Fig. 4.6 was conducted to demonstrate the performance of LOAM, LOAM-C, and LOAM-CF, respectively.

Qualitative and quantitative epoch-wise results are presented in Fig. 4.7 and Table 4.3. The DNN network may not fully detect all vehicles, so the discrepancy method based on range images is employed to further refine point cloud filtering using DNN labels. In scenarios (A) and (C), LOAM-C outperforms both LOAM and LOAM-CF, while in scenario (B), LOAM-CF achieves more accurate estimations than LOAM-C in terms of translation and rotation errors.

Table 4.2: Positioning results of LOAM and LOAM-C

Dataset	Length	Method	Trans. Error (m)		Rot. Error (deg)	
			RMSE	Mean	RMSE	Mean
HK-Data20190428	1.21 Km	LOAM	0.450	0.321	1.383	0.799
		LOAM-C	0.364	0.258	1.413	0.815

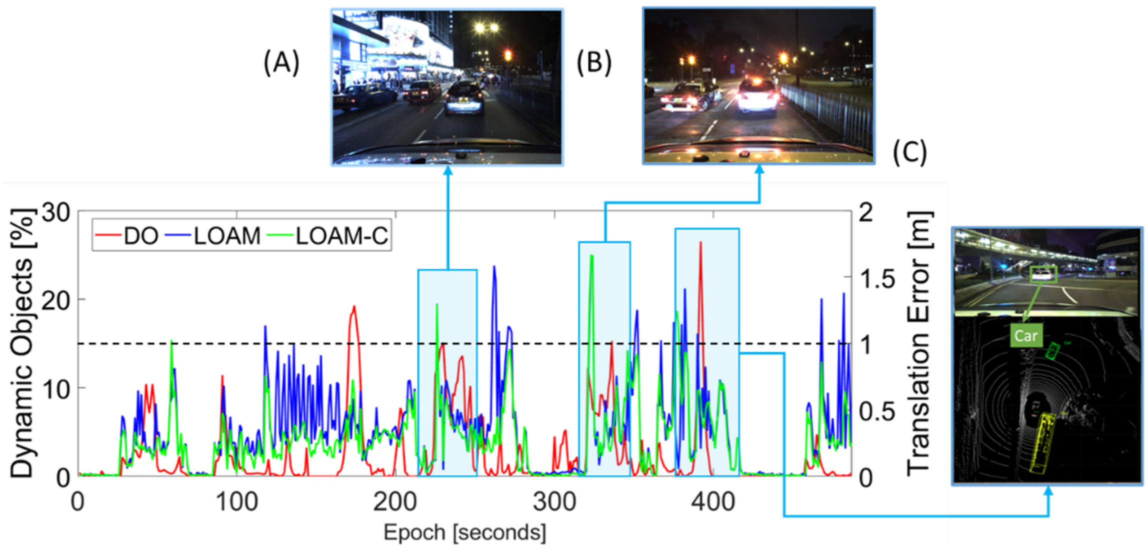


Figure 4.6: The comparison between dynamic objects (DO) and translation error in LOAM and FLOAM. (A), (B), and (C) represent scenarios with numerous dynamic objects where state estimation performance is degraded.

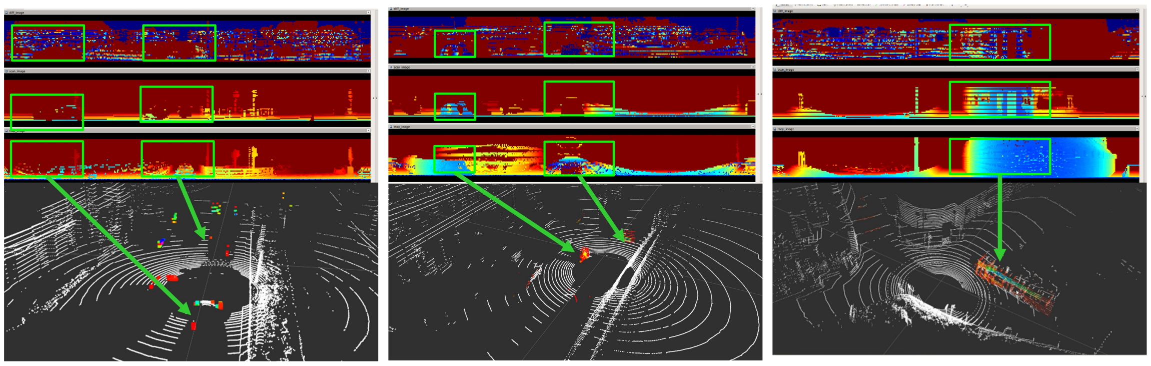


Figure 4.7: Qualitative evaluation of the refined dynamic objects in scenarios (A), (B), and (C) from Fig. 4.6 using the range image-based scan-to-submap method: (A) The ego-vehicle resumes movement as the traffic light turns green from red, with numerous dynamic vehicles and pedestrians nearby; (B) The ego-vehicle stops before the traffic light; (C) The ego-vehicle is close to a double-decker bus.

Table 4.3: Epoch-wise performance comparison of LOAM, LOAM-C, and LOAM-CF under scenarios labeled in Fig. 4.6 Top performance of the method in different scenarios is highlighted with bold.RTE is short for relative pose error while RRE is short for relative rotation error.

Scenario	**Method**	**2D RRE (m)**		**RRE (deg)**	
		RMSE	**Mean**	**RMSE**	**Mean**
A	LOAM	0.415	0.354	0.464	0.374
	LOAM-C	**0.382**	**0.337**	**0.414**	**0.333**
	LOAM-CF	0.387	0.341	0.434	0.352
B	LOAM	0.53	0.406	2.597	1.767
	LOAM-C	0.476	0.357	2.39	1.65
	LOAM-CF	**0.475**	**0.355**	**2.376**	**1.619**
C	LOAM	0.511	0.443	3.242	2.241
	LOAM-C	**0.511**	**0.443**	3.245	2.233
	LOAM-CF	0.517	0.446	**3.224**	**2.225**

Mapping Results

Fig. 4.8 and Fig 4.1 (d) present the refined map with satisfactory results compared to the original map. However, the dynamic pedestrian in Fig. 4.8 is not removed because the pedestrian is not trained in our network. However it could be filtered by the drivable lane. Generally speaking, removing dynamics using the proposed method in the urban canyons are a promising solution for constructing a static map for long-term usage in the autonomous system.

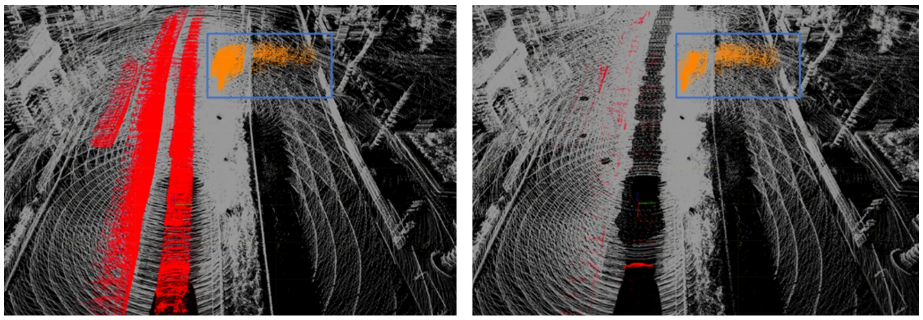


Figure 4.8: Mapping results of Scenario A. Left: The raw point cloud map generated by LiDAR SLAM. Static points are marked in greyscale, while dynamic points on the drivable lane are labeled in red and pedestrians are colored yellow; Right: The refined point cloud map using the proposed method.

4.3 Dynamic Object-aware LiDAR Odometry Aided by Weightings Optimization in Dense Urban Areas

Significant effort has been made to improve LiDAR odometry performance in the presence of dynamic objects [98; 94]. The most common approach involves retaining only static features for odometry estimation, while dynamic features are detected and removed. However, existing methods suffer from two main limitations: (1) In highly dynamic scenarios with numerous moving objects, the static features that remain after excessive dynamic feature removal often provide insufficient constraints for odometry estimation; (2) The DNN or conventional parameter-based methods [101] capable of dynamic object detection is required to remove them so that the odometry is free from dynamic measurements. Therefore, the performance improvement of the odometry suffering from dynamic objects is dependent on accurate detection methods. Unfortunately, there is no effective remedy for false detections which make the odometry even worse. To address these two challenges, this study proposes a dynamic object-aware LiDAR odometry method that effectively utilizes features from both dynamic and static objects, ensuring sufficient constraints even in highly dynamic scenes. Fig. 4.9 demonstrates the contributions of our proposed pipeline, which are summarized as follows:

(1) *Numerical analysis of the impacts of dynamic objects on LiDAR odometry:* This study investigates the impacts of dynamic objects on the degeneration of LiDAR odometry. Three key factors are taken into consideration including the number, the geometric distribution, and the velocity of the dynamic objects. For extensiveness and flexibility, the investigation is performed by adding simulated dynamic objects into real data, which could be custom-tailored by the researchers. The implementation kit is open-sourced at `https://github.com/DarrenWong/code_for_dynaLO` to benefit the research society.

(2) *Adaptive weighting estimation of LiDAR features from dynamic objects:* This study proposes an adaptive weighting-based LiDAR odometry method. Contributions from available features are evaluated comprehensively so that both static and dynamic features can be reasonably utilized. The unhealthy features are automatically assigned with low weights while healthy ones with high weights. The false exclusion of feature subsets caused by the false detection of dynamic objects is thus avoided.

4.3.1 Overview of the Proposed System

An overview of the proposed method is shown in Fig. 4.10. The system takes three main inputs: the LiDAR point cloud, an object database, and object pose information. The output of the system is the pose estimation of the LiDAR odometry. The pipeline consists of two key components: (1) The scene composition, which aggerates $\mathcal{P}_d^{L_k}$ and $\mathcal{P}_s^{L_k}$ with $\mathcal{P}^{L_k}$ to produce $\mathcal{P}_a^{L_k}$. It aims to investigate the impact of dynamic objects on the degradation of LiDAR odometry. Specifically, dynamic

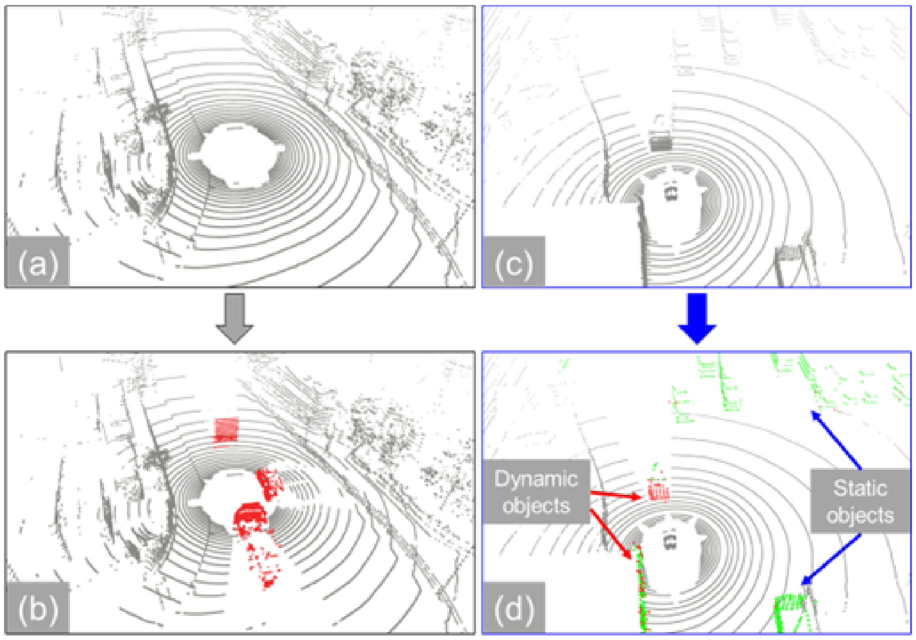


Figure 4.9: Illustration of the contributions of this work. LiDAR points in a 3D static environment are marked in grey. (a) The scanned point cloud in the static environment; (b) The red cloud represents dynamic vehicles generated by the proposed vehicle simulator, integrated with (a) to investigate the impact of dynamic objects on LiDAR odometry degradation; (c) The grey cloud represents the simulated point cloud with dynamic objects; (d) The proposed joint weighting optimization for LiDAR odometry with dynamic object awareness. Both healthy features (green, assigned high weight) and unhealthy features (red, assigned low weight) are simulated based on (b), representing features on static or dynamic objects.

objects from the object database can be integrated with $\mathcal{P}^{L_k}$ in a customizable manner, allowing the user to define their pose and velocity in the LiDAR body frame. Static objects from the object database are anchored in the world frame $\{.\}^W$ by the user and can be transformed into the LiDAR frame $\{.\}^{L_k}$. The transformation matrix $\mathbf{T}_W^{L_k}$ is provided by the NovAtel SPAN-CPT. ; (2) Dynamic object-aware LiDAR odometry (LO) supported by a weighting optimization scheme. Planar features are extracted from $\mathcal{P}_a^{L_k}$ and denoted as $\mathcal{F}_a^{L_k}$. A local map is constructed using planar features from several historical frames [41]. Each feature in the current scan is associated with a planar patch in the map, and the total association cost is optimized to generate accurate pose estimations, similar to the conventional LO methods [41]. Note that the weightings of associations provided by features on dynamic objects are jointly optimized. Large residuals induced by poorly conditioned associations with dynamic objects are down-weighted and minimized iteratively. The superiority of our proposed LO, enhanced by the weighting optimization scheme, is evaluated using the composite scene elaborated in part (1).

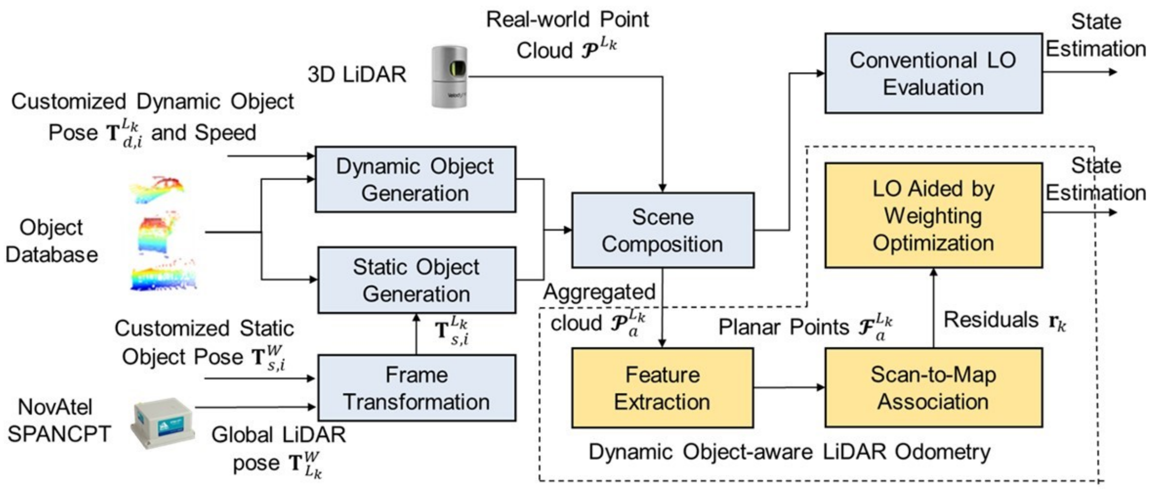

Real-world Point Cloud $\mathcal{P}^{L_k}$
3D LiDAR
Customized Dynamic Object Pose $\mathbf{T}_{d,i}^{L_k}$ and Speed
Dynamic Object Generation
Object Database
Static Object Generation
Customized Static Object Pose $\mathbf{T}_{s,i}^{W}$
$\mathbf{T}_{s,i}^{L_k}$
Frame Transformation
NovAtel SPANCPT
Global LiDAR pose $\mathbf{T}_{L_k}^{W}$
Scene Composition
Conventional LO Evaluation
State Estimation
LO Aided by Weighting Optimization
State Estimation
Aggregated cloud $\mathcal{P}_a^{L_k}$
Planar Points $\mathcal{F}_a^{L_k}$
Residuals $\mathbf{r}_k$
Feature Extraction
Scan-to-Map Association
Dynamic Object-aware LiDAR Odometry

Figure 4.10: Overview of the proposed method.

4.3.2 LiDAR Odometry Degeneracy Induced By Dynamic Objects

This section defines three key factors that represent the impact of dynamic objects on the performance of LiDAR odometry, namely (1) the number of dynamic points that have a direct impact on the data association of LiDAR odometry. Specifically, a higher density of dynamic objects is expected to have a more significant impact on the performance of LiDAR odometry. (2) the geometric distribution of dynamic points, which affects the constraint of the state optimization. In particular, it is expected that the uniformly distributed dynamic objects would lead to a negative impact on the LO. (3) the relative velocity of the dynamic objects, which explores moving dynamic points related to the accuracy of LiDAR odometry. In particular, higher velocities of dynamic objects are likely to lead to greater inconsistencies in feature association, thereby degrading the performance of LiDAR odometry. The listed three factors dominate the degree of degeneracy of the LiDAR odometry induced by the dynamic objects. The next section will introduce the generation of high-dynamic scenarios for further investigation of LiDAR odometry performance.

Number of Dynamic Objects

The number of dynamic objects is expressed as the percentage of dynamic features within the set of features used for LO estimation, defined as follows:

$$D_1 = \frac{N_{d,k}}{N_k} \tag{4.5}$$

in which N_k represents the total number of features used to estimate $\mathbf{T}_{L_k}^{W}$. $N_{d,k}$ indicates the number of features associated with dynamic objects within N_{L_k}. Intuitively, the larger number of dynamic objects induces more incorrect data associations, thus heavier odometry degeneration.

Geometric Distribution of Dynamic Objects

In GNSS, the dilution of precision (DOP) is used to quantify the error in receiver positional measurements that arises from the geometry of the satellites [102]. Inspired by DOP, we represent the covariance of the error in translation estimation induced by the geometric distribution of dynamic objects as follows,

$$\mathbf{Q}_d = \left(\mathbf{J}_{d,t}{}^{T}\mathbf{J}_{d,t}\right)^{-1} = \begin{bmatrix} \sigma_x^2 & \sigma_x\sigma_y & \sigma_x\sigma_z \\ \sigma_y\sigma_x & \sigma_y^2 & \sigma_y\sigma_z \\ \sigma_z\sigma_x & \sigma_z\sigma_y & \sigma_z^2 \end{bmatrix} \tag{4.6}$$

in which $\sigma_{(\cdot)}\sigma_{(\cdot)}$, with $(\cdot)$ representing x, y, or z, denotes the covariance on the specific dimensions of the translation estimation. $\mathbf{J}_{d,t}$ denotes the Jacobian matrix of the association costs produced by the dynamic planar features referring to the translation part t of the estimated pose,

$$\mathbf{J}_{d,t} = [\mathbf{J}_{d1,t}, \mathbf{J}_{d2,t}, \ldots, \mathbf{J}_{dm,t}]^T \in \mathbb{R}^{m\times 3} \tag{4.7}$$

For the i-th planar point in the dynamic features, the Jacobian matrix of residuals can be computed by [41],

$$\mathbf{J}_{di,t} = \frac{\partial \mathbf{r}_{di}}{\partial \mathbf{t}^T} = \mathbf{n}_i^T \in \mathbb{R}^{1\times 3} \tag{4.8}$$

where $\mathbf{r}_{di}$ is the association residual. $\mathbf{n}_i$ represents the normal vector [41] of the planar patch corresponding to the i-th planar feature. As clock bias [102] is not involved in LiDAR odometry, the geometric distribution of the dynamic objects is inherited from the position DOP (PDOP) [102] and represented as follows,

$$D_2 = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \tag{4.9}$$

Specifically, well-distributed dynamic features, such as those evenly distributed, result in ill-conditioned constraints for pose estimation compared to non-uniformly distributed ones. In this case, the covariance of the translation error is expected to increase and a smaller D_2. In other words, a smaller D_2 reflects the more uniform distribution of dynamic features, which significantly exacerbates LiDAR odometry degeneracy.

Relative Velocity of Dynamic Objects

The relative velocity of dynamic objects with respect to the LiDAR is expressed as,

$$D_3 = \frac{\left\| \mathbf{p}_d^{L_k} - \mathbf{p}_d^{L_{k+1}} \right\|}{\Delta t} \tag{4.10}$$

where the $\mathbf{p}_d^{L_k} \in \mathbb{R}^3$ and $\mathbf{p}_d^{L_{k+1}} \in \mathbb{R}^3$ represent the positions of dynamic features in the L_k frame and the L_{k+1} frame, respectively. Δt represents the time interval between the two frames. Within a moderate range, a larger relative velocity of the dynamic objects significantly degrades LiDAR odometry, as it tends to introduce more false associations. However, beyond this range, when the relative velocity is extremely high, the dynamic objects may lag far behind or move far beyond the LiDAR's detection capabilities. In such cases, these dynamic objects are not observed in two consecutive frames, resulting in no degeneracy being induced since no associations are provided for the LiDAR odometry by these dynamic objects.

Dynamic Scene Generation

To simulate a realistic dynamic scene with multiple vehicles, we initially delve into the LiDAR optical model. The raw reflection data from LiDAR can be categorized into several types [103], including solid targets (such as vehicles and pedestrians), soft targets (such as rain, fog, dust, or snow), and noise-related measurements. Our previous work [6] evaluated LiDAR positioning performance in adverse weather conditions. In this study, we aim to examine the impact of dynamic objects on the degradation of LiDAR odometry. Unlike adverse weather and noise factors, dynamic vehicles primarily affect LiDAR through backscattering. Fig. 4.11 depicts the backscattering effect of LiDAR signals interacting with vehicles and buildings.

The pulse propagation in the presence of scattered particles is discussed in [103]. The impulse response of the LiDAR estimates the transformation of the range and intensity between the original point cloud and the modified cloud that includes dynamic objects. Under clear weather conditions, the impulse response function [103] of LiDAR data can be derived as follows:

$$H(R) = \frac{\tau(R)}{R^2} \cdot \rho_0 \delta(R - R_0) \tag{4.11}$$

R is the range of the reflection point while $\tau(R)$ is the ratio of the area illuminated by the transmitter and the area observed by the receiver ($\tau(R)$ normally equals 1 for larger than 2-meter observation). ρ_0 denotes the reflectivity of the target. For the metal component of the vehicle, ρ_0 can be larger than $\frac{10}{\pi}$. $\delta(\cdot)$ is the Dirac delta function and R_0 is the range of the hard target. $\delta(\cdot)$ equals zero everywhere except at $R = R_0$.

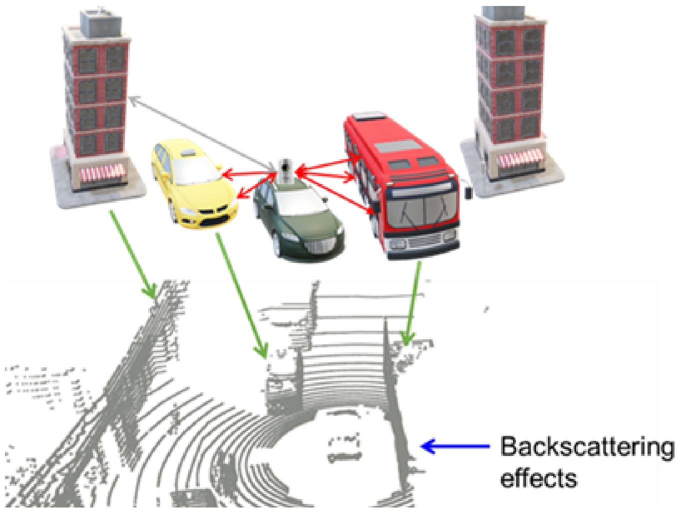


Figure 4.11: Illustration of the effect of hard targets (vehicles and buildings) on LiDAR in a complex dynamic environment (top: graphical representation; bottom: corresponding real point cloud). The red and grey double arrows indicate the backscattering effects from dynamic vehicles and buildings, respectively. The green arrow highlights LiDAR points reflected from vehicles and buildings, while the blue arrow shows the backscattering effect on a bus. A significant portion of LiDAR points strike dynamic objects, resulting in a large number of outliers.

To assess the impact of these factors on LiDAR odometry performance in dynamic scenarios, we utilize real-world dynamic objects and static environments to create realistic scenes based on the LiDAR model. Specifically, we construct a set of dynamic objects using data collected from vehicles and incorporate ground truth information to generate both dynamic and static objects.

We propose that more realistic dynamic scenarios can be created by leveraging dynamic objects from real-world data. To achieve this, various dynamic objects, including cars, trucks, and double-decker buses, were captured in different urban environments using 3D LiDAR. Highly dynamic scenarios are then constructed by transforming the i-th dynamic object $\mathbf{p}_{d,i}^{L_k}$ to the specific locations $\mathbf{T}_{c,i}^{L_k}$ relating to the ego vehicle.

$$\mathcal{P}_{d,i}^{L_k} = \mathbf{T}_{c,i}^{L_k} \mathcal{P}_{d,i} \tag{4.12}$$

$\mathcal{P}_{d,i}^{L_k}$ represents the i-th dynamic object in the LiDAR frame. Fig. 4.12 illustrates the composition of dynamic objects within a static environment (shown in Fig. 4.12 (a)). It is important to note that the points obstructed by the objects in Fig. 4.12 (b) are missing, as the vehicle object acts as a hard target [103] for the LiDAR scanning system, causing the LiDAR pulse to be reflected upon contact with the object. Additionally, a point cloud containing both dynamic and static objects is necessary to evaluate the rejection scheme of dynamic-aware LiDAR odometry. The ground truth information provided by our vehicle platform is utilized to transform the pre-defined static object $\mathbf{p}_{s,i}^{W}$ into the LiDAR local frame $\mathbf{p}_{s,i}^{L_k}$.

$$\mathcal{P}_{s,i}^{L_k} = \left(\mathbf{T}_{k}^{W}\right)^{-1} \mathcal{P}_{s,i}^{W} \tag{4.13}$$

$\mathbf{T}_{k}^{W}$ represents the ego vehicle pose at timestamp k in the world frame. The detail of the dynamic scenario generation is shown in Algorithm 1.

The generated $\mathcal{P}_{a}^{L_k}$ will be used for LiDAR odometry with dynamic object reweighting in the subsequent section. Fig. 4.13 compares the real-world point cloud with the simulated point cloud generated using the proposed LiDAR simulation across two datasets. The simulated point cloud, including dynamic objects, closely matched the real-world data.

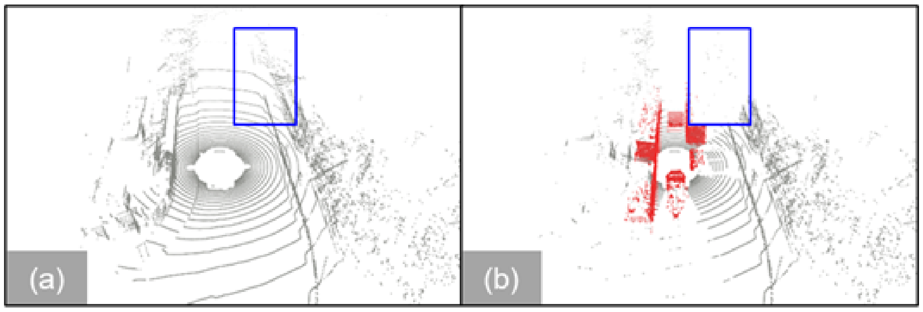


Figure 4.12: Illustration of the simulated LiDAR point cloud. LiDAR points in a 3D static environment are marked in grey, while points corresponding to dynamic objects are labeled in red. (a) The point cloud in a static urban environment. (b) A highly dynamic urban area featuring various real dynamic objects. The blue areas in (a) and (b) indicate the points blocked by the dynamic vehicle, which are not scanable in (b) using the proposed simulator.

4.3.3 LiDAR Odometry Aided By Weighting Optimization

The conventional feature-wise LiDAR odometry [18] can be expressed as,

$$\mathbf{T}_{L_k}^{W} = \arg\min_{\mathbf{T}_{L_k}^{W}} \frac{1}{2} \left\{ \sum_{i=1}^{m} \left\| \mathbf{r}_i^{L_k} \left(\mathbf{T}_{L_k}^{W} \right) \right\|^2 \right\} \tag{4.14}$$

m indicates the number of associated features, while $\mathbf{r}_i^{L_k}$ represents the residuals from edge and planar features. This method performs well in constrained environments with few dynamic objects. However, the conventional formulation is adversely affected by incorrect data associations arising from an excess of dynamic features, leading to significant degradation in pose estimation [94]. To address these challenges, this work introduces a novel term $\rho(*)$, inspired by switchable constraints [105].

$$\rho(\omega_i, \mathbf{r}_{ds,i}^{L_k}) = \omega_i^2 \|\mathbf{r}_{ds,i}^{L_k}\|^2 + k^2(1-\omega_i)^2 \tag{4.15}$$

Algorithm 1: Dynamic scenario generation

```
Inputs: Original LiDAR point cloud at timestamp $k$ as $\mathcal{P}^{L_k}$, objects and their pose parameters
Outputs: The simulated point cloud $\mathcal{P}_a^{L_k}$
Step 1: Initialize $\mathcal{P}_a^{L_k} \leftarrow$ empty.
Step 2: Object creation
    • Step 2-1: Transform the dynamic object points $\mathcal{P}_{d,i}^{c} = \{\mathbf{p}_{d,1}^{c}, \mathbf{p}_{d,2}^{c}, \ldots, \mathbf{p}_{d,n}^{c}\}$ which docked at the origin into the LiDAR local frame $\left\{\mathbf{p}_{obj,1}^{L_k}, \mathbf{p}_{obj,2}^{L_k}, \ldots, \mathbf{p}_{obj,n}^{L_k}\right\}$ using the object transformation.
    • Step 2-2: Transform the static object points $\left\{\mathbf{p}_{s,1}^{W}, \mathbf{p}_{s,2}^{W}, \ldots, \mathbf{p}_{s,n}^{W}\right\}$ into LiDAR local frame $\left\{\mathbf{p}_{s,1}^{L_k}, \mathbf{p}_{s,2}^{L_k}, \ldots, \mathbf{p}_{s,n}^{L_k}\right\}$ using the ego-vehicle ground truth via Equation 4.13.
    • Step 2-3: Filtering the points in the $\mathcal{P}_k^{L_k}$ as $\mathcal{P}_k^{L_k}$ if the line segments between LiDAR origin and the points intersect with any objects based on LiDAR ray casting [104].
    • Step 2-4: Composing the dynamic object points $\mathcal{P}_d^{L_k}$ and static points $\mathcal{P}_s^{L_k}$ over the $\mathcal{P}_k^{L_k}$:
          $\mathcal{P}_a^{L_k} = \mathcal{P}_k^{L_k} + \mathcal{P}_d^{L_k} + \mathcal{P}_s^{L_k}$
Step 3: Finish the algorithm and output the aggregated point cloud as $\mathcal{P}_a^{L_k}$.
```

where $\mathbf{r}_{ds,i}^{L_k}$ represents the residual from both dynamic and static LiDAR object points. $\omega_i \in [0, 1]$ denotes the switchable variable associated with the dynamic residuals, while k stands for the cost parameter for the switch prior constraints term. The switchable constraints can include or exclude the LiDAR features through the associated weighting factor ω_i. The switchable prior constraints are required to anchor the switchable variables at their initialization. All the initial values of the switchable variables are set to 1, which means that all the residuals are accepted as static features in the beginning. Then, the LiDAR point is determined as a static point if ω_i is close to 1 or classified as a dynamic point if ω_i is near 0. Recalling Equations 4.14 and 4.15, the adaptively weighted LiDAR odometry can be written as:

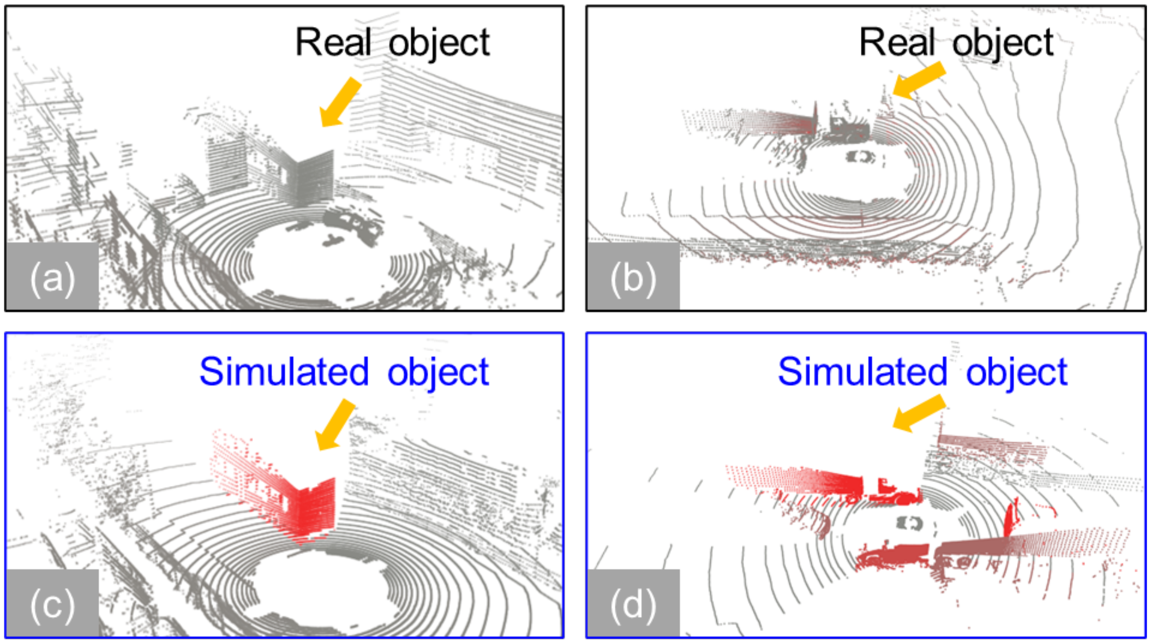


Figure 4.13: Illustration of the simulated point cloud (c) and (d) that contained objects that matched the real data in (a) and (b).

$$\mathbf{T}_{L_k}^{W} = \arg\min_{\mathbf{T}_{L_k}^{W}, \mathcal{W}} \frac{1}{2}\left(\sum_{i=1}^{m} \|\mathbf{r}_{ev,i}^{L_k}(\mathbf{T}_{L_k}^{W})\| + \sum_{i=1}^{N_{ds,k}} \rho(\omega_i, \mathbf{r}_{ds,i}^{L_k})\right) \tag{4.16}$$

where $\mathcal{W}$ denotes the sets of weighting ω_i of each residual. $\mathbf{r}_{\mathrm{ev},i}^{L_k}$ indicates the residual from environmental points $\mathcal{P}_{\mathrm{ev},i}^{L_k}$. $N_{\mathrm{ds},k}$ indicates the number of associated LiDAR points from the simulated objects. The process of the joint weightings estimation is shown in Algorithm 2. To minimize the cost function 4.16, the first-order gradient of $\rho(*)$ relative to ω_i is defined as,

$$\frac{\partial \rho}{\partial \omega_i} = 2\omega_i \|\mathbf{r}_{ds,i}^{L_k}\|^2 - 2k^2(1-\omega_i) \tag{4.17}$$

The optimal ω_i can be achieved where the gradient is zero,

$$\omega_i = \frac{k^2}{\|\mathbf{r}_{ds,i}^{L_k}\|^2 + k^2} \tag{4.18}$$

The optimal $\hat{\rho}$ can be computed by substituting Equation 4.18 into Equation 4.16,

$$\hat{\rho}(\omega_i, \mathbf{r}_{ds,i}^{L_k}) = \frac{k^2 \|\mathbf{r}_{ds,i}^{L_k}\|^2}{k^2 + \|\mathbf{r}_{ds,i}^{L_k}\|^2} \tag{4.19}$$

Algorithm 2: LiDAR odometry aided by Joint Weightings Estimation

```
1: **Inputs:** Aggregated point cloud $\mathcal{P}_a^{L_k}$, environment point cloud $\mathcal{P}_{ev}^{L_k}$, and object point cloud $\mathcal{P}_{ds}^{L_k}$ which aggregated point cloud $\mathcal{P}_k^{L_k} = \{\mathcal{P}_{ev}^{L_k}, \mathcal{P}_{ds}^{L_k}\}$
2: **Outputs:** The optimal pose estimation $\mathbf{T}_{L_k}^W$
3: **Step 1:** Extract features $\mathcal{F}_a^{L_k}$ from $\mathcal{P}_a^{L_k}$
4: **Step 2:** Obtain the residuals $\mathbf{r}_{ev}^{L_k}$ and $\mathbf{r}_{ds}^{L_k}$ from environment features and object features, respectively.
5: **Step 3:** For each $\mathbf{r}_{ev,i}^{L_k}$ in $\mathbf{r}_{ev}^{L_k}$
     • A squared residual function is applied via conventional state estimation Equation 4.14: $\|\mathbf{r}_{ds,i}^{L_k}\|^2$
6: **Step 4:** For each $\mathbf{r}_{ds,i}^{L_k}$ in $\mathbf{r}_{ds}^{L_k}$
     • A switchable constraint is applied [105] using the following cost function:
       $$\rho(\omega_i, \mathbf{r}_{ds,i}^{L_k}) = \omega_i^2 \|\mathbf{r}_{ds,i}^{L_k}\|^2 + k^2(1-\omega_i)^2$$
7: **Step 5:** Minimize the cost function to obtain optimal $\mathbf{T}_{L_k}^W$
       $$\mathbf{T}_{L_k}^W \leftarrow \arg \min_{\mathbf{T}_{L_k}^W, \mathcal{W}} \frac{1}{2} \left( \sum_{i=1}^{m} \|\mathbf{r}_{ev,i}^{L_k}(\mathbf{T}_{L_k}^W)\|^2 + \sum_{i=1}^{N_{ds,k}} \rho(\omega_i, \mathbf{r}_{ds,i}^{L_k}) \right)$$
```

Figs. 4.14 and 4.15 illustrate the adaptive loss corresponding to residual $\mathbf{r}_{ds,i}^{L_k}$ over the range $[-1.0, 1.0]$, with different values of ω_i and k. As depicted in Fig. 4.14, the loss remains a convex function except when $\omega_i = 0$, where the gradient becomes zero. Consequently, the weight assigned to dynamic objects with large residuals can be optimized to approach zero. Fig. 4.15 shows that increasing k raises the loss, indicating a greater influence on Equation 4.16 compared to the residual from static environments $\mathbf{r}_i^{L_k}$. Determining an appropriate value for the prior k is crucial for the reweighting process, and this will be further elaborated in the experimental validation.

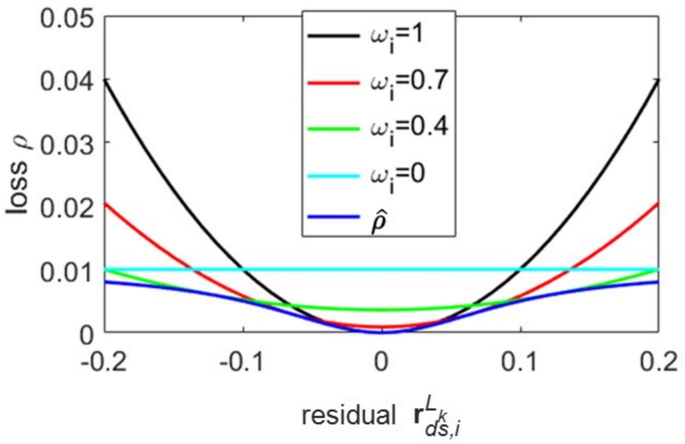


Figure 4.14: $\rho(*)$ with respect to different ω_i if $k = 0.1$. $\hat{\rho}$ represents the optimal loss.

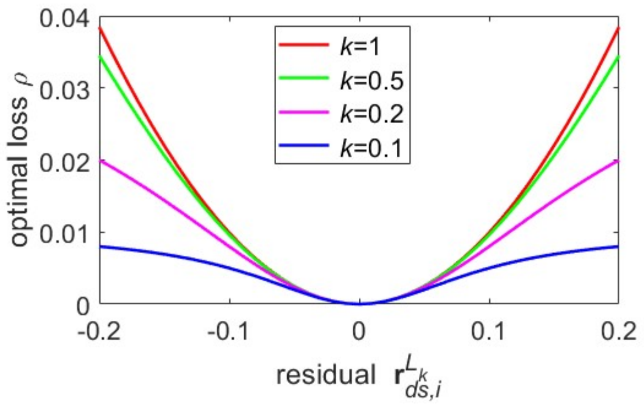


Figure 4.15: The optimal$\hat{\rho}$ with respect to different k.

4.3.4 Performance Evaluation

Experiment Setup

Sensor setups: The performances of the proposed method are evaluated using *UrbanNav* [3] and the *nuScenes* [106] datasets.

Our open-sourced UrbanNav [3] dataset comprises data collected from GNSS, inertial navigation systems (INS), cameras, and LiDAR sensors mounted on a vehicle platform, as illustrated in Fig. 4.16 (a). The NovAtel SPAN-CPT [107] system integrates a fiber optic gyroscope (FOG) with GNSS-RTK technology to deliver

GT positioning data. Additionally, measurements from the SPAN-CPT are tightly coupled using NovAtel's Inertial Explorer [107] software, achieving centimeter-level positioning accuracy. In this experiment, point clouds are captured using the Velodyne HDL-32E [108] at a frequency of 10 Hz, while the LiDAR pose's ground truth is derived from the NovAtel SPAN-CPT at a frequency of 100 Hz. The extrinsic transformation matrix between the two sensors has been pre-calibrated. Synchronization of the timestamps from the two sensor types is accomplished using GPS time and pulse per second (PPS) signals [109].

To enhance the evaluation's diversity, the *nuScenes* dataset [106] was collected in Boston and Singapore under various weather conditions. The sensor configuration for nuScenes is depicted in Fig. 4.17 (a). A 32-line LiDAR system was employed to capture point clouds at a frequency of 20 Hz. The ground truth, with an accuracy of less than 10 centimeters, was achieved through sensor fusion in conjunction with high-definition mapping. Furthermore, the dataset includes 3D bounding boxes and semantic-level annotations for all objects in keyframes, encompassing image, LiDAR, and radar data.

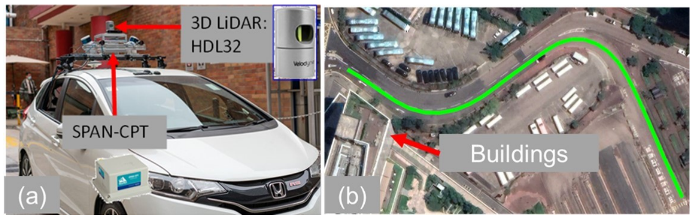


Figure 4.16: (a) Setup for the sensor platform in *UrbanNav*; (b) The evaluated static scene and its ground truth trajectory.

Experimental scenes: To verify the effectiveness of the proposed method, we conducted numerous experiments in typical urban areas in Hong Kong and *nuScenes* datasets. The urban environments are collected in urban areas without dynamic objects, as shown in Fig. 4.16 (b) and Fig. 4.17 (b). We first conducted the experiment in which the static scenes were composed with different percentages of

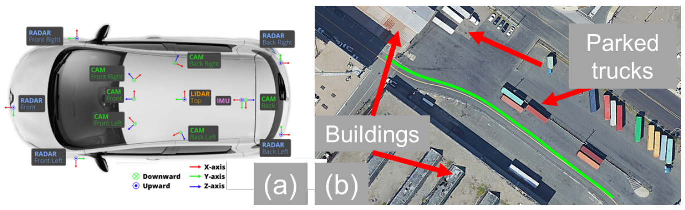


Figure 4.17: (a) Sensor setup for the nuScenes dataset collection; (b) Bird's-eye view of a typical scene in the *nuScenes* dataset, highlighting parked trucks and buildings. The ground truth trajectory of the data collection vehicle is represented by a green curve.

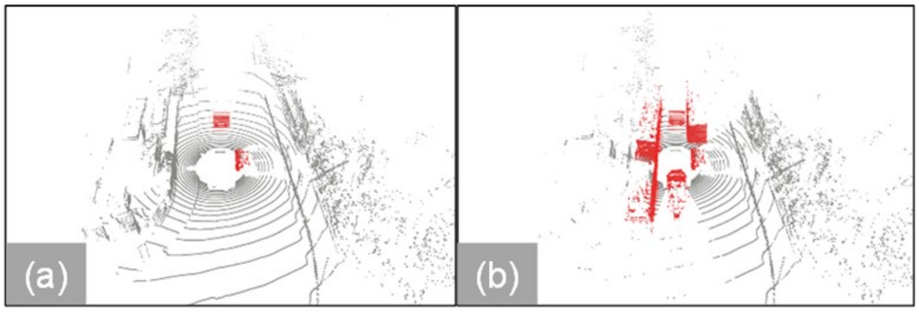


Figure 4.18: Illustration of the simulated point cloud with varying proportions of dynamic objects: (a) 10% dynamic objects, (b) 60% dynamic objects.

dynamic objects (shown in Fig. 4.18). Then, we performed another experiment in which the same amount of dynamic objects was integrated into different directions under the LiDAR body frame (shown in Fig. 4.19). Besides, we performed the third experiment investigating the degeneracy caused by the speed of dynamic objects (shown in Fig. 4.20). Finally, we evaluated the performance of the proposed method using integrated scenes from *UrbanNav* and *nuScenes*, which include both dynamic and static objects, as shown in Fig. 4.21 and Fig. 4.22.

Evaluation metrics: We first analyzed the performance of LiDAR odometry with different levels of dynamic objects. Second, the performance of the proposed method was evaluated via the relative pose error (RPE) to investigate the local consistency of the trajectory with standard practice [49]. To verify the effectiveness of the proposed method in this paper, we evaluate the following method,

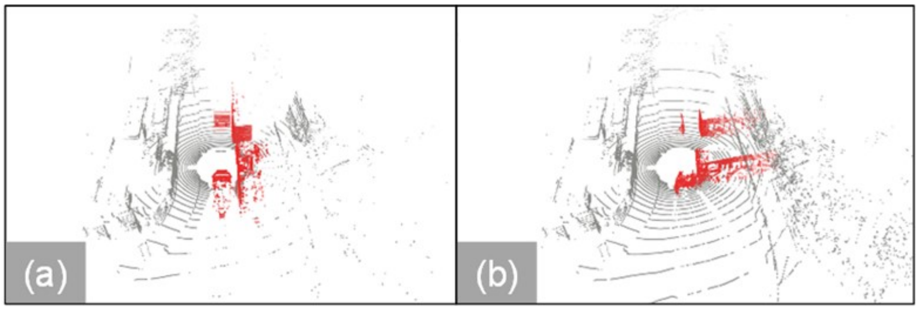


Figure 4.19: Illustration of the simulated point cloud with dynamic points distributed vertically in (a) and horizontally in (b).

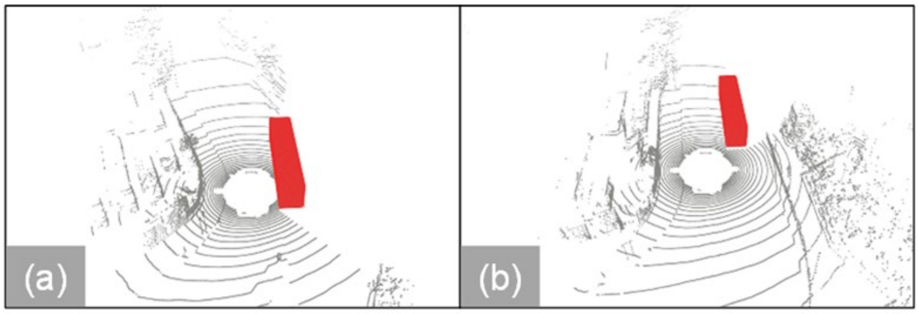


Figure 4.20: Illustration of the simulated point cloud (a) and (b) with dynamic objects at a relative speed of 5m/s.

- **LO**: The conventional LiDAR Odometry.
- **T-LOAM** [110]: The state-of-the-art LiDAR odometry with outlier resistance. The truncated least squares method is adopted to mitigate the effect of dynamic objects.
- **LO-R**: The LiDAR Odometry aided dynamic object removal in the previous section 4.2.
- **LO-RW**: The proposed LiDAR Odometry with dynamic object reweighting. We use $k = 0.1$ for the penalty const parameter in Equation 4.15 experimentally.

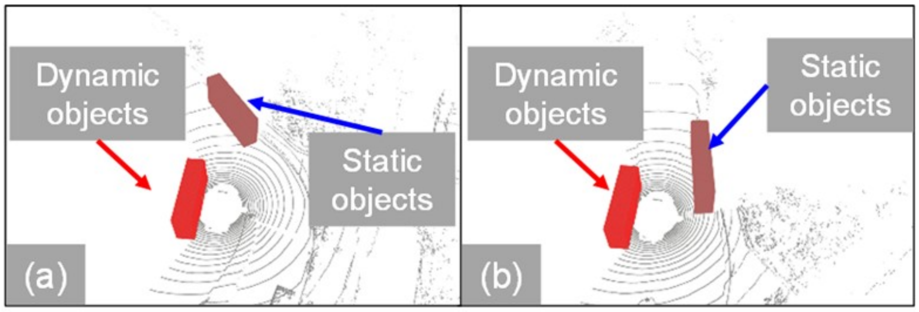


Figure 4.21: Illustration of the integrated point cloud with dynamic objects for evaluation in *UrbanNav*. (a) the scanned point cloud at timestamp t; (b) the scanned point cloud at timestamp $t + 1$.

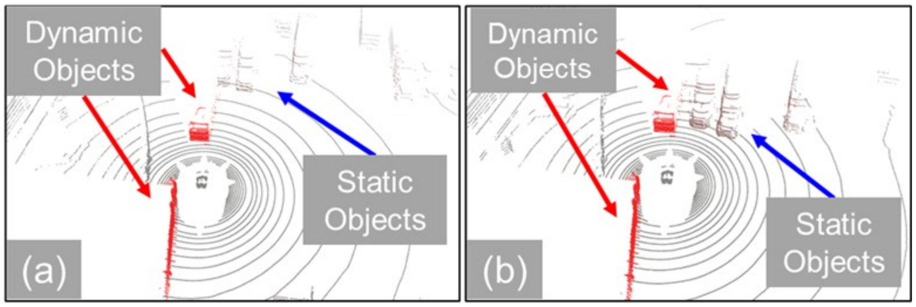


Figure 4.22: Illustration of the integrated point cloud with dynamic objects for evaluation in *nuScenes* (including a parked bus in the data). (a) the scanned point cloud at timestamp t; (b) the scanned point cloud at timestamp t+1

Verification of LiDAR Odometry Degeneracy Induced by Dynamic Objects

1) Verification of the Number of Dynamic Objects

Table 4.4 presents the performance evaluation of LiDAR odometry as influenced by the number of dynamic objects. The first column indicates the varying percentages of dynamic objects based on Equation 4.5. The first row shows the positioning error in static environments (0% dynamic objects), where an RMSE of 0.170 meters was observed with a standard deviation of 0.085 meters. Notably, dynamic objects comprising up to 40% had minimal impact on state estimation, according to our experimental results. However, when the number of dynamic objects increased to 50%, the positioning error rose significantly to 1.330 meters, with a standard deviation of

1.170 meters. This indicates that the number of dynamic objects gradually degrades LiDAR odometry performance. At 80% dynamic objects, the performance worsened dramatically, with the error reaching up to 6.484 meters.

Table 4.4: Performance of LiDAR odometry in terms of numbers of dynamic objects.

Number of Dynamic Objects	**RMSE (m)**	**MEAN (m)**	**STD (m)**	**Max (m)**
0%	0.170	0.147	0.085	0.569
10%	0.171	0.148	0.086	0.573
20%	0.181	0.158	0.088	0.572
30%	0.177	0.154	0.087	0.572
40%	0.191	0.167	0.091	0.566
50%	1.330	0.632	1.170	4.318
60%	2.998	1.682	**2.481**	7.146
70%	6.417	6.394	0.545	7.457
80%	**6.484**	**6.459**	0.566	**7.477**

2) Verification of Geometric Distribution of dynamic objects

Table 4.5 presents the performance of LiDAR odometry under different geometric distributions of dynamic objects. Based on the results from Table 4.4, where significant positioning errors were observed, we selected the scenario with 50% dynamic objects as the baseline for verification. We then redistributed these dynamic objects to create varied configurations (illustrated in Fig. 4.19). The second column in Table 4.5 shows the D_2 values corresponding to the distributions of dynamic objects, calculated using Equation 4.9. The baseline (50% dynamic objects) yielded an RMSE of 1.330 meters with a standard deviation of 1.170 meters, where D_2 equals 0.00357. As the D_2 value increased, the positioning error significantly decreased, dropping from 1.330 meters to 0.839 meters with a standard deviation of 0.736 meters. In Sequence 3, the error was further reduced to 0.357 meters with an even higherD_2 value. It can be concluded that a more even geometric distribution of dynamic objects leads to lower accuracy in LiDAR odometry.

3) Verification of Relative Velocity of Dynamic Objects

Table 4.5: LiDAR odometry performance is evaluated under varying geometric distributions (D_2) with 50% dynamic objects. In Sequence 1, the baseline scenario includes 50% dynamic objects. In Sequence 2, the dynamic objects are rotated 90 degrees around their z-axis to create a different distribution. In Sequence 3, the objects are repositioned 2 meters away from the ego-vehicle.

Scenario	D_2	**RMSE (m)**	**MEAN (m)**	**STD (m)**
Sequence 1	0.00357	1.330	0.632	1.170
Sequence 2	0.00630	0.839	0.403	0.736
Sequence 3	0.00846	0.357	0.237	0.266

Table 4.6 shows the accuracy of LiDAR odometry under various relative velocity conditions (illustrated in Fig. 4.20). The second column lists the different speeds of the dynamic vehicles, while the third column presents the positioning results of LiDAR odometry. The results for Sequences 1-5 in Table 4.6 indicate that the presence of a single moving vehicle has minimal impact on state estimation. However, when an additional vehicle moving at 5 m/s (related to ego-vehicle) is introduced on the left, the positioning error increases to 4.375 meters, with a standard deviation of 3.699 meters. In Sequence 5, the error further escalates to 8.192 meters compared to Sequence 4 as the speed increases. Interestingly, the positioning error is drastically reduced to 0.408 meters, with a standard deviation of 0.271 meters. This significant reduction could be due to the fact that outliers with large movements are identified as corresponding features by the k-d tree data association algorithm.

In short, the positioning results are affected by the relative velocity factor, but faster speed might not lead to worse results.

4) Analysis of Residuals

To show further details of three factors denominating the performance of LiDAR odometry, we analyzed the residuals from evaluated scenarios, as illustrated in Fig. 4.23. The x-axis represents the residual values while the y-axis indicates the volume associated with residuals in the histogram.

Firstly, Fig. 4.23 (a) presents the residuals associated with 20% of dynamic ob-

Table 4.6: Performance of LiDAR odometry in terms of the relative velocity of dynamic objects. D_3 represents the relative speed of the dynamic objects, while right and both represent the placement of the dynamic objects on the right side and both sides.

Scenario	D_3 **(m/s)**	**RMSE (m)**	**MEAN (m)**	**STD (m)**
Sequence 1	5, right	0.198	0.167	0.106
Sequence 2	10, right	0.181	0.153	0.097
Sequence 3	20, right	0.178	0.148	0.098
Sequence 4	5, both	4.375	2.336	3.699
Sequence 5	10, both	8.192	4.688	6.717
Sequence 6	20, both	0.408	0.305	0.271

jects. These dynamic objects are more easily identifiable due to their larger residuals, resulting in minimal impact on the accuracy of LiDAR odometry, as indicated in Table 4.4. However, as the number of dynamic objects increases, the identification of outlier residuals becomes more challenging, as they tend to have values similar to those of static features (Fig. 4.23 (b)-(c)), leading to significant positioning errors.

Secondly, Fig. 4.23 (d)-(f) illustrates the residuals corresponding to different D_2 values. As shown in Fig. 4.23 (e), the residuals from dynamic objects increase with a higher D_2 value. Consequently, outliers can be more effectively managed if the distribution of dynamic objects is less uniform, even when the total number of dynamic objects remains the same.

Thirdly, Fig. 4.23 (g)-(i) illustrates the residuals associated with dynamic objects at varying relative speeds. Notably, dynamic objects with higher relative speeds exhibit fewer residuals compared to those with lower speeds. This suggests that outliers characterized by rapid movement may not be effectively associated with the corresponding features between consecutive frames.

Verification of Reweighting LiDAR Odometry

1) Verification of Proposed Method in the Mid-Dynamic Scene

To assess the effectiveness of the proposed reweighting method, experiments were

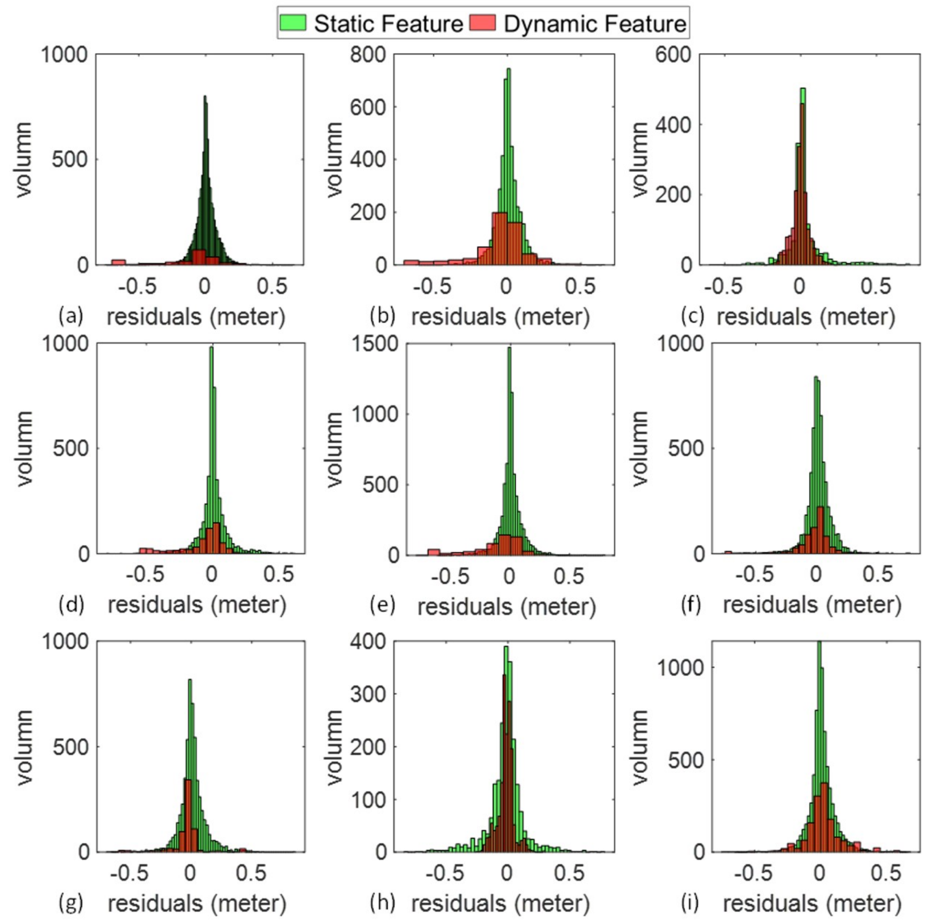


Figure 4.23: Illustration of the residual distributions of the compared scenarios on a single epoch. The x-axis represents the value of residuals. The y-axis indicates the volume associated with each bin of the residuals in the histogram. The residuals from static features are shown in green, and those from dynamic features are shown in brown. The residuals are presented under scenarios (a) 20% of dynamic objects; (b) 50% of dynamic objects; (c) 80% of dynamic objects; (d) D_2 value equals 0.00357; (e) D_2 value equals 0.00630; (f) D_2 value equals to 0.00846; (g) right side dynamic objects with relative speed 5 m/s; (h) both side dynamic objects with relative speed 10 m/s; (i) both side dynamic objects with relative speed 20 m/s. Note that some residuals plots are not on the same scale for illustration purposes.

carried out in a *mid-dynamic scene* featuring 24% dynamic objects and 30% static objects. Table 4.7 summarizes the positioning results for the four evaluated methods, while Fig. 4.24 and Fig. 4.25 illustrate the positioning errors and trajectories of these methods. The LO method produced an RMSE of 0.206 meters, with a standard deviation of 0.147 meters. By employing truncated least squares, the RMSE for T-LOAM was improved to 0.096 meters, making it the most accurate among the four methods tested. After dynamic object points were removed, the RMSE further decreased from 0.206 to 0.147 meters. With the introduction of the proposed reweighting method (LO-RW), both the RMSE and standard deviation were further reduced to 0.118 meters and 0.033 meters, respectively. The LO-RW method demonstrated enhanced performance, particularly in the turning area highlighted in Fig. 4.25. These findings validate the effectiveness of the proposed method in *mid-dynamic* environments. To further evaluate its performance, another challenging experiment is discussed in the following section.

Table 4.7: Positioning results of the proposed method in the mid-dynamic scene (24% dynamic points).

Results	**LO**	**T-LOAM**	**LO-R**	**LO-RW**
RMSE (m)	0.206	**0.096**	0.147	0.118
MEAN (m)	0.144	**0.081**	0.142	0.113
STD (m)	0.147	0.053	0.038	**0.033**

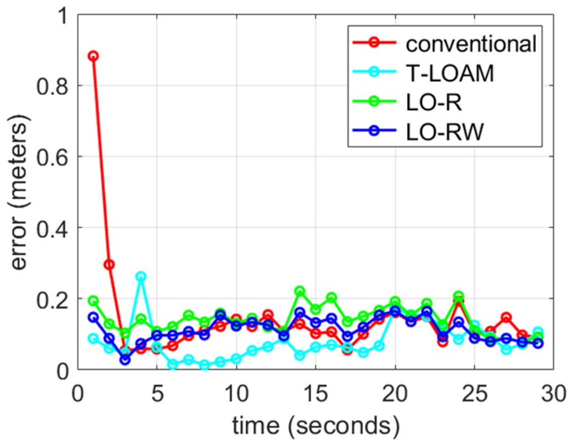


Figure 4.24: Positioning error of the four methods. The x-axis and y-axis denote the epoch and error, respectively.

2) Verification of Proposed Method in the High-Dynamic Scene

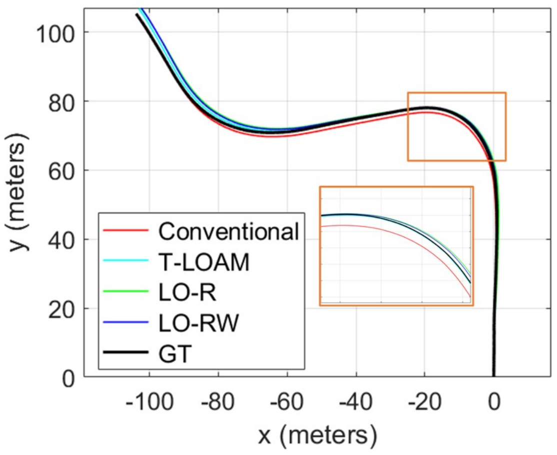


Figure 4.25: 2D positioning trajectories of the four methods. The x-axis and y-axis denote the x and y directions, respectively. The black curve represents the ground truth.

To further challenge the performance of the proposed method, an experiment was conducted in a highly dynamic environment with 40% moving objects. The results, shown in Table 4.8, highlight the effectiveness of the proposed LiDAR odometry approach. T-LOAM failed to optimize the trajectory, as seen in Fig. 4.27, due to incorrect outlier rejection in the presence of a large number of dynamic objects. Using the conventional LO method, an RMSE of 1.349 meters was obtained, with a standard deviation of 1.104 meters. This performance was worse than in the mid-dynamic scene, owing to the excessive number of dynamic objects in this scenario. After excluding all dynamic objects with LO-R, the error decreased to 0.146 meters. With the addition of object weighting optimization, the error was further reduced to 0.110 meters using LO-RW. The improvement in the results demonstrates the effectiveness of the proposed method for LiDAR odometry in highly dynamic environments.

Fig. 4.26 and Fig. 4.27 show the positioning error and trajectories, respectively. The more accurate trajectory is estimated using LO-RW with the help of dynamic object reweighting. After applying the object reweighting, the positioning performance of LiDAR odometry is improved significantly in the dynamic scene.

Table 4.8: Positioning results of the proposed method in the high-dynamic scene (40% dynamic points).

Results	LO	T-LOAM	LO-R	LO-RW
RMSE (m)	1.340	5.660	0.146	**0.110**
MEAN (m)	0.759	5.555	0.141	**0.112**
STD (m)	1.104	1.084	0.035	**0.041**

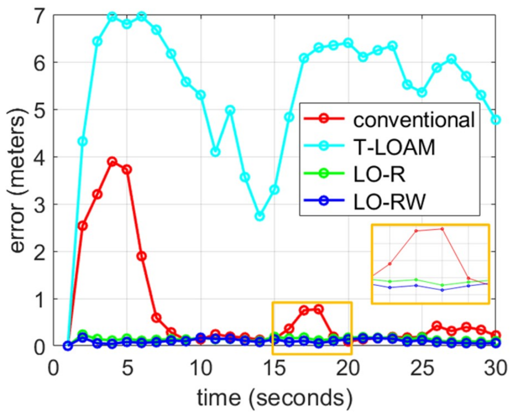


Figure 4.26: Positioning error of the four methods. The x-axis and y-axis denote the epoch and error, respectively.

3) Verification of Proposed Method using nuScenes Dataset

To evaluate the performance of the proposed method in challenging conditions, the third experiment was conducted combined with simulated dynamic objects from the *nuScenes* [106] Dataset. The positioning results are presented in Table 4.9. The conventional LO method yielded an RMSE of 2.592 meters, accompanied by a standard deviation of 1.755 meters. Similar to the high-dynamic scenario, T-LOAM struggled to optimize the trajectory, leading to substantial errors. The performance of LO-R was inferior to that of LO due to a lack of constraints after removing all vehicle points, as shown in Fig. 4.28-(b). In contrast, LO-RW achieved the best accuracy with an RMSE of 0.729 meters, benefiting from the reweighting strategy. These results confirm the effectiveness of the proposed method across extensive datasets. Fig. 4.9-(d) shows the final weighting assigned to all objects in the *nuScenes* dataset.

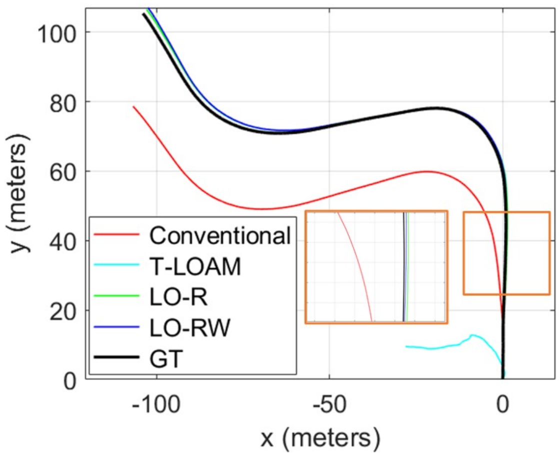


Figure 4.27: 2D positioning trajectories of the four methods. The x-axis and y-axis denote the x and y directions, respectively. The black curve represents the ground truth.

Notably, most points from the front of the dynamic vehicle received low weight due to inconsistencies, while points from the sides of the dynamic object were assigned higher weights because the residuals of point-to-surface distances remained small, despite the vehicle moving forward. Additionally, static features were assigned high weights, positively contributing to the final optimization.

Table 4.9: Positioning results of the proposed method in the *nuScenes* dataset.

Results	**LO**	**T-LOAM**	**LO-R**	**LO-RW**
RMSE (m)	2.592	6.043	5.069	**0.729**
MEAN (m)	1.907	5.729	4.354	**0.446**
STD (m)	1.755	1.922	2.595	**0.576**

4) Analysis of Weighting and Residual

As mentioned in Section 4.3.4, the constant penalty parameter for adaptive weighting is experimentally set to 0.1. Fig. 4.31 illustrates the weighting with different penalty parameters in a single epoch. From Fig. 4.31-(a), it can be observed that the weighting for dynamic objects is close to 1 when the penalty parameter is set to 1. The final residual in Equation 6 is significantly influenced by the penalty term due to the large residual introduced by this parameter. As the penalty parameter

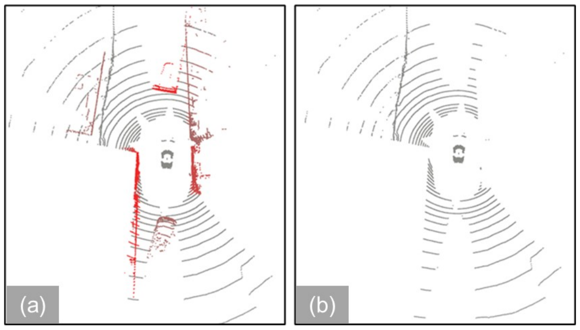


Figure 4.28: The example cloud in *nuScenes* is shown in (a) with dense objects marked in red; (b) illustrates the cloud after the removable objects have been removed.

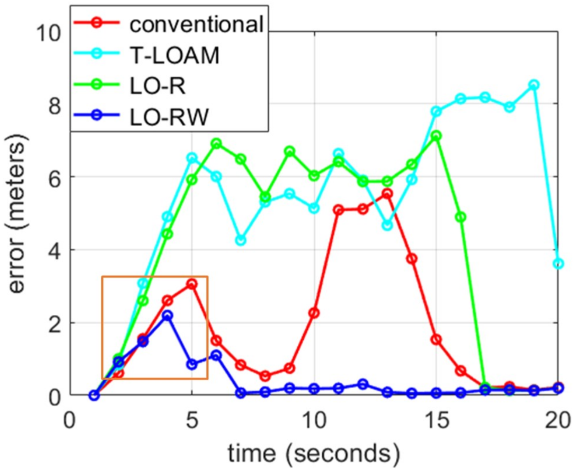


Figure 4.29: Positioning error of the four methods in *nuScenes*. The x-axis represents the epoch, while the y-axis indicates the error. The orange box highlights the tunnel-like area (see Fig. 4.28-a), which contains both dynamic and static objects. It has a negative impact on state estimation.

decreases, the inconsistencies (car front and rear) of the dynamic objects between consecutive epochs are assigned lower weights, as expected, as shown in Fig. 4.31-(b). However, Fig. 4.31-(c) indicates that the weighting for both dynamic and static objects approaches 0 when the penalty parameter is further reduced to 0.01, resulting in less constraint on state estimation. In other words, the performance of the proposed method becomes similar to that of the LO-R method with dynamic object removal.

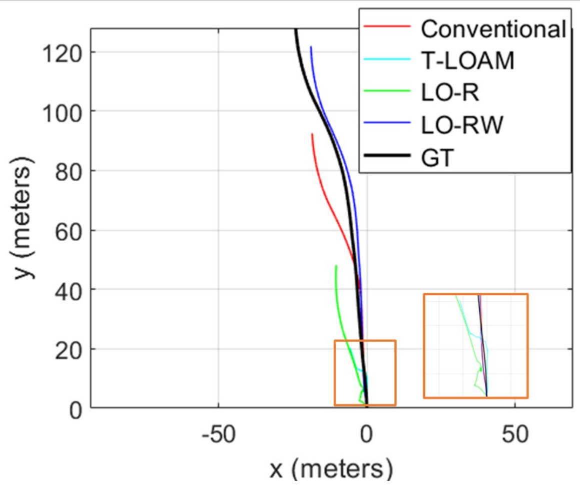


Figure 4.30: 2D positioning trajectories of the four methods in *nuScenes*. The x-axis represents the x-direction, while the y-axis denotes the y-direction. The black curve indicates the ground truth. The orange box highlights the area with a tunnel-like scene (see Fig. 4.28-a). It has a negative impact on state estimation.

To evaluate the residual of the optimization after applying the dynamic object reweighting method, Table 4.10 presents the residuals from different iterations with respect to the estimated states for both static and object points. It can be observed that the residuals for static points and total cost decrease gradually as the number of iterations increases. The reweighting scheme provides a better initial guess for static feature association, suggesting that the performance of the proposed method can be further enhanced through coarse-to-fine optimization.

5) Analysis of Convexity

To prevent our solution from being trapped in a local minimum within the solution space, we investigate the convexity of the state estimation using the proposed method. We introduced translation offsets of 10, 100, and 1000 meters to the initial guess of the state following feature association. Table 4.11 displays the residuals examined after each optimization using Ceres Solver. It can be observed that the cost can be minimized after several iterations, even with an initial guess offset of 1000 meters. Therefore, we can conclude that our problem remains convex within a

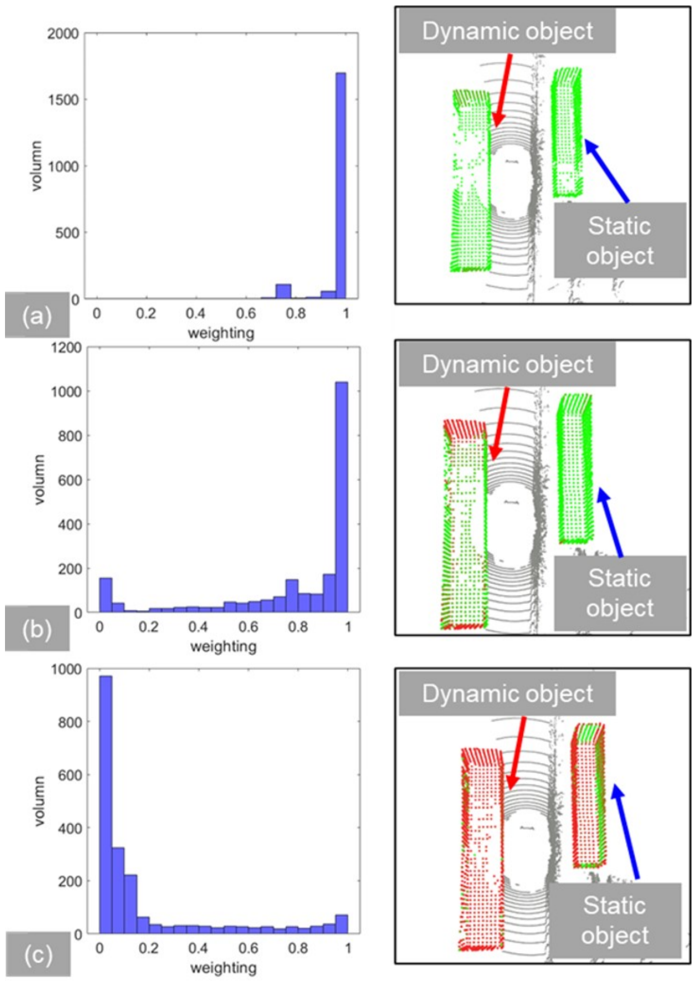


Figure 4.31: Illustration of the weighting effects based on different penalty parameters during state optimization. (a) Penalty parameter k=1; (b) Penalty parameter k=0.1 ; (c) Penalty parameter k=0.01. Lower weightings are indicated in red, while higher weightings are marked in green.

range of less than 1000 meters.

6) Analysis of Computational Time Cost

To assess the computational efficiency of the proposed method, the computation time for the evaluated methods is presented in Table 4.12. We compare the feature extraction, optimization processes, and total odometry using our evaluated dataset. During the optimization phase, our proposed LO-RW method required an additional 40 milliseconds compared to LO, primarily due to the estimation of more parame-

Table 4.10: Residuals of the proposed method are analyzed in relation to iterations within a single epoch. Each iteration utilizes the initial guess from the previously estimated state.

Iteration	Static Residual (m)	Dynamic Residual (m)	Total Residual (m)
1	9.588	0.805	10.393
2	9.216	0.780	9.995
3	8.952	0.809	9.760
4	8.810	0.827	9.637
5	8.729	0.829	9.558
6	8.780	0.835	9.615
7	8.775	0.834	9.609
8	8.775	0.833	9.608

Table 4.11: Results of the cost based on various initial guesses within a single epoch.

Offset (m)	Iteration	Cost	Cost Change
10	1	2.715e+05	/
	2	42.508	2.71e+05
	3	22.886	19.622
	4	22.842	0.042
100	1	2.701e+07	/
	2	86.257	2.70e+07
	3	23.790	0.963
	4	22.826	0.029
1000	1	2.684e+09	/
	2	6.574e+05	2.68e+09
	3	259.113	6.57e+05
	4	24.997	234.116
	5	22.679	2.318
	6	22.659	0.020

ters. Overall, the performance of the proposed method remains suitable for real-time applications.

4.4 Summary

Section 4.3 presents a comprehensive evaluation of how dynamic objects negatively impact the performance of LiDAR odometry. By analyzing various scenarios,

Table 4.12: The computation time for the evaluated methods is presented. MD stands for mid-dynamic scene, while HD refers to the high-dynamic scene. ”ms” denotes milliseconds. The odometry time is the sum of feature extraction time and optimization time.

Results	LO	T-LOAM	LO-R	LO-RW
MD Feature Extraction				
MEAN (ms)	14.5	24.4	14.8	15.0
MAX (ms)	20.6	48.0	22.4	22.1
STD (ms)	2.0	4.1	2.1	2.1
MD Optimization				
MEAN (ms)	72.1	36.6	71.5	111.7
MAX (ms)	96.9	67.0	100.0	175.2
STD (ms)	13.3	5.2	11.7	20.1
MD Odometry				
MEAN (ms)	86.6	61.0	86.3	126.7
HD Feature Extraction				
MEAN (ms)	15.4	24.6	15.5	15.4
MAX (ms)	22.0	43.0	21.2	21.0
STD (ms)	1.8	3.7	2.1	2.1
HD Optimization				
MEAN (ms)	82.9	48.0	71.8	116.3
MAX (ms)	111.8	73.0	108.5	196.3
STD (ms)	17.2	12.0	15.8	26.9
HD Odometry				
MEAN (ms)	98.3	72.6	87.3	131.7

this section highlights the specific challenges posed by dynamic elements in urban environments. In contrast to the work discussed in Section 4.2, this study proposes a reweighting strategy for dynamic objects, which enables the system to adaptively adjust the influence of these objects during state estimation. This approach results in significantly improved accuracy in both highly dynamic and adverse scenes. The experimental results in our *UrbanNav* dataset and the open-sourced *nuScenes* dataset demonstrate the effectiveness of the proposed method, which outperforms both traditional approaches and those involving dynamic object removal [16]. By effectively addressing the complexities introduced by dynamic objects, our method enhances the reliability and robustness of LiDAR odometry, paving the way for more accurate navigation in challenging urban areas.

Chapter 5

LiDAR Aided Cycle Slip Detection for GNSS Real-time Kinematic Positioning in Urban Areas

5.1 Introduction

Existing GNSS positioning methods typically utilize EKF [24] or FGO [111] to estimate the position of the GNSS receiver, achieving meter-level accuracy [112; 113; 114] through pseudorange and Doppler measurements. To enhance GNSS positioning accuracy, GNSS-RTK techniques have been introduced [115], enabling centimeter-level positioning to meet the navigation requirements of autonomous systems. Specifically, GNSS-RTK mitigates systematic errors by applying the DD operation [24] between observations from a reference station and those from the user. Ideally, centimeter-level accuracy can be attained in open areas when high-resolution DD carrier-phase and pseudorange measurements are employed, provided that the integer ambiguities associated with the DD carrier-phase are correctly resolved, resulting in a fixed solution. However, the accuracy of GNSS-RTK positioning is significantly compromised in urban environments, such as Hong Kong, due to excessive cycle slips [116] caused by signal reflections from buildings. The estimation of the fixed solution for GNSS-RTK heavily relies on integer ambiguity resolution, which assumes that the resolved integer ambiguity remains constant [24]. Unfortunately, this assumption is often violated in the presence of cycle slips, leading to substantial errors in GNSS-RTK if these slips are not properly detected prior to ambiguity resolution. For instance, even a single cycle slip can introduce a range error of up to 19 centimeters for GPS L1 measurements [117]. There are three major sources [118] of cycle slips. First of all, the cycle slips are caused by signal reflection due to the buildings, bridges, trees, etc., which are common elements in urban environments. Secondly, the cycle slip can suffer from signal-to-noise ratio (SNR) losses due to the multipath, satellite with low elevation angle, or high dynamics of the receiver. Third, false signal processing [119] might occur because of the receiver software failure. In short, effective cycle slip detection is crucial for achieving precise GNSS-RTK positioning in urban canyons.

To mitigate the errors caused by cycle slips, numerous studies [117; 120] have focused on detecting these inconsistencies. The underlying principles of these methods are generally similar, as they leverage additional information or sensors to identify discrepancies in carrier-phase measurements. For instance, Bisnath [120] and Gao [121] utilized DD observations to detect cycle slips through L1 and L2 observable combinations using typical dual-frequency GNSS receivers. Additionally, Blewitt [122] introduced the TurboEdit method, which employs undifferenced dual-frequency carrier-phase data to identify cycle slips. However, these methods are specifically designed for dual-frequency systems and are not applicable to single-frequency receivers. For single-frequency receivers, techniques such as code-phase difference, Doppler integration, and time difference of carrier-phase measurements can be employed [123] to detect cycle slips. However, these methods also have limitations. The code-phase difference is often ineffective at detecting small cycle slips due to significant noise in code measurements. While Doppler observations are generally resilient to cycle slips [119] and can be combined with carrier measurements, they still struggle to detect small cycle slips [117] and are susceptible to multipath effects in urban environments.

Similarly, the time difference of carrier-phase measurements shows inadequate detection accuracy in high-dynamic conditions, such as those encountered in autonomous applications. In summary, the accuracy of cycle slip detection for small slips and under high receiver dynamics remains limited.

To address these limitations, some researchers [124; 125] have integrated inertial navigation systems (INS) with GNSS receivers for cycle slip detection. The INS provides high-frequency relative position data by integrating acceleration and rotation measurements based on the previous position obtained from the GNSS system. By comparing the geodetic range between the satellite and the INS-predicted state with the carrier-phase measurements, the probability of cycle slips can be assessed. However, a standard automobile-level INS may not suffice, as its performance heavily depends on the correction of internal biases [126] in the accelerometers and gyroscopes. In summary, these methods often incur significant costs due to the reliance on high-quality INS.

Recently, LiDAR sensors have gained popularity in autonomous systems due to their ability to provide accurate and high-frequency LO [18]. The LO method estimates the state by accumulating relative transformations between consecutive frames. Our previous work [1] demonstrated that LiDAR sensors can achieve accurate relative state estimation in urban environments. Inspired by this, this chapter proposes a method for detecting cycle slips using time-differenced carrier-phase measurements in conjunction with LiDAR odometry. The precise LiDAR pose estimation is utilized to determine the relative motion of the GNSS receiver. Cycle slip detection is performed by comparing consecutive epochs of LiDAR-predicted ranges with the received carrier-phase measurements. Once cycle slips are identified, the integer ambiguity can be re-estimated to obtain a fixed solution for GNSS-RTK.

5.2 Methods

The proposed framework is illustrated in Fig. 5.1. The pipeline consists of two main components: (1) the predicted state derived from LiDAR odometry and orientation provided by the attitude and heading reference system (AHRS) [127], and (2) cycle slip detection based on the time difference between DD carrier-phase measurements and DD LiDAR-derived ranges. The difference between two DD measurements taken over different epochs is referred to as the triple difference. The primary purpose of using the triple-difference method is to eliminate systematic errors. Note that the relative pose estimation concerning the LiDAR local frame should be transformed into the GNSS global frame before its utilization in cycle slip detection. To achieve this, the orientation from the AHRS is directly adopted. The system inputs include point cloud data from the LiDAR sensor, orientation from the AHRS, and raw measurements from the GNSS receiver and the reference station. The pose estimated by LiDAR odometry is aligned with the ECEF frame, corresponding to the light-blue boxes in Fig. 5.1, which will be discussed in subsequent sections. Cycle slip detection is performed by comparing the triple-differenced carrier-phase measurements with predictions from the LiDAR odometry, as represented by the light-yellow boxes in Fig. 5.1. Finally, the ambiguity is re-estimated to achieve a fixed solution.

5.2.1 State Estimation Based on LiDAR Odometry

This section outlines the methodology of LiDAR odometry and the transformation between the LiDAR and ECEF frames. The study in [18] demonstrated that LiDAR odometry can achieve low-drift state estimation in urban environments.

ECEF Transformation

Given the pre-calibrated extrinsic parameters among the LiDAR, AHRS, and GNSS receiver, the LiDAR pose in the LiDAR frame can be transformed into the receiver

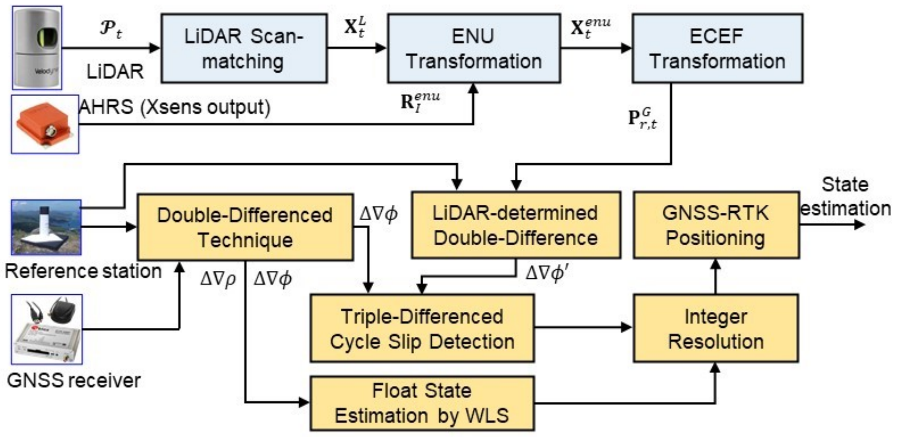


Figure 5.1: Overview of LiDAR-aided cycle slip detection. WLS denotes the weighted least squares method.

pose in the ENU frame as follows,

$$\mathbf{x}_{r,t}^{\text{enu}} = \begin{bmatrix} \mathbf{R}_I^{\text{enu}} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}_{4\times 4} \mathbf{T}_I^r \mathbf{T}_L^I \mathbf{x}_t^L \tag{5.1}$$

where $\mathbf{T}_L^I$ is extrinsic to transform the pose from the LiDAR frame to the AHRS frame. $\mathbf{T}_I^r$ is the extrinsic parameter between the AHRS and the antenna of the GNSS receiver. $\mathbf{R}_I^{\text{enu}}$ is obtained by AHRS orientation estimation.

The origin of the ECEF coordinate system is located at the Earth's center of mass based on the WGS 84 ellipsoid [128]. To transform coordinates from ENU to ECEF, a reference point in the ENU system is required. The first fixed solution, $\mathbf{P}_{\text{fix}}^G$ obtained through RTK-GNSS, is selected as the reference point for the ENU frame. The state in the ENU frame can be converted to the ECEF frame as outlined in [129],

$$\mathbf{P}_{r,t}^G = \begin{bmatrix} -\sin\lambda & -\sin\phi\cos\lambda & \cos\phi\cos\lambda \\ \cos\lambda & -\sin\phi\sin\lambda & \cos\phi\sin\lambda \\ 0 & \cos\phi & \sin\phi \end{bmatrix} \text{tran}(\mathbf{x}_{r,t}^{\text{enu}}) + \mathbf{P}_{\text{fix}}^G \tag{5.2}$$

where the ϕ and λ denote the geodetic latitude and longitude of the reference point, respectively. Note that the first point is selected as the reference point in this section. The operator tran($*$) is defined as the translation of the state.

5.2.2 LiDAR-Aided Cycle Slip Detection

Cycle Slip Detection Aided by LiDAR Sensor

A carrier-phase measurement in units of length from the GNSS receiver can be expressed as [130],

$$\phi_{r,t}^{s} = r_{r,t}^{s} + c(\delta r_{r,t} - \delta s_{r,t}) - I_{r,t}^{s} + T_{r,t}^{s} + \lambda^{s} N_{r,t}^{s} + \epsilon_{r,t}^{s} \tag{5.3}$$

where λ^s denotes the carrier wavelength of the corresponding GNSS signal. $r_{r,t}^{s}$ is the range distance between the satellite and the GNSS receiver. c denotes the speed of light. $\delta r_{r,t}$ and $\delta s_{r,t}$ represent the receiver clock bias and the satellite clock bias, respectively. $I_{r,t}^{s}$ and $T_{r,t}^{s}$ represent the delay due to ionospheric and tropospheric layers, respectively. $N_{r,t}^{s}$ is the integer ambiguity value of the carrier-phase. $\epsilon_{r,t}^{s}$ denotes the unmodeled error such as receiver thermal noise and multipath.

An overview of DD carrier-phase measurement is illustrated in Fig. 5.2. The single difference between the receiver and base station using a common master satellite eliminates satellite bias, as well as ionospheric and tropospheric effects. Typically, the satellite with the highest elevation angle is chosen as the master satellite, which is the approach followed in this section. The receiver clock bias is further removed by applying a between-satellite single difference. The DD carrier-phase measurement for GNSS-RTK can be formulated as follows [112],

$$\Delta\nabla\phi_{r,t}^{s} = \left(\phi_{r,t}^{s} - \phi_{b,t}^{s}\right) - \left(\phi_{r,t}^{w} - \phi_{b,t}^{w}\right) = \Delta\nabla r_{r,t}^{s} + \lambda\Delta\nabla N_{r,t}^{s} + \Delta\nabla\epsilon_{r,t}^{s} \tag{5.4}$$

where the satellite w denotes the master satellite, selected based on the highest elevation angle. $\Delta\nabla N_{r,t}^{s}$ represents the DD ambiguity that must be resolved to obtain the fixed solution. $\Delta\nabla\epsilon^s_{r,t}$ denotes the noise associated with the DD carrier-phase measurements.

Given the LiDAR-predicted receiver pose in the ECEF frame, as derived in Equation 5.2, the range distance from satellite s to the receiver is calculated as,

$$\phi'^s_{r,t} = \left\|\mathbf{P}^s_t - \mathbf{P}^G_{r,t}\right\| = \sqrt{(P^s_{t,x} - P^G_{r,t,x})^2 + (P^s_{t,y} - P^G_{r,t,y})^2 + (P^s_{t,z} - P^G_{r,t,z})^2} \tag{5.5}$$

where the $\phi'^s_{r,t}$ is the LiDAR-determined range distance between satellite and receiver. $\mathbf{P}^s_t = (P^s_{t,x}, P^s_{t,y}, P^s_{t,z})$ and $\mathbf{P}^G_{r,t} = (P^G_{r,t,x}, P^G_{r,t,y}, P^G_{r,t,z})$ are the position of the satellite and the receiver in the ECEF frame, respectively. Therefore, the predicted DD range measurement aided by LiDAR sensors can be expressed as,

$$\Delta\nabla\phi'^s_{r,t} = (\phi'^s_{r,t} - \phi'^s_{b,t}) - (\phi'^w_{r,t} - \phi'^w_{b,t}) \tag{5.6}$$

Fig. 5.2 illustrates the difference of DD carrier-phase measurement, so-called the triple difference measurements, between two successive epochs, which can be expressed as follows,

$$\begin{aligned}\delta\Delta\nabla\phi^s_{r,t} &= \Delta\nabla\phi^s_{r,t} - \Delta\nabla\phi^s_{r,t-1} \\ &= \Delta\nabla r^s_{r,t} - \Delta\nabla r^s_{r,t-1} \quad + \lambda\left(\Delta\nabla N^s_{r,t} - \Delta\nabla N^s_{r,t-1}\right) \quad + \delta\Delta\nabla\varepsilon^s_{r,t}\end{aligned} \tag{5.7}$$

$\delta\Delta\nabla\epsilon^s_{r,t}$ indicates the change in the unmodeled DD error between consecutive epochs.

Similarly, the time-differenced LiDAR-predicted range can be expressed as,

$$\delta\Delta\nabla\phi'^s_{r,t} = \Delta\nabla\phi'^s_{r,t} - \Delta\nabla\phi'^s_{r,t-1} \tag{5.8}$$

Given that LiDAR odometry can provide highly accurate relative motion between two frames with a short time difference, the $\Delta\nabla r^s_{r,t}$ should equal to $\Delta\nabla\phi'^s_{r,t}$. The difference between $\delta\Delta\nabla\phi^s_{r,t}$ and $\delta\Delta\nabla\phi'^s_{r,t}$ can be denoted as,

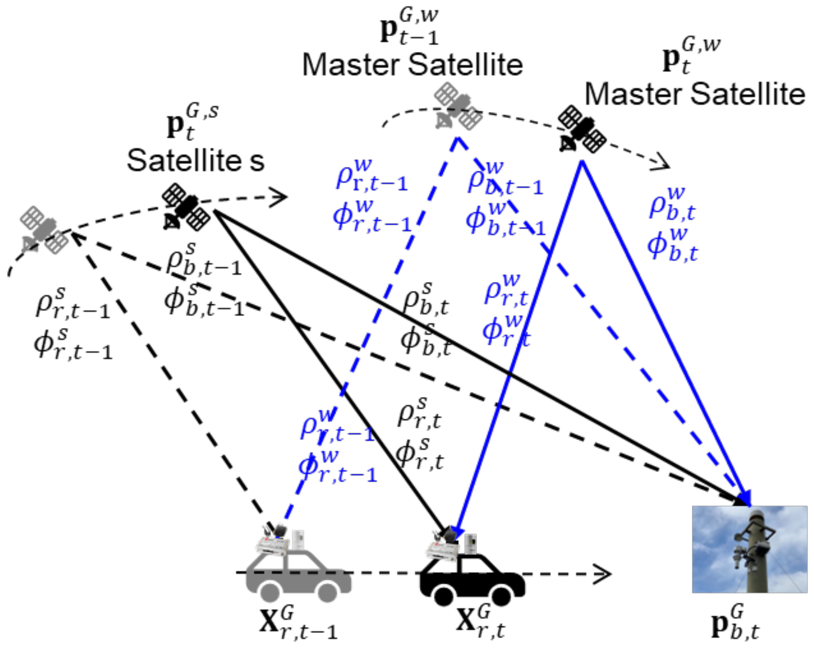

$\mathbf{p}_{t-1}^{G,w}$
Master Satellite
$\mathbf{p}_{t}^{G,w}$
Master Satellite
$\mathbf{p}_{t}^{G,s}$
Satellite s
$\rho_{r,t-1}^{w}$ $\phi_{r,t-1}^{w}$
$\rho_{b,t-1}^{w}$ $\phi_{b,t-1}^{w}$
$\rho_{b,t}^{w}$ $\phi_{b,t}^{w}$
$\rho_{r,t}^{w}$ $\phi_{r,t}^{w}$
$\rho_{b,t-1}^{s}$ $\phi_{b,t-1}^{s}$
$\rho_{r,t-1}^{s}$ $\phi_{r,t-1}^{s}$
$\rho_{b,t}^{s}$ $\phi_{b,t}^{s}$
$\rho_{r,t}^{s}$ $\phi_{r,t}^{s}$
$\mathbf{X}_{r,t-1}^{G}$
$\mathbf{X}_{r,t}^{G}$
$\mathbf{p}_{b,t}^{G}$

Figure 5.2: Illustration of triple-differenced carrier-phase measurements.

$$d_{\delta\phi,t}^{s} = \delta\Delta\nabla\phi_{r,t}^{s} - \delta\Delta\nabla\phi_{r,t}^{\prime s} = \lambda(\Delta\nabla N_{r,t}^{s} - \Delta\nabla N_{r,t-1}^{s}) \tag{5.9}$$

where the $d_{\delta\phi,t}^{s}$ represents the difference between the triple-differenced carrier-phase measurements and LiDAR-predicted range measurements, referred to as DCL. When the carrier-phase measurement is free of cycle slips, the DD ambiguity $\Delta\nabla N_{r,t}^{s}$ remains constant across consecutive epochs. As a result, the DCL residual should remain small if there is no cycle slip. Therefore, Equation 5.9 can be used to detect cycle slips if $d_{\delta\phi,t}^{s}$ exceeds an experimentally determined threshold T_{DCL},

$$|d_{\delta\phi,t}^{s}| > T_{DCL} \tag{5.10}$$

GNSS-RTK Positioning Aided by Cycle Slip Detection

GNSS-RTK positioning involves two key steps: float state estimation and integer ambiguity resolution. The first step uses WLS [24] to estimate the float solution. In the second step, the integer ambiguity resolution (AR) is carried out using the least-squares ambiguity decorrelation adjustment (LAMBDA) algorithm [130] algorithm in the second stage. If a cycle slip occurs, the AR must be reprocessed to obtain a fixed solution. The GNSS-RTK implementation is based on RTKLIB [24].

Cycle slip detection using LLI: The receiver provides a Loss of Lock Indicator (LLI) indicator representing the status of cycle slips. As defined by the Receiver Independent Exchange Format (RINEX) 3.03 [131], 3 bits are used for cycle slip detection. If the bits are set to 0 or left blank, it indicates either no cycle slip or an unknown status. Bit 0 being set indicates a potential cycle slip, while bit 1 signals the presence of a half-cycle ambiguity or slip. Additionally, bit 2 is set when BOC tracking occurs on an MBOC-modulated signal [131]. In urban environments, cycle slips are often flagged by the LLI due to signal reflections from nearby buildings.

5.3 Performance Evaluation

5.3.1 Experiment Setup

To validate the performance of the proposed LiDAR-aided cycle slip detection method, we conducted experiments in typical urban environments in Hong Kong using our open-source *UrbanNav* [3] datasets. These datasets include measurements from multi-GNSS receivers, an INS, cameras, and multiple LiDAR sensors. The GT positioning was obtained using the NovAtel SPAN-CPT, which integrates a fiber optic gyroscope with GNSS-RTK, as shown in Fig. 5.3. To maximize trajectory accuracy, we post-processed the GT data from the SPAN-CPT using the state-of-the-art NovAtel Inertial Explorer software. In this experiment, a commercial u-blox F9P GNSS receiver was employed to collect raw measurements from GPS and BeiDou at a frequency of 1 Hz, while the LiDAR operated at a frame rate of 10 Hz. Additionally, the GNSS time and LiDAR timestamps were hardware-synchronized [109] during data collection. The extrinsic parameters of the GNSS receiver, LiDAR sensor, and NovAtel SPAN-CPT were calibrated in advance.

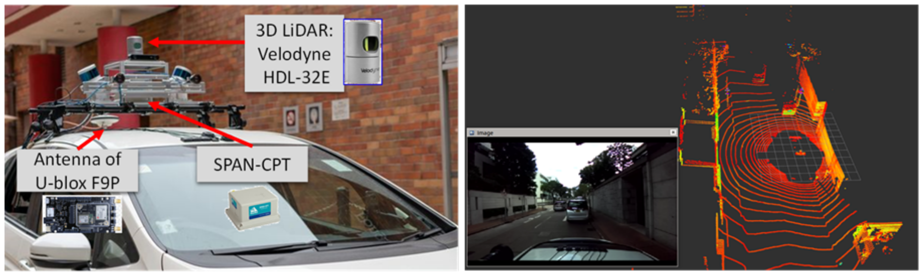


Figure 5.3: Left: Setup of the sensor platform, which includes a u-blox F9P receiver, Velodyne HDL-32E 3D LiDAR, and ground truth positioning provided by the SPAN-CPT. Right: A typical urban environment in Hong Kong for evaluation.

5.3.2 Evaluation Metrics and Methods

Cycle Slip Ground Truth Labeling

To validate the effectiveness of the proposed cycle slip detection method, we use ground truth positioning data from the NovAtel SPAN-CPT to label cycle slips. Specifically, the relative pose from the ground truth estimation is substituted for LiDAR odometry in Equation 5.7.

The range measurement between the satellite and the receiver at epoch t can be obtained using Equation 5.5, after transforming the SPAN-CPT data to the receiver's antenna. Furthermore, the triple-difference ground truth-based range measurement can be computed from the DD range measurements across consecutive epochs.

Recall Equation 5.7, the difference $d^s_{\delta\phi,gt,t}$ between the triple-differenced carrier-phase $\delta\Delta\nabla\phi^s_{r,t}$ and GT-based range measurement $\delta\Delta\nabla r^s_{r,t}$ can be expressed as follows,

$$d^s_{\delta\phi,gt,t} = \delta\Delta\nabla\phi^s_{r,t} - \delta\Delta\nabla r^s_{r,t} = \lambda(\Delta\nabla N^s_{r,t} - \Delta\nabla N^s_{r,t-1}) \tag{5.11}$$

$d^s_{\delta\phi,gt,t}$ can be used to label the cycle slips when the threshold is exceeded.

Performance Evaluation of the GNSS-RTK Cycle Slip Detection

In this work, the performance evaluation of the LiDAR-aided GNSS-RTK cycle slip detection method is based on the cycle slip ground truth labeling results from the previous section. We evaluate only the data epochs where DD measurements are available in consecutive epochs, as the ground truth labeling relies on the triple-differenced measurements. The accuracy of the cycle slip detection can be defined as follows:

$$P_{\mathrm{dr}} = \frac{N_{\mathrm{cs}} \cap N_{\mathrm{gt}}}{N_{\mathrm{gt}}} \tag{5.12}$$

where P_{dr} denotes the percentage of detection rate. N_{cs} denotes the amount of cycle slip satellites detected using the proposed method. N_{gt} denotes the number of cycle slip satellites detected using the ground truth labeling in the previous Section. A higher value of P_{dr} means a larger overlapping level of the cycle slip detection and GT labeling.

To evaluate the contributions of the proposed method in cycle detection, the following methods are evaluated:

- **LLI:** The cycle slips are marked by the LLI flags.
- **LAD:** The proposed LiDAR-aided cycle slip detection scheme.

Performance Evaluation of the GNSS-RTK Positioning

The fixed solution is determined using raw measurements from both the user's GNSS receiver (rover) and the base station's GNSS receiver through RTKLIB [24]. The configuration for RTKLIB evaluation is detailed in Table 5.1.

For improved classification, the availability and fixing rate of GNSS-RTK is also defined. Availability P_a is calculated as the percentage of epochs N_{sol} successfully

Table 5.1: Process Setting in RTKLIB

Parameter	**Value**	**Parameter**	**Value**
Positioning Mode	Kinematic	Satellite System	GPS/Beidou
Ionosphere Model	Broadcast	Frequency	L1
Troposphere Model	Saastamoinen Model	Elevation Mask	15 degrees
Integer Ambiguity Resolution	Fix and Hold	Ephemeris	Hong Kong Land Department
Min Ratio to Fix Ambiguity	3.0	Filter Type	Forward

resolved by RTKLIB, divided by the total number of epochs N_{eph} in the GNSS measurement,

$$P_a = \frac{N_{\text{sol}}}{N_{\text{eph}}} \tag{5.13}$$

The fixing rate P_{fix} is defined as the percentage of epochs where the integer fixed successfully N_{fix}, divided by the total number of epochs N_{sol},

$$P_{\text{fix}} = \frac{N_{\text{fix}}}{N_{\text{sol}}} \tag{5.14}$$

To validate the contributions of the proposed method, we evaluate the following four approaches:

1. **RTKLIB:** This conventional GNSS-RTK positioning method [24] uses LLI to detect cycle slips. If a cycle slip occurs, the ambiguity will be re-estimated.

2. **RTK-LA:** This method involves GNSS-RTK positioning with LiDAR-aided cycle slip detection only. Ambiguity re-estimation occurs if a cycle slip is detected.

3. **RTK-LAE:** This approach combines LiDAR-aided cycle slip detection with LLI flags. Satellites identified with cycle slips are excluded from position estimation.

4. **RTK-LAR:** The proposed method integrates LiDAR-aided cycle slip detection with LLI flags. If a cycle slip is detected, the estimated ambiguity is reset and re-fixed.

5.3.3 Experimental Evaluation in Urban Environments

The experiment was conducted in a 675-meter path within an urban area of Kowloon Town, Hong Kong. An example scenario is depicted on the right side of Fig. 5.3. The environment, characterized by numerous trees and buildings, represents a typical urban setting where frequent cycle slips occur.

GNSS-RTK Cycle Slip Detection in Urban Areas

The results of cycle slip detection using the LLI and LiDAR-aided methods are presented in Table 5.2. A total of 104 cycle slips were labeled by the ground truth out of 1,235 measurements, indicating that cycle slips accounted for 8.4% of this dataset. Among these, 33 common-view measurements were identified from the 402 LLI slip detections based on the ground truth labeling. The remaining 369 LLI flags were not included in the cycle slip comparison because the ground truth labeling only considers successive carrier-phase measurements. In other words, LLI flags from unlabeled measurements were not taken into account. With the help of LiDAR odometry, the RTK-LA method successfully identified 94.2% of the cycle slips, whereas only 10.6% were detected by LLI flags. Additionally, we observed that the number of cycle slips indicated by LLI is considerably higher than the number labeled by the ground truth or detected by the LiDAR-aided method. This discrepancy arises because cycle slips are often produced and flagged by LLI at the onset of signal tracking or in low SNR conditions typical of urban environments.

Table 5.2: Performance evaluation of cycle slip detection in the urban environment. $N_{\mathrm{cs}} \cap N_{\mathrm{gt}}$ denotes the number of cycle slip satellites detected by both the proposed method and the ground truth labeling from the previous section.

Results	**Ground Truth Labeled**	**LLI**	**LAD**
Number of Detection	104	402	105
Number of Correctly Detected, $N_{\mathrm{cs}} \cap N_{\mathrm{gt}}$	/	33	98
Detection Rate P_{dr}	/	10.6%	**94.2%**

Fig. 5.4 and Fig. 5.5 illustrate the cycle slip detection results for G01 and C11, respectively. Most cycle slips can be identified based on the experimental threshold, marked in red dash-dot at 0.57 m (equivalent to 3 cycles for L1 measurements). Notably, the majority of LLI flags in G01 and C11, are highlighted in the red boxes of Figs. 5.4 and 5.5, are detected at the beginning of tracking or during periods of discontinuous measurement in urban environments. In contrast, the RTK-LA method effectively detects cycle slips during continuous measurements. In summary, the cycle slip detection capabilities of LLI flags and the LiDAR-aided approach could complement each other for GNSS-RTK positioning.

GNSS-RTK Fixing rate and Positioning results in Urban Areas

To effectively evaluate the cycle slip detection capabilities of the LLI and LiDAR-aided methods, several positioning approaches are assessed to explore their combination. The results of the fixing rates for the four methods are presented in Table 5.3. The proposed RTK-LAR achieved a fixing rate of 25.25%, the highest among the evaluated GNSS-RTK methods. In contrast, RTK-LA performed poorly regarding the fixing rate due to some discontinuous carrier-phase measurements, which impacted the usability of the proposed triple-differenced formulation. RTK-LAE demonstrated better performance by excluding measurements from satellites with

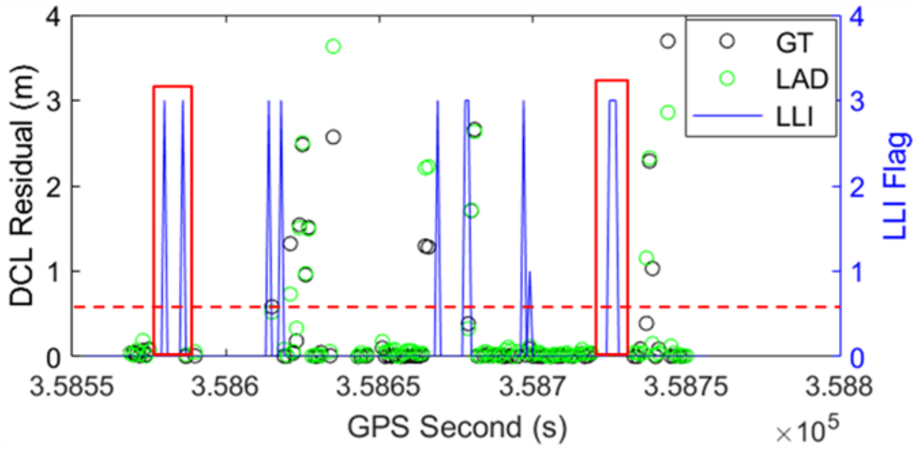


Figure 5.4: The cycle slip detection results for satellite G01. The x-axis represents GPS seconds, the left y-axis shows the absolute value of the DCL residual, and the right y-axis indicates the LLI flag. The red dash-dot line denotes the threshold for cycle slips in DCL residuals, set at 0.57 m..

cycle slips. However, it suffered from lower availability, as it excessively excluded all measurements with detected cycle slips.

Table 5.3: Performance evaluation of GNSS-RTK in terms of fixing rate and 2D position error.

Results	**RTKLIB**	**SPP-INS** [132]	**PPK-INS** [132]	**RTK-LA**	**RTK-LAE**	**RTK-LAR**
Availability P_a	99.02%	93.14%	93.14%	99.02%	97.55%	**99.02%**
Fixing Rate P_{fix}	7.92%	/	10%	3.96%	14.57%	**25.25%**
Improvement of Fixing Rate	/	/	2.08%	-3.96%	6.65%	**17.33%**
MEAN (m)	1.298	4.822	**0.963**	2.715	1.173	**1.118**
STD (m)	2.459	2.536	**1.658**	3.590	2.451	**2.446**
Improvement of Positioning	/	/	**25.81%**	/	9.64%	**13.86%**

In terms of positioning accuracy, the RTK-LAR method outperforms other approaches, achieving a mean error of 1.118 m and a 13.86% improvement in mean positioning error compared to conventional GNSS-RTK, as shown in Table 5.3. The

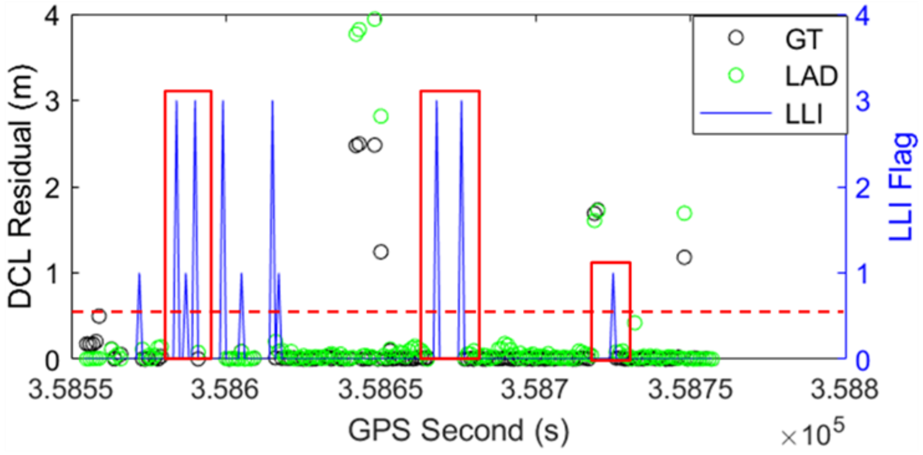


Figure 5.5: The cycle slip detection result for satellite C11. The x-axis represents GPS seconds, the left y-axis shows the absolute value of the DCL residual, and the right y-axis indicates the LLI flag. The red dash-dot line denotes the threshold for cycle slips in DCL residuals, set at 0.57 m.

commonly used INS-aided PPK-INS loosely coupled solutions [132] offer superior positioning performance compared to LiDAR-aided GNSS-only solutions, as they integrate GNSS with accelerometer and gyroscope data, enabling improved position and orientation estimation while also detecting cycle slips through GNSS carrier-phase cross-checking.

Fig. 5.6 and Fig. 5.7 illustrate the trajectories and positioning errors for the listed methods, respectively. The RTK-LAR method, aided by LiDAR cycle slip detection, provides a more accurate trajectory estimate, particularly in the areas highlighted in Fig. 5.6. The implementation of LiDAR-aided cycle slip detection significantly enhances GNSS-RTK positioning performance in this urban environment, especially in the regions marked in purple and orange in Figs. 5.6 and 5.7. However, unsatisfactory positioning accuracy is observed in the grey area of these figures, attributed to excessive signal reflections. Consequently, integrating additional onboard sensors, such as an IMU [132], is essential for improving state estimation performance.

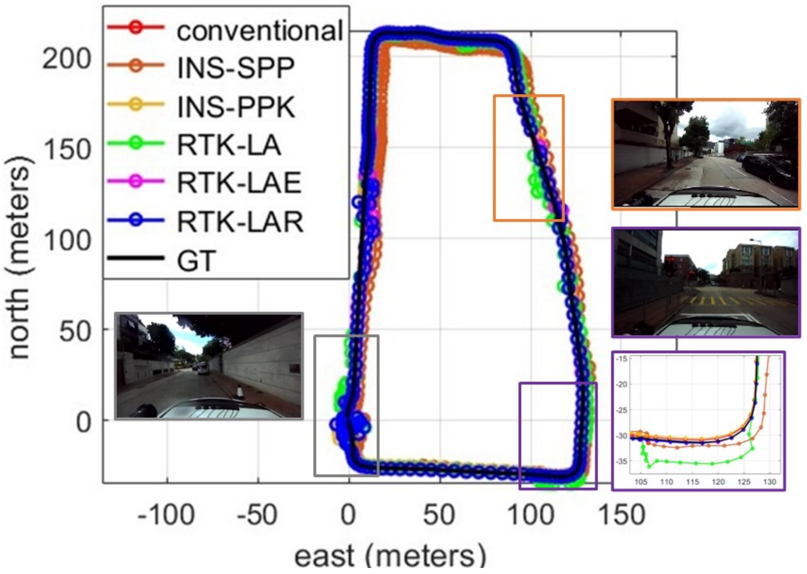


Figure 5.6: 2D positioning trajectories of the listed methods. The x-axis and y-axis represent the east and north directions, respectively. The black curve indicates the ground truth positioning. The areas marked in grey, purple, and orange correspond to three typical urban environments.

5.4 Summary

Cycle slips pose a significant challenge in GNSS-RTK systems, especially in urban environments where signal reflections are common. When cycle slips go undetected, they can adversely affect both the fixing rate and positioning accuracy, resulting in unreliable navigation solutions. Consequently, effective cycle slip detection is crucial for enhancing the overall performance of GNSS-RTK systems.

In this chapter, we propose a novel cycle slip detection method that leverages predicted triple-differenced range measurements obtained from a LiDAR sensor. This innovative approach shows promising results, achieving a satisfactory fixing rate and accurate state estimation in urban areas. By effectively addressing the issue of cycle slips, our method improves the reliability and robustness of GNSS-RTK systems, particularly in challenging environments.

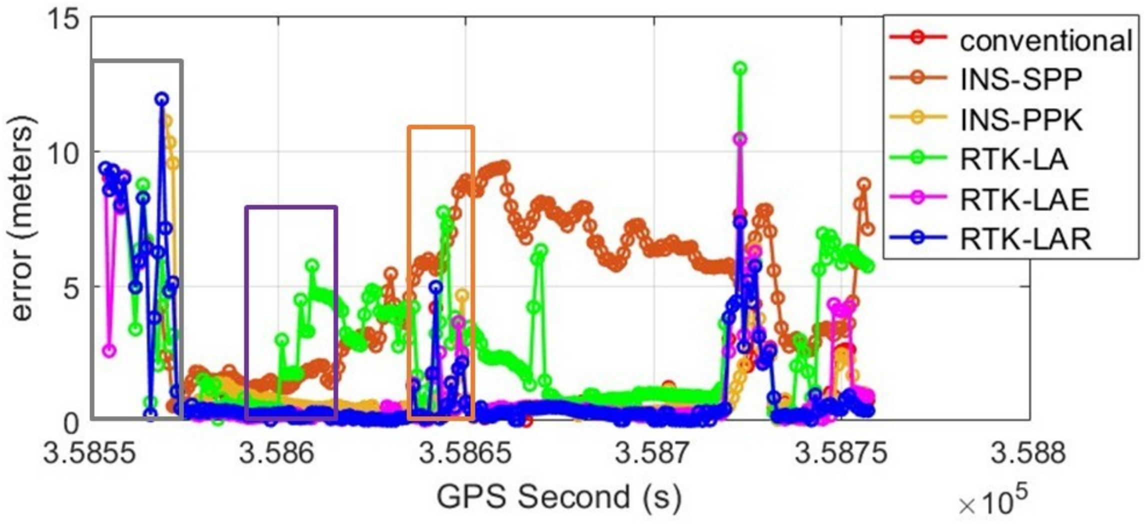


Figure 5.7: 2D positioning errors of the listed methods. The x-axis and y-axis denote the epoch and 2D error, respectively. The areas marked in grey, purple, and orange represent the corresponding area in Fig. 5.6, respectively.

Chapter 6

Roadside Infrastructure Assisted Navigation in Urban areas

6.1 Introduction

With the rapid advancement of RSU and 5G V2X technology, vehicle-infrastructure cooperation has garnered significant attention for its potential to enhance sensing capabilities, supporting various downstream applications for autonomous driving. Fig. 6.1 illustrates the proposed Vehicle-to-Infrastructure (V2I) system concept. Intelligent street poles are equipped with RSU devices, edge sensors, and a Multi-Lamp Poles Sensor Fusion Server (MLP-SFS), while the multi-sensory vehicle platform features an OBU for V2X communication. Roadside infrastructure plays a crucial role in smart cities and will be deployed at scale to facilitate V2X cooperation, which is outlined in the smart mobility roadmap for Hong Kong [32], announced by the Hong Kong Transport Department. Inspired by this initiative, we aim to investigate the potential of roadside infrastructure to enhance navigation performance in urban areas. Firstly, we explore the use of roadside LiDAR to provide accurate state information, which serves as a global constraint in LIO graph-based optimization. Secondly, we incorporate consistent DD observations provided by roadside GNSS, which are jointly optimized within the factor graph framework. Finally, we introduce an error-map-aided multi-sensor integrated system that leverages error information collected by sensor-rich autonomous vehicles. This error information is uploaded to the roadside infrastructure and subsequently distributed to autonomous vehicles.

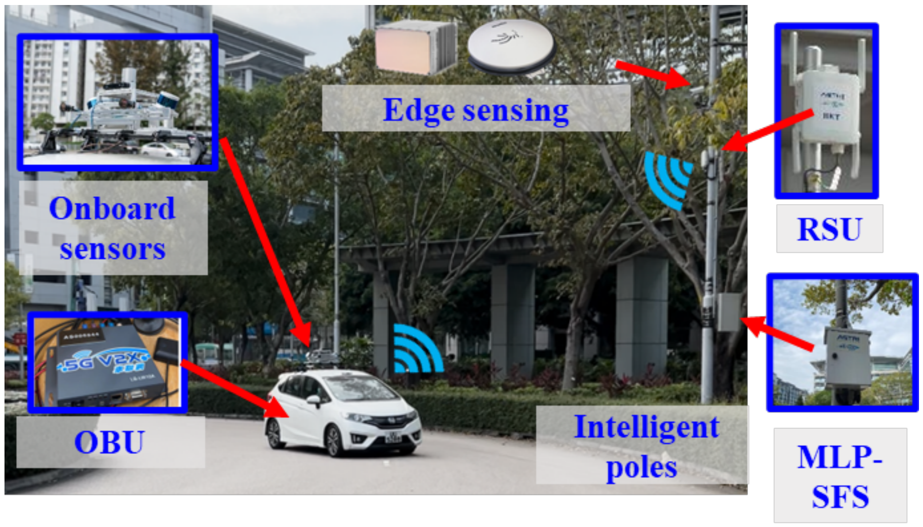


Figure 6.1: Illustration of the Vehicle-Infrastructure Cooperation System in Hong Kong C-V2X testbed. The multi-sensory vehicle platform is equipped with an OBU that communicates with the RSU through a 5G V2X network operating in the 5905-5925 MHz frequency range. The Multi-Lamp Poles Sensor Fusion Server (MLP-SFS) processes edge sensing data in real-time. Sensing information is shared between vehicles and infrastructure over a low-latency network. The communication range for RSUs is 500 meters, while for OBUs, it is 200 meters. The direct C-V2X (PC5) communication delay is less than 20 ms.

6.2 Roadside Infrastructure assisted LiDAR/Inertial-based Mapping for Intelligent Vehicles in Urban Areas

To mitigate error accumulation in LiDAR-based odometry, numerous methods [11; 9; 133] have been proposed to correct accumulated drift. One common approach is to loosely integrate GNSS with LIO [134] using factor graph optimization to provide global pose constraints. However, the effectiveness of such methods is difficult to guarantee in urban areas, where GNSS performance is significantly degraded due to NLOS conditions and multipath effects [25]. The study in [135] explored loop closure factor constraints to reduce drift accumulation. Unfortunately, revisiting routes is not always available for mapping tasks in urban environments. Another research team [136] proposed using globally consistent LiDAR matching to minimize residuals between frames across the entire map. However, this approach is still impacted by unexpected outliers, such as dynamic objects. In summary, existing LiDAR-based mapping and localization solutions relying on onboard sensors from a single vehicle face challenges in complex urban scenarios. To address the limitations of LiDAR-based mapping using onboard sensors, this work proposes an RSI-assisted LIO method for urban mapping and localization. This approach leverages global constraints provided by RSI based on point registration between the ego-vehicle and RSU point clouds. The drift in the LIO is corrected by the registered absolute positioning with the assistance of the RSIs. To evaluate the performance of our proposed method, we collected multi-view RSIs and vehicle sensor data in the Hong Kong C-V2X testbed [137]. The contributions of this work are highlighted as follows:

1. We present the deep learning-based vehicle detection framework using multi-RSI LiDARs, which provides an accurate initial guess for aligning the ego-vehicle and RSI LiDAR clouds.

2. We propose an RSI-assisted LIO to reduce global drift using the global constraint provided by the point cloud registration between the vehicle and RSI.

3. We extensively validate the effectiveness of the proposed method using the real-world data collected in the Hong Kong C-V2X testbed. The results demonstrate that the positioning performance is significantly improved aided by the RSI in challenging urban areas. We also open-sourced our data at `https://github.com/DarrenWong/RSI-aided_LIO` to benefit the research society.

6.2.1 Method

Overview of the Proposed Architecture

An overview of the proposed system is illustrated in Fig. 6.2. The system comprises two main components: (1) vehicle detection and tracking using multi-RSI point clouds, and (2) LIO aided by global constraints from RSIs. The system inputs include the vehicle's LiDAR point cloud, IMU measurements, and roadside LiDAR point clouds collected from multiple RSUs. First, the Cloud Fusion module integrates point clouds from multiple RSIs to generate a unified representation. These fused RSI point clouds are then used for vehicle detection and tracking. The detected vehicle poses serve as an initial guess for aligning the onboard local map with the multi-RSI point clouds, as described in the Point Cloud Registration module. Finally, the registered vehicle pose is utilized as an absolute constraint in factor graph optimization, ensuring accurate positioning.

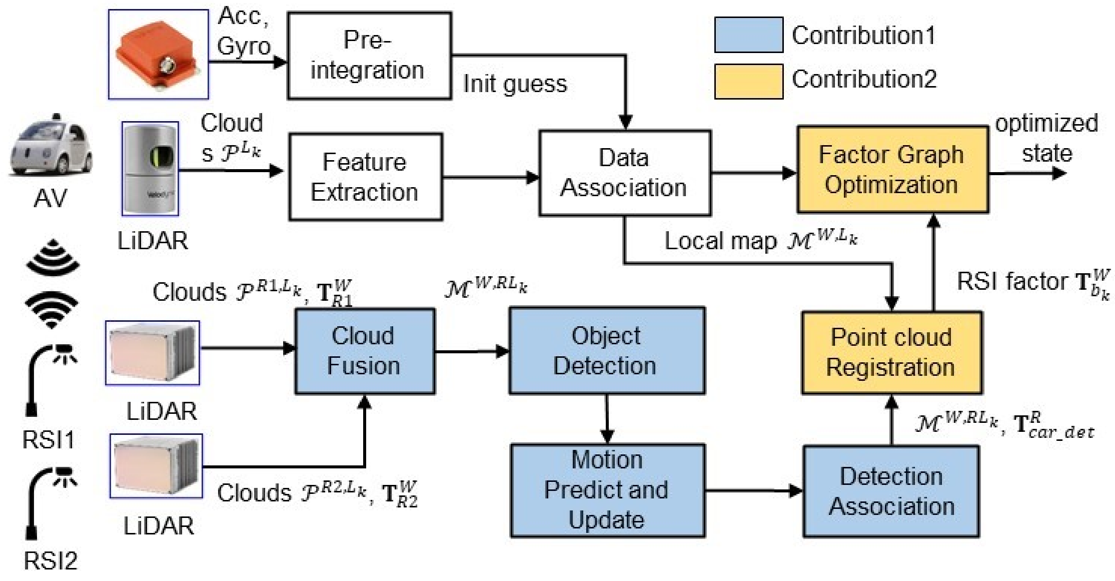

Acc, Gyro
Pre-integration
Init guess
Contribution1
Contribution2
AV
LiDAR
Clouds $\mathcal{P}^{L_k}$
Feature Extraction
Data Association
Factor Graph Optimization
optimized state
Local map $\mathcal{M}^{W,L_k}$
RSI factor $\mathbf{T}^W_{b_k}$
Clouds $\mathcal{P}^{R1,L_k}$, $\mathbf{T}^W_{R1}$
$\mathcal{M}^{W,RL_k}$
Cloud Fusion
Object Detection
Point cloud Registration
RSI1
LiDAR
$\mathcal{M}^{W,RL_k}$, $\mathbf{T}^R_{car\_det}$
Motion Predict and Update
Detection Association
Clouds $\mathcal{P}^{R2,L_k}$, $\mathbf{T}^W_{R2}$
LiDAR
RSI2

Figure 6.2: Overview of the proposed pipeline. $\mathbf{T}^R_{\text{car\_det}}$ indicates object detection in the RSI base frame. $\mathbf{T}^W_{R1}$, $\mathbf{T}^W_{R2}$ and $\mathbf{T}^W_{b_k}$ denote the RSI1 pose, RSI2 pose and vehicle IMU body in the world frame. The data transmission delay between the AV and RSIs is around 20 milliseconds

RSI-based Object Detection and Tracking

To minimize the "blind zones" created by static traffic elements and moving dynamic objects, it is essential to optimally deploy a set of RSIs and LiDAR sensors throughout the area to generate a precise and comprehensive point cloud map of the specific traffic environment. By merging point clouds from different RSI LiDAR systems, a dense map of the area is created, facilitating various downstream tasks related to autonomous vehicles. The initial translation matrix for the multi-RSI LiDARs can be derived by aligning multiple landmark points located within overlapping FOV. The extrinsic matrix between two RSI LiDARs can be further optimized using the ICP method [33].

$$\min_{\mathbf{T}_{R2}^{R1}} \sum_{i=1,\ \mathbf{p}_i^{R1,L_k} \epsilon \mathcal{P}^{R1,L_k}, \mathbf{p}_i^{R2,L_k} \epsilon \mathcal{P}^{R2,\ L_k}}^{N^{R1,2}} \left\| \mathbf{p}_i^{R1,L_k} - \mathbf{T}_{R2}^{R1} \mathbf{p}_i^{R2,L_k} \right\|^2 \tag{6.1}$$

where $\mathbf{T}_{R2}^{R1}$ represents the optimized registration between the point clouds of RSI1 and RSI2. $N^{R1,2}$ indicates the total number of corresponding points from the RSIs. $\mathcal{P}^{L_k}$ and $\mathcal{P}^{R,L_k}$ denote the k-th frame of the vehicle's LiDAR point cloud and the RSI LiDAR point cloud, respectively. The i-th points in $\mathcal{P}^{L_k}$ and $\mathcal{P}^{R,L_k}$ are represented as $\mathbf{p_i}^{L_k}$ and $\mathbf{p_i}^{R,L_k}$. The merged RSI point cloud $\mathcal{M}^{R,RL_k}$ in the RSI base frame can then be computed as follows:

$$\mathcal{M}^{R,RL_k} = \mathbf{T}_{R2}^{R1} \mathcal{P}^{R2,Lk} + \mathcal{P}^{R1,Lk} \tag{6.2}$$

Fig. 6.3 illustrates the cloud fusion results in the base frame using the optimal extrinsic parameters. It can be observed that the points corresponding to the road curb (Fig. 6.3 (A)) and the roundabout (Fig. 6.3 (B)) are partially covered by different RSI LiDAR systems, resulting in a single coherent model. With the pre-calibrated RSI poses in the world coordinate system, the merged cloud or individual

RSI cloud can be transformed into the global frame.

$$\mathcal{M}^{W,RL_k} = \mathbf{T}_R^W \mathcal{M}^{R,RL_k} \tag{6.3}$$

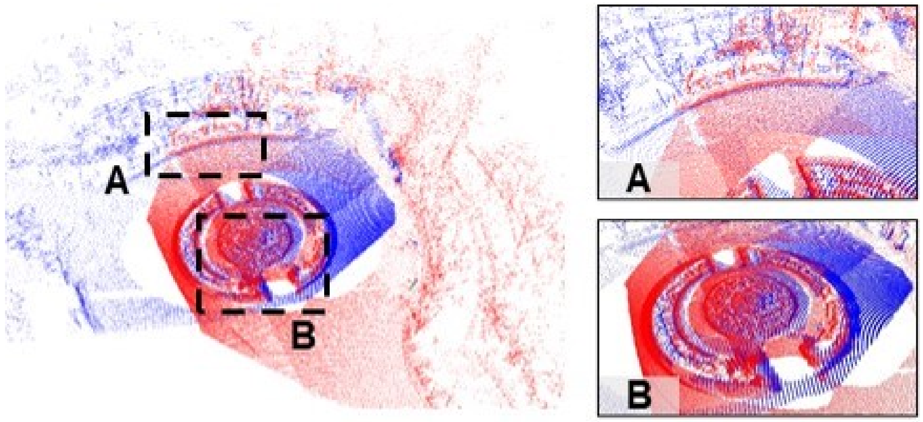


Figure 6.3: Illustration of the point cloud from multi-RSI LiDAR. The cloud is merged by transforming the data from RSI1 (blue) and RSI2 (red) into the base frame. The highlighted areas indicate that the point clouds of the road curb (A) and roundabout (B) are well-aligned.

Object detection and tracking algorithm is then applied on the fused point cloud $\mathcal{M}^{R,RL_k}$. The tracking pipeline necessitates that objects be detected first, with the detection results serving as inputs for the tracking algorithm.

3D object detection: The CenterPoint algorithm [138] was selected for object detection using RSI LiDAR due to its efficiency and effectiveness. This two-stage detector first identifies the center of each object and subsequently estimates attributes such as the size and rotation of the 3D bounding box. Additional point features are then utilized to refine these initial estimations. The detector exhibits robust performance in identifying and localizing objects in 3D space, as demonstrated in the Waymo Open Dataset [139] and our customized RSI data.

3D object tracking: 3D object tracking involves monitoring objects across multiple point cloud frames over time, capturing their position, velocity, and other attributes. Each object is tracked using a bounding box with a unique ID. To achieve this, we leverage the Simpletrack algorithm [140], which employs the "tracking-by-detection" concept to maintain tracking performance while reducing computational costs. Simpletrack [140] uses non-maximum suppression (NMS) to eliminate extraneous boxes and retain valid boxes with low scores after processing the object detection bounding boxes. Generalized IoU (GIoU) [141] serves as the association metric, and the Hungarian algorithm [142] is utilized to match bounding boxes. The Kalman Filter is then applied to estimate the motion of the bounding boxes in subsequent frames. To manage the lifecycle of tracklets [141], a tracklet is initiated after three consecutive matches and terminated after two misses. Finally, the specific ego-vehicle can be identified through RSI with object tracking and V2X communication.

The factor graph of the proposed RSI-assisted LIO is illustrated in Fig. 6.4. It is important to note that the RSI factor acts as a unary constraint within the state. The process of RSI-assisted odometry comprises two main components: point cloud registration aided by object tracking and a factor graph that incorporates the RSI factor.

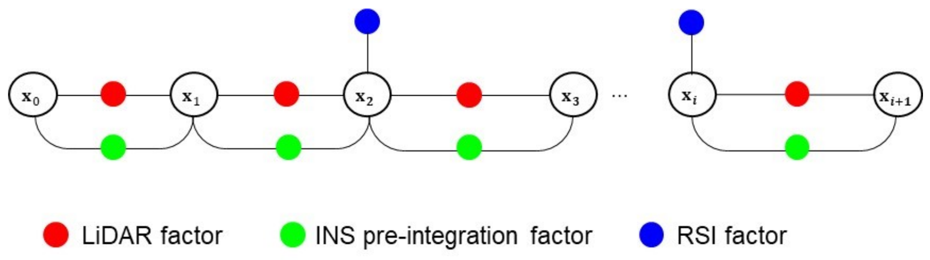


Figure 6.4: The graph of the RSI-assisted LIO. x_i denotes the estimated state.

The accumulated vehicle-side local map $\mathcal{M}^{W,L_k}$ at timestamp k can be expressed as,

$$\mathcal{M}^{W,L_k} = \sum_{s=1}^{N^s} \mathbf{T}_{L_s}^{W} \mathcal{P}^{L_k,s} \tag{6.4}$$

where N^s denotes the extracted N surrounding keyframes. $\mathbf{T}_{L_s}^{W}$ and $\mathcal{P}^{L_k,s}$ represent the pose and LiDAR point cloud of the corresponding keyframe s, respectively.

The vehicle local point cloud map $\mathcal{M}^{W,L_k}$ can be transformed into the RSI frame $\mathcal{M}^{R,L_k}$ based on the vehicle tracking pose $\mathbf{T}^R_{V\_det}$ and the $\mathbf{T}^W_L$,

$$\mathcal{M}^{R,L_k} = \mathbf{T}^R_{V\_det}\mathbf{T}^V_L\left(\mathbf{T}^W_{L_k}\right)^{-1}\mathcal{M}^{W,L_k} \tag{6.5}$$

where $\mathbf{T}^V_L$ indicates the extrinsic parameter of the vehicle center and its LiDAR. The point clouds of the vehicle and RSI can be further aligned using point-based ICP [33] after removing the ground points,

$$\min_{\mathbf{T}^R_{L_k}} \sum_{i=1}^{N^{RL}} \left\| \mathbf{p}_i^{R,L_k} - \mathbf{T}^R_{L_k}\mathbf{p}_i^{RL,L_k} \right\|^2 \tag{6.6}$$

where $\mathbf{T}^R_{L_k}$ is the optimized alignment between RSI and vehicle LiDAR point clouds at timestamp k. N^{RL}, $\mathbf{p}_i^{R,L_k}$, and $\mathbf{p}_i^{RL,L_k}$ indicate the total number of corresponding points, points in merged RSI cloud $\mathcal{M}^{R,RL_k}$, and merged vehicle cloud $\mathcal{M}^{R,L_k}$, respectively. The details of the point cloud registration process are described in Algorithm 3.

Algorithm 3: Vehicle-RSI Point Cloud Registration

```
1: **Inputs:** multi-RSI clouds $\mathcal{M}^{R,RL_k}$, vehicle local map $\mathcal{M}^{W,L_k}$
2: **Outputs:** $\mathbf{T}^R_{L_k}$
3: **Step 1:** Transform the $\mathcal{M}^{W,L_k}$ from the world frame to the RSI frame $\mathcal{M}^{R,L_k}$
   based on the detection information $\mathbf{T}^R_{V\_det}$ using Equation 6.5.
4: **Step 2:** Remove the ground points of the $\mathcal{M}^{R,RL_k}$ and $\mathcal{M}^{R,L_k}$ based on the
   ground height.
5: **Step 3:** Search the corresponding points between $\mathcal{M}^W_{L_k}$ and $\mathcal{M}^R_{L_k}$ based on the
   kd-tree.
6: **Step 4:** Iteratively compute the optimized $\mathbf{T}^R_{L_k}$ by minimizing the residuals
   between the clouds from the vehicle and RSI using Equation 6.6.
```

The output of Algorithm 3 is the registered vehicle pose in the RSI frame. With the pre-calibrated RSI position $\mathbf{T}^W_R$ in the world frame, the registered accurate vehicle pose $\hat{\mathbf{T}}_{b_k}^{W}$ can be further transformed into the world frame to form the RSI factor,

$$\hat{\mathbf{T}}_{b_k}^{W} = \mathbf{T}_{b}^{L}\mathbf{T}_{R}^{W}\mathbf{T}_{L_k}^{R} \tag{6.7}$$

where $\mathbf{T}_{b}^{L}$ indicates the extrinsic parameter of LiDAR and IMU. Finally, the cost function of the proposed method aided by the RSI factor can be expressed as,

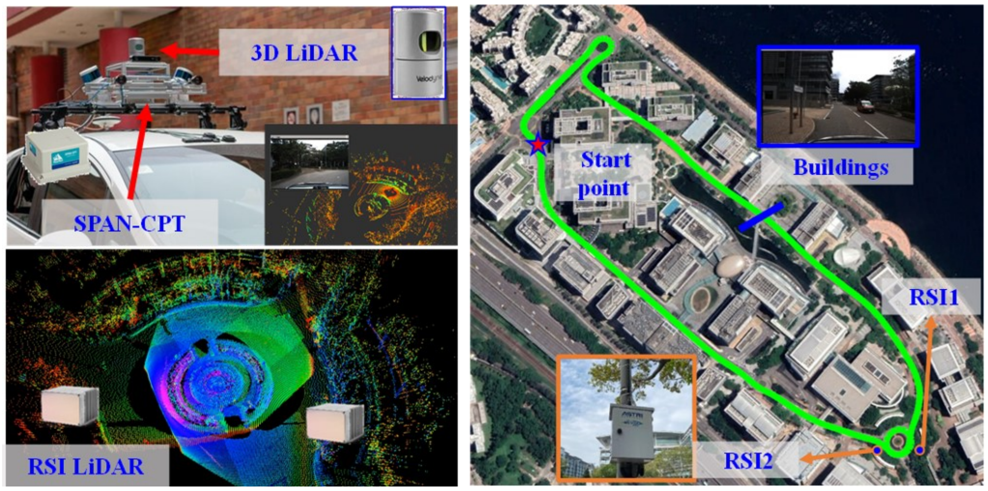
$$\min_{\mathbf{T}_{b_k}^{W}} \left(\left\| \mathbf{r}_L \left(\mathbf{T}_{b_k}^{W} \right) \right\|_{P_L}^{2} + \left\| \mathbf{r}_R \left(\hat{\mathbf{T}}_{b_k}^{W} \right) \right\|_{P_R}^{2} \right) \tag{6.8}$$

where $\mathbf{r}_L$, $\mathbf{r}_R$ indicate the residuals of the LIO factor and RSI factor, respectively. $P_{(\cdot)}$ represents the covariance matrix of each term. IMU pre-integration is used to provide the initial guess and aid in point cloud drift. The accumulated drift of the LIO measurement can be corrected effectively with the RSI factor.

6.2.2 Performance Evaluation

Experiment Setup

Sensor Setups: We utilized the UrbanNav [3] vehicle platform to conduct onboard experiments, equipping it with GNSS, INS, cameras, and LiDAR sensors. The NovAtel SPAN-CPT integrates a fiber optic gyroscope and GNSS-RTK to provide GT positioning. Additionally, measurements from the SPAN-CPT are tightly coupled using NovAtel Inertial Explorer to maximize GT accuracy. Each RSI in the Hong Kong testbed [137] is equipped with GNSS, a 300-line LiDAR, high-performance V2X communication, and edge computing capabilities. In this experiment, we collected 10 Hz LiDAR data and 400 Hz IMU data from one vehicle platform, along with 10 Hz LiDAR data from two RSIs, all simultaneously in a challenging urban environment. Fig. 6.5 presents the sensor setup and evaluated trajectory with a total length of 1.85 km. The RSIs are located in the roundabout area (around half) of the trajectory. The evaluated route returns to the starting point to assess the performance of LiDAR loop closure constraints versus the global constraints provided by the RSIs. PPS time synchronization with the GPS source is implemented on our vehicle platform, while Precision Time Protocol (PTP) time synchronization with the GPS source is conducted on the RSI side.



(a) (b)

Figure 6.5: (a) Setup for the vehicle-infrastructure platform, featuring a Velodyne 32-line LiDAR and SPAN-CPT on the vehicle side, along with two Innovusion Jaguar LiDARs integrated with two RSIs. (b) The evaluated scene of the Hong Kong C-V2X testbed, along with its ground truth trajectory.

Evaluation Metrics: The effectiveness of the listed method is evaluated via the ATE to investigate the global accuracy of the pose estimation.

Evaluated Methods: To verify the performance of the proposed algorithms, we use the following methods,

1. **LIO-SAM** [61]: The state-of-the-art LiDAR-inertial odometry.

2. **RSIA-LIO**: The proposed RSI-assisted LiDAR-inertial method. The RSI factor will constrain the state in the factor graph once available.

Experimental Validation in Urban Areas

1) Results of Object Detection by RSIs

To conduct the evaluation, models for our self-collected RSI data were trained using 80% of the available data, while the remaining 20% was reserved for validation. All detection results are measured by the average precision (AP) at the 3D-GIoU [141] threshold of 0.7 for the cars class, with the AP on the validation set calculated using 40 recall positions.

Table 6.1 presents the performance evaluation of CenterPoint [138] on the validation set of our self-collected dataset. The trained model achieved a 3D detection AP of 57.17% and an average orientation similarity (AOS) AP of 90.09%. As shown in Table 6.1, most vehicles were successfully predicted with satisfactory accuracy. Examples of the testing results are illustrated in Fig. 6.6. Furthermore, vehicles can be continuously tracked [140] across different frames, as indicated by their ID numbers in Fig. 6.6.

Table 6.1: Detection results of the CenterPoint on our validation set.

Results	**Self-collected RSI data**
3D detection AP (%)	57.17
AOS AP (heading angle) (%)	90.09

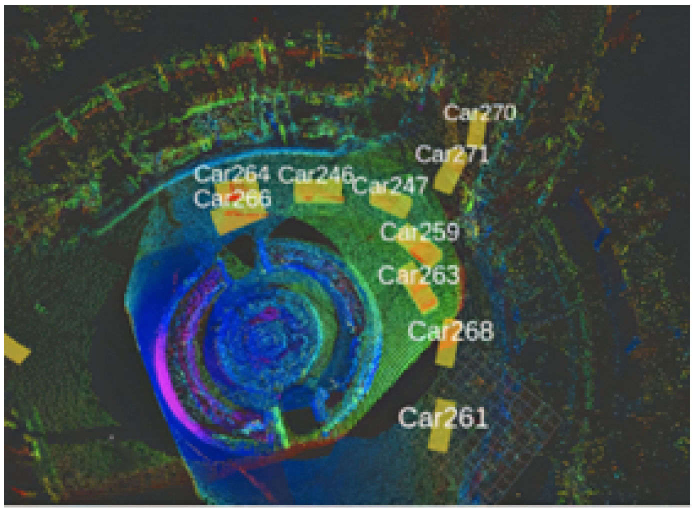


Figure 6.6: Illustration of the RSI detection results.

2) Results of Point Registration Figure 6.7 illustrates the point cloud alignment results between the onboard local map and the multi-RSIs. It can be observed that there is a significant error of several meters (shown in Fig. 6.7(a)) when aligning the point cloud solely based on the detection information provided by the RSUs. However, with the assistance of ICP-based [33] point cloud registration, we successfully achieve high-accuracy alignment of the vehicle-RSI point cloud, as demonstrated in Fig. 6.7(b).

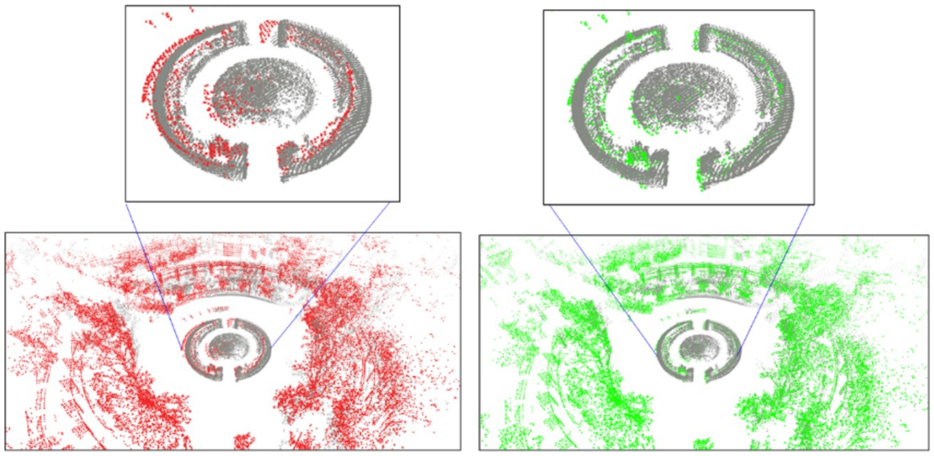

(a) (b)

Figure 6.7: (a) Setup of the vehicle-infrastructure platform, featuring a Velodyne 32-line LiDAR and SPAN-CPT on the vehicle side, along with two Innovusion Jaguar LiDARs integrated into two RSIs. (b) The evaluated scene of the Hong Kong C-V2X testbed, including its ground truth trajectory.

3) Results of RSI-assisted LIO

To validate the contributions of the proposed RSI-assisted LIO methods, experiments were conducted in a challenging urban environment. Table 6.2 presents the positioning performance of the evaluated methods. The existing LIO-SAM achieved an RMSE of 17.004 meters, with a standard deviation of 10.784 meters. When loop closure was enabled, LIO-SAM recorded an RMSE of 18.992 meters, failing to correct the significant drift accumulation from LiDAR odometry in urban areas. In contrast, the proposed RSI-aided method reduced both the RMSE and standard deviation to 5.614 meters and 4.318 meters, respectively. Notably, the RSIA-LIO method with loop closure outperformed all other methods, achieving an RMSE of 2.676 meters. The combination of the LiDAR loop closure constraint and the global constraint from the RSI provides complementary benefits for reliable odometry and mapping in urban environments.

Figs 6.8 and 6.9 illustrate the positioning errors and trajectories using the evaluated methods, while the mapping results of the proposed method are presented in Fig. 6.10. It is evident that the positioning accuracy of LIO-SAM does not benefit from the loop closure factor in the grey area of Fig. 6.8. This is likely because it can only reduce positioning errors at revisit points, but it introduces additional errors in the estimated trajectory due to significant drift accumulation. After applying the RSI global constraint, positioning performance improved by 67% in urban areas, particularly in the yellow area highlighted in Figs.6.8 and 6.9. Notably, the error of the RSIA-LIO method can be further reduced by incorporating the loop closure factor at the revisit point (around epoch 250). The RMSE decreased from 5.614 meters (RSIA-LIO) to 2.676 meters (RSIA-LIO with loop closure), reflecting an improvement of 84.3% compared to the existing LIO method. Additionally, Fig. 6.10 shows that the mapping result using the proposed method aligns well with Google Maps. However, unexpected inaccuracies are still observed around epoch 200 (shown in Fig. 6.8) due to drift accumulation. Therefore, it is crucial to incorporate more RSIs to enhance the performance of state estimation for autonomous driving.

Results	**LIO-SAM**	**LIO-SAM w/ loop**	**RSIA-LIO**	**RSIA-LIO w/ loop**
RMSE (m)	17.004	18.992	5.614	**2.676**
MEAN (m)	13.147	14.077	3.588	**2.392**
STD (m)	10.784	12.748	4.318	**1.200**
Max (m)	33.152	36.177	19.293	**5.781**
Improvement	/	/	67.0%	**84.3%**

Table 6.2: Comparison of LIO-SAM, RSIA-LIO, and their loop variants.

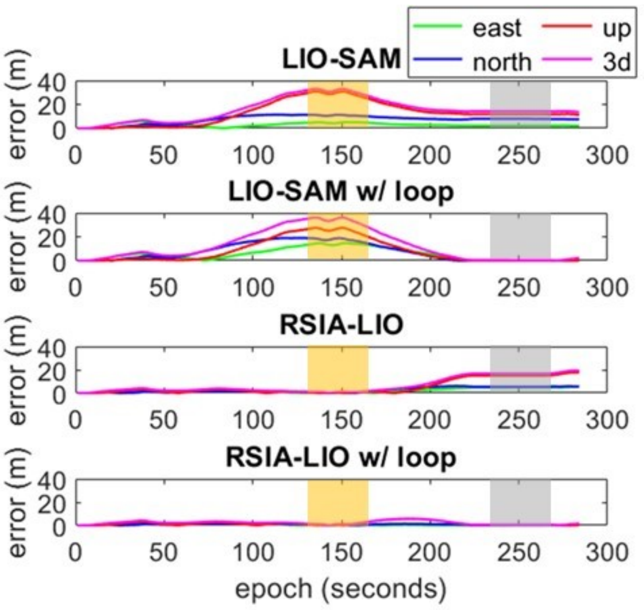


Figure 6.8: Positioning Errors in the ENU Directions. The x-axis represents the timestamp, while the y-axis indicates the errors in meters. The area marked in yellow highlights the errors reduced by the RSI-assisted global constraints, while the area marked in grey represents the loop closure factor when revisiting the starting points.

6.3 Roadside GNSS Aided Multi-Sensor Integrated Positioning for Intelligent Vehicles in Urban Areas

The GNSS, three-dimensional (3D) LiDAR, and IMU are widely used in autonomous systems. GNSS provides an absolute positioning service for navigation applications. However, it suffers from NLOS and multipath effects [25; 143] by tall buildings in urban areas. 3D LiDAR provides dense surrounding point clouds which are widely used for positioning and mapping solutions. IMU captures high-rate acceleration and orientation measurements, which is critical in achieving robust sensor integration. The combination of LiDAR and IMU [40; 61] has been commonly adopted for achieving robust odometry and mapping. However, the positioning performance of such methods can be degenerated by urban canyons such as moving objects [1] and

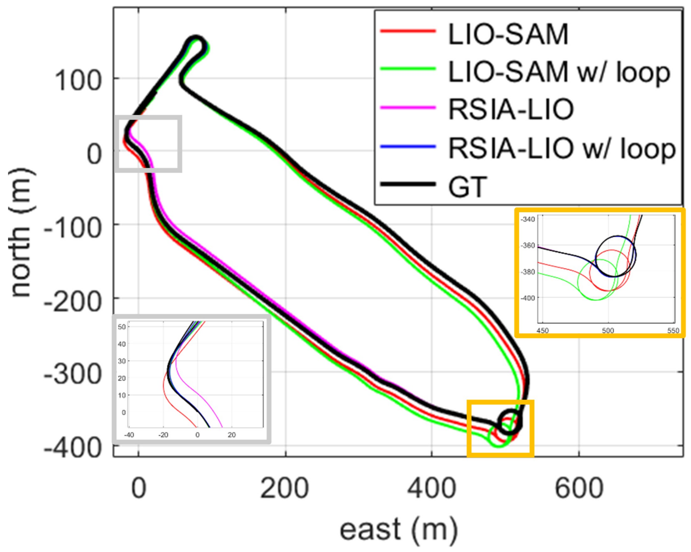


Figure 6.9: 2D positioning of the listed methods in the urban area. The x-axis represents the east direction, while the y-axis indicates the north direction. The areas marked in yellow and grey correspond to the yellow and grey areas in Fig. 6.8, respectively.

featureless environments, which results in accumulated drift for urban navigation. To address the accumulated error of LIO, fusing with GNSS is a promising solution. A common approach is loosely integrating the information from multiple sensors [9] at the positioning level. However such methods are unreliable in urban canyons due to sensor degradation [1] in complex urban. The study [65] proposed a tightly coupled method by accurate outlier exclusion based on residual checks that request a good initial state and healthy measurements which are not always available in dense urban. Our previous work [144] proposes two-stage optimization which integrates the GNSS, LiDAR, and IMU measurements using a graph-based method. However, the unhealthy GNSS is not always mitigated due to continuous NLOS and multipath effects in urban areas. In short, the existing multi-sensor integration relying on onboard sensors is still a significant challenge for navigation in complex urban environments.

In recent years, the significant development in the C-V2X has drawn consider-

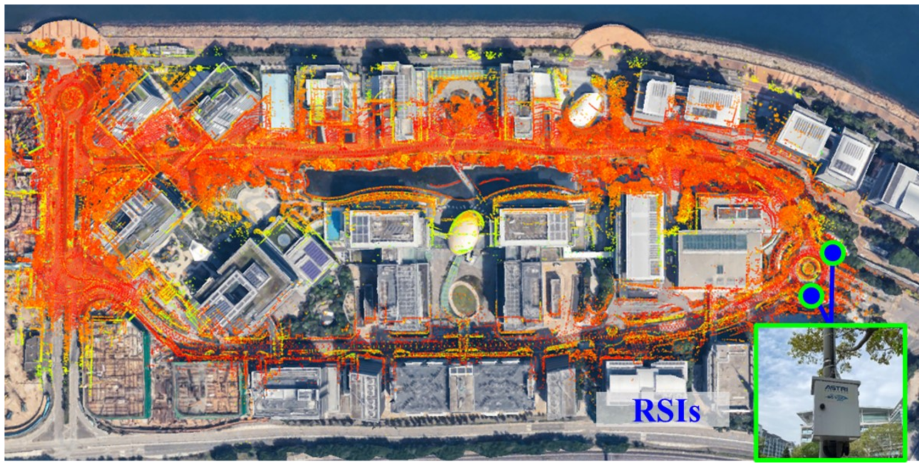


Figure 6.10: Mapping results from the proposed RSI-assisted LIO with loop closure enabled showed high alignment with Google Maps. Note that the ENU orientation of the map has been rotated for better presentation.

able interest because of its capability to enable low-latency information sharing for ITS. According to the CAV Development Study [145] conducted by Deloitte China and ASTRI in April 2024, roadside infrastructure and sensors are fundamental components in the CAV ecosystems. In urban environments, the GNSS measurements in the same region share similar random errors [79] such as NLOS and multipath. Motivated by this, this work proposes a roadside GNSS-aided GNSS/LiDAR/IMU (RSG-GLIO) method for urban navigation, which leverages the accurate DD constraint between the onboard and roadside GNSS. The consistent roadside DD measurements are identified based on the coarse-to-fine residual evaluation. Multiple roadside GNSS and onboard sensor data are collected in the Hong Kong C-V2X testbed [137] to validate the effectiveness of our proposed method, as depicted in Fig. 6.11. The contributions of this study are highlighted as follows:

1. We propose an RSG-GLIO that fuses onboard measurements and roadside GNSS measurements through factor graph optimization.

2. We introduce a coarse-to-fine selection scheme to identify consistent DD measurements from available roadside GNSS measurements. The accurate roadside

DD constraints are jointly optimized into the factor graph optimization.

3. We evaluate the effectiveness of the proposed RSG-GLIO method in the Hong Kong C-V2X testbed. The results report a significant improvement in vehicle positioning performance aided by multiple roadside GNSS in challenging urban areas. Additionally, the potential of roadside GNSS as low-cost base stations in urban areas is explored.

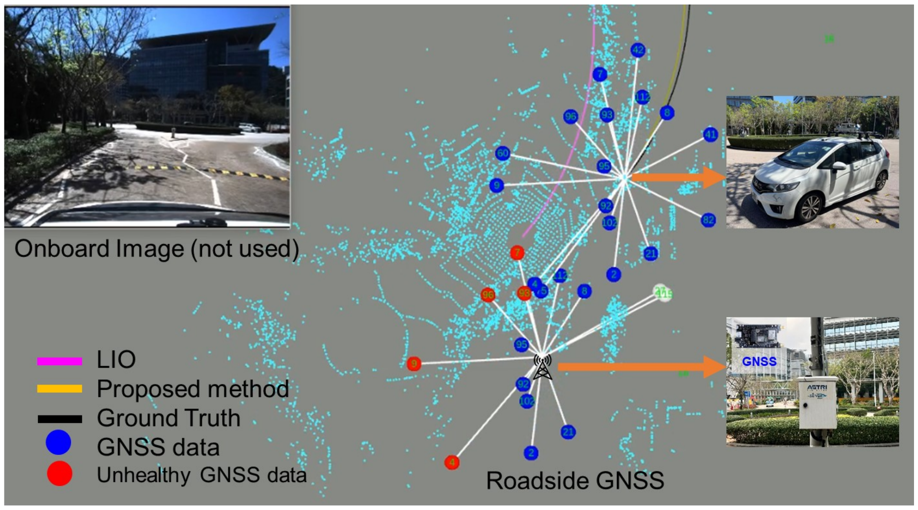


Figure 6.11: Demonstration of the roadside GNSS-aided multi-sensor integrated positioning at Hong Kong C-V2X testbed. The GNSS measurements marked in blue circles are identified as normal GNSS data while the roadside GNSS measurements marked in the red circles are identified as unhealthy GNSS data through coarse-to-fine selection. The roadside unit (RSU) and onboard unit (OBU) have a communication range of 500 meters and 200 meters, respectively.

6.3.1 Method

System Overview

Fig. 6.12 provides an overview of the proposed system. It consists of two main components: (1) Roadside GNSS-assisted GNSS/LiDAR/IMU integration using FGO, and (2) Coarse-to-fine RSG measurement selection. The system takes inputs such as the onboard LiDAR point cloud, IMU measurements, onboard GNSS measurements, and roadside GNSS data from multiple RSI. Firstly, the point clouds, IMU, and GNSS collected by the onboard vehicle are fused to estimate the state. Secondly, the consistent roadside DD factor is identified by the coarse-to-fine selection based on the initial state from onboard state estimation. Finally, the roadside DD factor is incorporated as an additional constraint in FGO for state optimization.

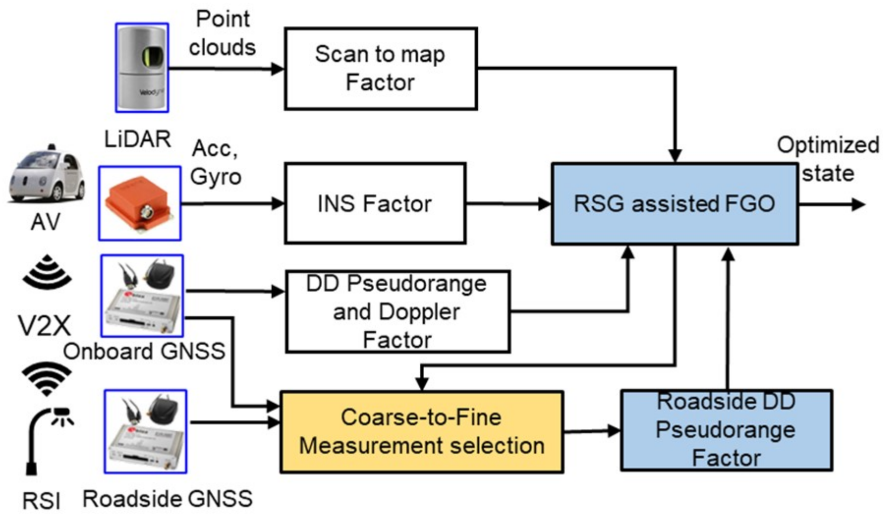


Figure 6.12: Overview of the proposed RSG-GLIO. RSG is short for Roadside GNSS.

GNSS/LiDAR/IMU Integration Aided by RSG

A GNSS pseudorange observation can be defined as [129],

$$\rho_{r,t}^{s} = r_{r,t}^{s} + c\left(\delta_{r,t} - \delta_{r,t}^{s}\right) + I_{r,t}^{s} + T_{r,t}^{s} + \varepsilon_{r,t}^{s} \tag{6.9}$$

where superscript s, subscript r and t denote the satellite, receiver and epoch, respectively. $r_{r,t}^{s}$ is the true range from the satellite to the GNSS receiver. c indicates the velocity of the light. $\delta_{r,t}$ and $\delta_{r,t}^{s}$ denote the clock bias from the receiver and the satellite, respectively. $I_{r,t}^{s}$ and $T_{r,t}^{s}$ denotes the ionospheric delay and tropospheric delay, respectively. $\varepsilon_{r,t}^{s}$ represents the unmodeled noise such as receiver thermal noise and multipath.

Fig. 6.13 illustrates the DD among onboard GNSS, the typical base station $\mathbf{p}_{b,t}^{G}$ and roadside GNSS $\mathbf{p}_{I,t}^{G}$. The single difference (SD) [129] between the GNSS receiver and base station eliminates the satellite clock bias and atmospheric effects. The receiver clock bias can be further mitigated through a between-satellites single difference, also referred to as DD. The DD pseudorange measurement with the base station can be formulated as below,

$$\Delta\nabla\rho_{rb,t}^{s} = \left(\rho_{r,t}^{s} - \rho_{b,t}^{s}\right) - \left(\rho_{r,t}^{w} - \rho_{b,t}^{w}\right) = \Delta\nabla r_{rb,t}^{s} + \Delta\nabla\varepsilon_{rb,t}^{s} \tag{6.10}$$

where subscript b denotes the base station. Satellite w indicates the master satellite, selected based on the highest elevation angle. $\Delta\nabla r_{rb,t}^{s}$ denotes the DD expected range between onboard GNSS and the base station. $\Delta\nabla\varepsilon_{r,t}^{s}$ denotes the noise associated with the DD pseudorange measurements, which are typically meter-level [146] in the open sky. However, the error can be significantly increased to tens of meters due to multipath and NLOS in urban areas. The roadside GNSS and the onboard GNSS have a similar environment (within 100-200 meters V2X communication range), the GNSS atmospheric noise, multipath, and NLOS in the DD pseudorange then can be further mitigated by,

$$\Delta\nabla\rho_{rI,t}^{s} = \left(\rho_{r,t}^{s} - \rho_{I,t}^{s}\right) - \left(\rho_{r,t}^{w} - \rho_{I,t}^{w}\right) = \Delta\nabla r_{rI,t}^{s} + \Delta\nabla\varepsilon_{rI,t}^{s} \tag{6.11}$$

where subscript I denotes the roadside GNSS. $\Delta\nabla r_{rI,t}^{s}$ and $\Delta\nabla\varepsilon_{r,t}^{I}$ are the DD expected range and DD noise between the onboard GNSS and the roadside GNSS, respectively, which is free of atmospheric delay, multipath, and NLOS effects [79] as nearby roadside GNSS share similar environmental patterns.

Fig. 6.13. Illustration of double-differenced GNSS measurements among onboard GNSS $\mathbf{P}_{r,t}^{G}$, roadside GNSS $\mathbf{p}_{I,t}^{G}$ and base station $\mathbf{p}_{b,t}^{G}$.

The residual of roadside DD pseudorange can be further derived based on Equation 6.11,

$$r_{DD,rI,t}^{s} = \Delta\nabla\rho_{I,t}^{s} - \ \Delta\nabla r_{rI,t}^{s} \tag{6.12}$$

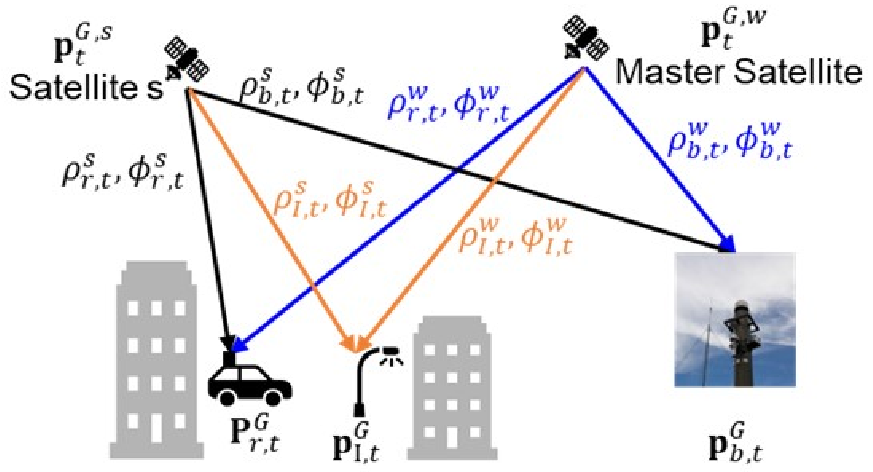

$\mathbf{p}_t^{G,s}$
Satellite s
$\mathbf{p}_t^{G,w}$
Master Satellite
$\rho_{b,t}^s, \phi_{b,t}^s$
$\rho_{r,t}^w, \phi_{r,t}^w$
$\rho_{b,t}^w, \phi_{b,t}^w$
$\rho_{r,t}^s, \phi_{r,t}^s$
$\rho_{I,t}^s, \phi_{I,t}^s$
$\rho_{I,t}^w, \phi_{I,t}^w$
$\mathbf{P}_{r,t}^G$
$\mathbf{p}_{I,t}^G$
$\mathbf{p}_{b,t}^G$

Figure 6.13: Illustration of double-differenced GNSS measurements among onboard GNSS $\mathbf{P}_{r,t}^G$, roadside GNSS $\mathbf{p}_{I,t}^G$ and base station $\mathbf{p}_{b,t}^G$.

To facilitate residual calculation, the state is converted from the ENU frame to the ECEF frame, as the state is maintained in the ENU frame. Conversion between ENU and ECEF can be referred to [15] in detail.

Doppler measurements can be used to determine the receiver's velocity and aid in precise positioning calculation. A Doppler measurement can be modeled as [129],

$$r_{r,t}^{s,D} = \lambda d_{r,t}^s - \quad rr_{r,t}^s \tag{6.13}$$

where λ is the wavelength. $rr_{r,t}^s$ indicates the expected range rate of the ego GNSS receiver.

The inertial sensors provide high-frequency acceleration and angular velocity for efficient sensor fusion as it is immunized to environmental features. Normally we pre-integrate multiple raw measurements [62] to establish the relative constraint between keyframes. Details of IMU pre-integration can be referred to [62] for more information.

Scan to map scheme [18] is adapted to process the point cloud data from the LiDAR sensors. The scan-to-map factor can provide locally accurate pose estimation in urban canyons, the residual from planar feature $\mathbf{F}_{p,\ k,i}^L$ to the fitted plane can be represented as,

$$r_{L,t}^{s2m} = \frac{\left| \left(\mathbf{F}_{p,k,i}^{L} - \mathbf{F}_{p,k,a}^{L}\right) \cdot \left(\mathbf{F}_{p,k,a}^{L} - \mathbf{F}_{p,k,b}^{L}\right) \times \left(\mathbf{F}_{p,k,a}^{L} - \mathbf{F}_{p,k,c}^{L}\right) \right|}{\left|\left(\mathbf{F}_{p,k,a}^{L} - \mathbf{F}_{p,k,b}^{L}\right) \times \left(\mathbf{F}_{p,k,a}^{L} - \mathbf{F}_{p,k,c}^{L}\right)\right|} \tag{6.14}$$

where $\mathbf{F}_{p,k,a}^{L}, \mathbf{F}_{p,k,b}^{L},\ \mathbf{F}_{p,k,c}^{L}$ denotes the three nearest planar patches in the keyframe k among planar points in the local map. Then, it can be further transformed from LiDAR to IMU frame $r_{b,t}^{s2m}$ by applying the pre-calibrated extrinsic $\mathbf{T}_{l}^{b}$.

Fig. 6.14 illustrates the factor graph of the RSG-GLIO. The process of the RSG-GLIO includes two parts, a factor graph incorporating the roadside DD pseudorange, and a coarse-to-fine RSG measurement selection to ensure only consistent roadside DD measurement is added. We follow the work in [144] to fuse the IMU factor, LiDAR factor, DD pseudorange, and Doppler factor, plug the innovative roadside DD pseudorange factor into the graph optimization,

$$\begin{aligned} \chi^{*} = \arg\min_{\chi} \sum_{S,r,k,t} & \left(\left\|\mathbf{r}_{p} - \mathbf{H}_{p}\chi\right\|^{2} + \left\|\mathbf{r}_{b,t}^{s2m}\right\|_{\sigma_L}^{2} + \left\|\mathbf{r}_{b,k}\right\|_{\sigma_b}^{2} \right. \\ & \left. + \left\|\mathbf{r}_{r,t}^{s,D}\right\|_{\sigma_d}^{2} + \left\|\mathbf{r}_{DD,rb,t}^{s}\right\|_{\sigma_{\rho b}}^{2} + \left\|\mathbf{r}_{DD,rI,t}^{s}\right\|_{\sigma_{\rho I}}^{2}\right) \end{aligned} \tag{6.15}$$

where $\mathbf{r}_{p}$, $\mathbf{r}_{b,t}^{s2m}$, $\mathbf{r}_{b,k}$,$\mathbf{r}_{r,t}^{s,D}$, $\mathbf{r}_{DD,rb,t}^{s}$, $\mathbf{r}_{DD,\ rI,t}^{s}$ represent the residuals of the prior factor, LIO factor, IMU factor, Doppler factor, DD pseudorange factor, and roadside DD pseudorange factor, respectively. All the sensor residual is modeled from the corresponding sensor frame to the IMU-centric ENU frame by applying the transformation $\mathbf{T}_{(\bullet)}^{EN}$. $\sigma_{(\bullet)}$ indicates the covariance matrix of each term. An experimental covariance $\sigma_{\rho I} = 0.1 \times \sigma_{\rho b}$ is set to roadside DD pseudorange factor compared to DD pseudorange factor as $\mathbf{r}_{DD,\ rI,t}^{s}$ should be small according to Equation 6.12.

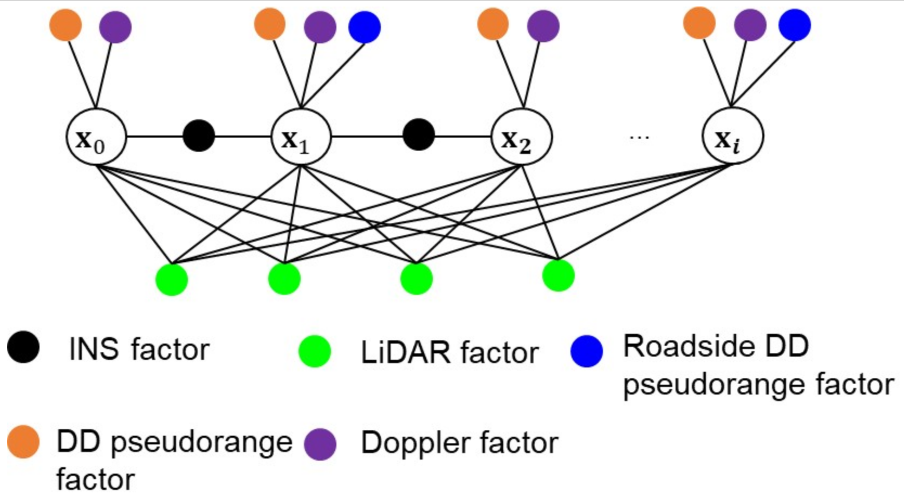


Figure 6.14: The factor graph of the RSG-GLIO. $\mathbf{x}_i$ indicates the state to be estimated at the time i.

Coarse-to-Fine RSG Measurement Selection

The coarse-to-fine roadside GNSS measurement selection is a two-stage approach designed to identify high-quality measurements from roadside GNSS receivers. In the coarse phase, the onboard state first is used to compute the estimated DD range between RSG and onboard GNSS. Given the onboard vehicle state, the estimated geometric distance $\widehat{r}^s_{r,t}$ for satellite s to the onboard receiver is calculated as,

$$\widehat{r}^s_{r,t} = \| \mathbf{P}^s{}_t - \mathbf{P}^G_{r,t}\| = \sqrt{\left(P^s_{t,x} - P^G_{r,t,x}\right)^2 + \left(P^s_{t,y} - P^G_{r,t,y}\right)^2 + \left(P^s_{t,z} - P^G_{r,t,z}\right)^2} \tag{6.16}$$

Therefore, the estimated DD range measurement can be expressed as,

$$\Delta\nabla\widehat{r}^I_{r,t} = \left(\widehat{r}^s_{r,t} - r^s_{I,t}\right) - \left(\widehat{r}^w_{r,t} - r^w_{I,t}\right) \tag{6.17}$$

Recall Equation 6.11, the estimated residual between RSG and onboard GNSS can be obtained,

$$\widehat{\varepsilon}^s_{rI,t} = \Delta\nabla\rho^s_{rI,t} - \Delta\nabla\widehat{r}^I_{r,t} \tag{6.18}$$

$\widehat{\varepsilon}^s_{rI,t}$ is then employed to exclude RSG measurements from the FGO optimization if exceeding an experimentally determined threshold of 1.0 meters. In the fine stage, the RSG DD measurements are iteratively evaluated and selected for inclusion in the FGO similar to the coarse stage. The selected RSG DD measurements are with increasing weighting as more precise RSG DD constraints can be provided compared to other existing DD pseudorange factors. The details of the coarse-to-fine RSG measurement selection are described in Algorithm 4.

Algorithm 4: Coarse-to-Fine RSG Selection

```
1: Inputs: onboard GNSS measurements, RSG measurements, estimated state
2: Outputs: consistent DD RSG measurements
3: Step 1: Calculate the DD RSG residual between RSG and onboard in the
   coarse stage using Equation 6.18.
4: Step 2: Remove the unhealthy DD RSG measurements and the consistent DD
   measurements are added into FGO.
5: Step 3: Iteratively evaluate and select the healthy DD RSG measurements in
   the fine stage.
6: Step 4: Iteratively optimize the state by minimizing the residuals of the
   factors using Equation 6.15 and then repeat Step 3 for accurate DD RSG
   selection until convergence.
```

6.3.2 Performance Evaluation

Experiment Setup

Our UrbanNav vehicle platform [3] is employed for onboard data collection, which includes GNSS, INS, cameras, and LiDARs. NovAtel SPAN-CPT and Inertial Explorer software [107] are used to provide centimeter-level GT positioning. Each RSI in the Hong Kong C-V2X testbed [137] includes GNSS measurements, high-resolution LiDAR, and low-latency V2X communication. Fig. 6.15-(left) presents a static experiment evaluating RTK positioning performance using an urban roadside base station compared to a conventional open-sky base station. In the dynamic environment shown in Fig. 6.15-(right), we collected 1Hz u-blox F9P GNSS data, 10 Hz Velodyne HDL 32-line LiDAR data, 400 Hz Xsens Mti-10 IMU data from our vehicle platform, and 1 Hz u-blox F9P GNSS data from two different RSIs. The pose of the roadside GNSS is computed by Precise Point Positioning (PPP) [147] using 2-hour static data collection to make it convergent. Fig. 6.16 illustrates the vehicle-infrastructure platform and GT trajectory covering a total distance of 1.73 kilometers. Both the roundabout area and the west area (with tall buildings) of the trajectory are with roadside sensors and V2X communication. RSG measurements within 100 meters between RSG and onboard are utilized based on the V2X coverage. Our vehicle platform and RSI sensors are synchronized with PTP and GNSS source.

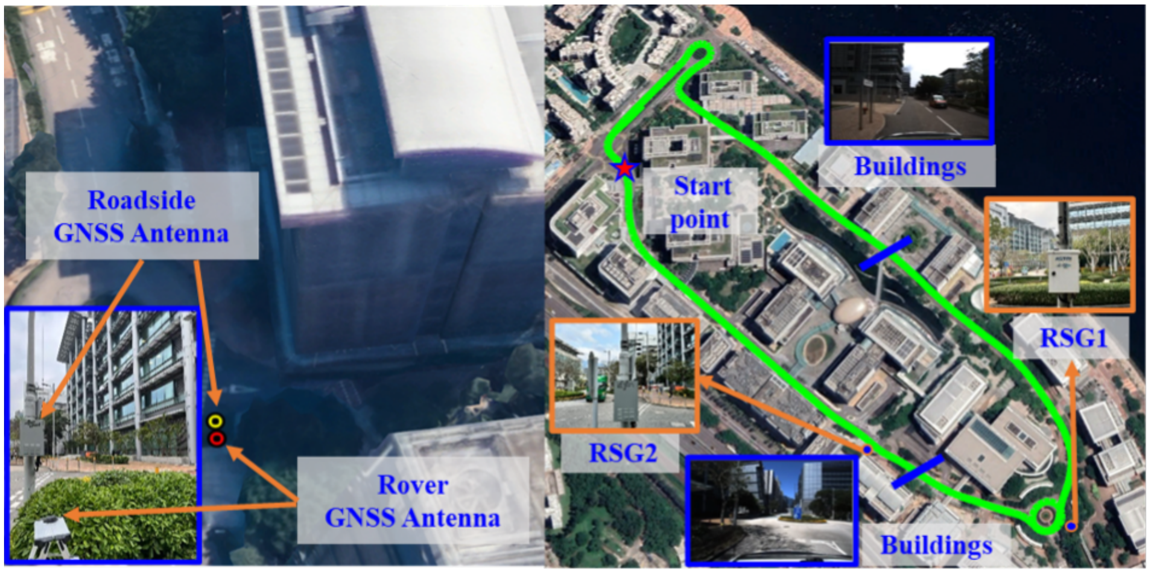


Figure 6.15: (left) The static experiment near a tall building using one RSG and one rover GNSS compared with the traditional open-sky base station (baseline:4.34 km) and one rover GNSS (short baseline: 2.26 m); (right) The dynamic experiment in Hong Kong C-V2X testbed which contains multiple RSGs and the evaluated ground truth trajectory.

To evaluate the performance of the proposed RSG-GLIO, the following methods are employed,

(1) **RTKLIB** [24]**:** The popular GNSS RTK positioning method. Available L1 measurements are used with forward and backward solutions.

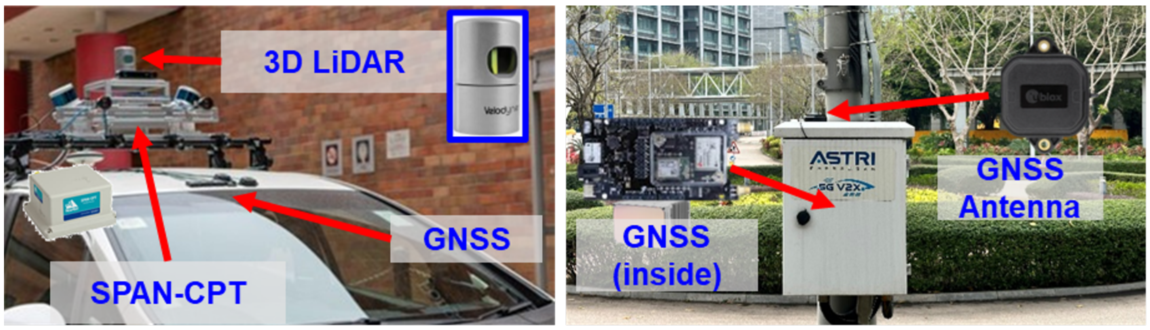


Figure 6.16: (left) Sensor setup for the onboard and RSG platform. A u-blox F9P GNSS, Velodyne 3D LiDAR, Xsens MTi-10 IMU, and SPAN-CPT for the onboard while two u-blox GNSS are connected with different roadside infrastructures; (right) U-blox GNSS is integrated with each roadside infrastructure.

(2) **GICI-Lib** [148] **:** The state-of-the-art FGO-based GNSS RTK positioning method with batch optimization. We excluded the satellite with an elevation angle lower than 15 degrees.

(3) **LIO-SAM** [61]**:** The representative LiDAR-inertial odometry.

(4) **GLIO** [144]**:** The tightly coupled integration system that combines GNSS, LiDAR, and IMU data.

(5) **RSG-GLIO:** The proposed roadside GNSS-aided sensor integration. The selected consistent roadside DD measurements will constrain the state in the graph iteratively. We also compare the performance of using single roadside GNSS and multi-roadside GNSS sensors. Available L1 measurements are used in the optimization.

Experimental Evaluation in C-V2X testbed

A. Static Experimental Evaluation Results

We first conducted a 1-hour static experiment to evaluate the effectiveness of using RSG compared to base stations for RTK positioning in urban areas. Table 6.3 presents the accuracy evaluation results of four positioning methods. For RTKLIB with the traditional base configuration using publicly available Hong Kong Observatory station [149], the RMSE is 19.007 m. For RTKLIB-RSG, using a nearby RSG to replace the base station as the differential reference station, the RMSE significantly decreases to 12.627 m, indicating a significant improvement in positioning accuracy after introducing the RSG which shares similar multipath effects mentioned in 6.3.1. GICI-Lib represents the positioning method based on the FGO-based RTK positioning [148], also employing the combination of a rover and traditional base station, the RMSE is 18.955 m, suggesting that its positioning accuracy fluctuates considerably. Similarly, GICI-Lib-RSG indicates the method that introduces a nearby RSG for replacement on the basis of the aforementioned setup, the RMSE is the lowest at only 3.822 m. It demonstrates that this method exhibits the best performance in both positioning accuracy and stability. We also observe that GICI-LIB diverges compared to the EKF-based method, as shown in Fig. 6.17. This divergence might indicate that the method is unable to handle large residual cases after 1 hour of testing. Overall, using a nearby RSG to replace a traditional reference station can significantly improve the accuracy and stability of the positioning system by mitigating the multipath by DD observations.

Table 6.3: 2D positioning results of listed methods.

Method	**RMSE (m)**	**MEAN (m)**	**STD (m)**	**Max (m)**
RTKLIB	19.007	18.813	2.714	28.366
RTKLIB-RSG	12.627	12.428	2.232	18.520
GICI-Lib	18.955	17.438	7.429	29.824
GICI-Lib-RSG	**3.822**	**3.463**	**1.616**	**14.032**

Fig. 6.17 demonstrates the accuracy performance of four positioning methods. The results clearly demonstrate that users can obtain more accurate location fixes by integrating RSG information. In addition, we present a histogram visualization comparison of the DD residual distribution under two differential GNSS positioning strategies in Fig. 6.18. The method to calculate the residuals of DD observations can

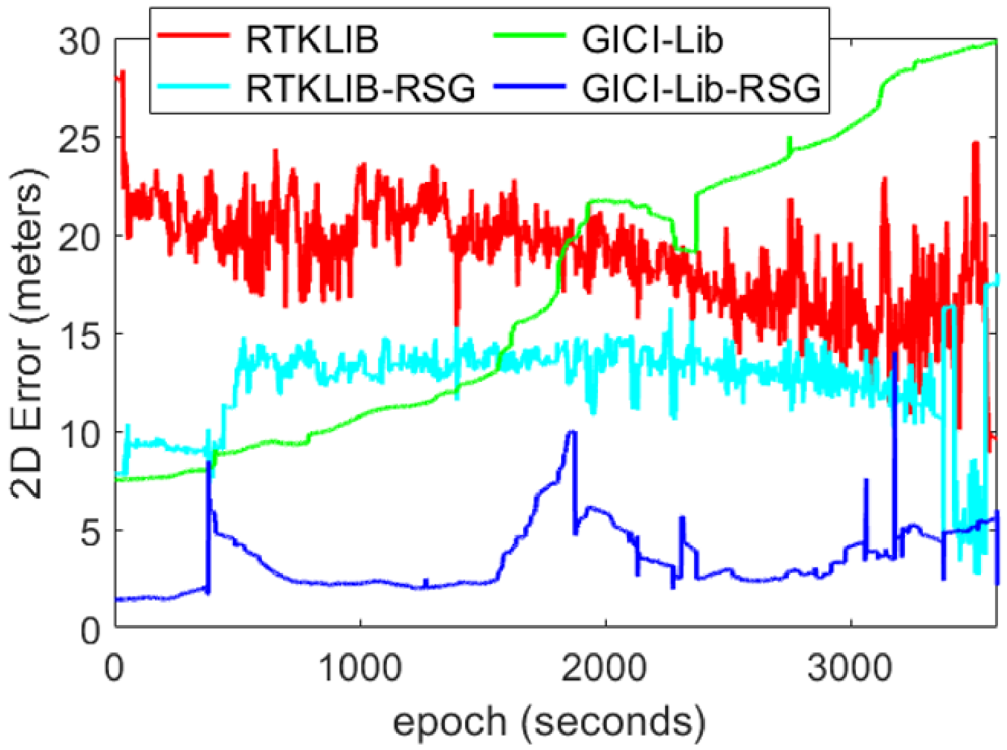


Figure 6.17: 2D Positioning errors of the evaluated methods in the static experiment.

be found in [150]. It can be seen from Fig. 6.18 that the residual distribution of the Rover-RSG methods shows significant concentration: its histogram peak is sharp, and most error values are closely centered around the zero-error line, indicating that most positioning errors are concentrated near the zero value, the distribution range is relatively narrow, and the tail is short. In contrast, the residual distribution of the Rover-Base method is more dispersed, with a gentle peak and a wider distribution range. Especially in the positive error direction, the histogram tail extends longer, indicating that the probability of large error values is significantly increased due to the multipath and NLOS. This intuitive difference in distribution patterns initially reveals the advantages of the Rover-RSG method in error control.

Furthermore, the error metrics in Table 6.4 show that the Rover-RSG is superior to the Rover-Base method in terms of residual. The RMSE of Rover-RSG is 30.494

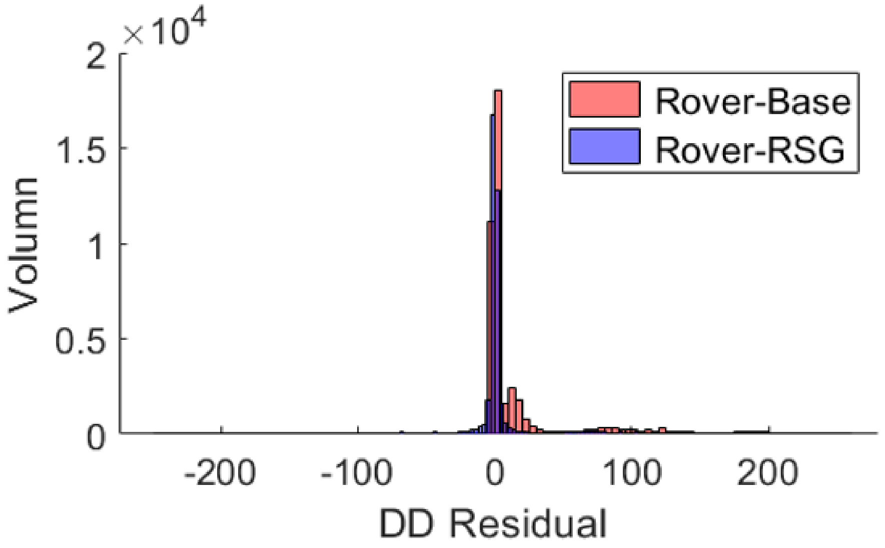


Figure 6.18: Histogram of the DD residual with GT pose. The ground truth of the double-difference pseudorange is obtained by differentially calculating the ground truth and the satellite position, and the double-difference pseudorange errors under the two differential GNSS positioning strategies are evaluated on this basis. This double-difference pseudorange error can directly reflect the unmodeled errors in the measured values.

Table 6.4: Quantitative metrics of DD residual between Rover-RSG and Rover-Base error. CEP95 is short for a circular error probability of 95%.

Method	RMSE (m)	MEAN (m)	STD (m)	CEP95
Rover-RSG	19.007	18.813	2.714	28.366
Rover-Base	12.627	12.428	2.232	18.520

m, which is significantly lower than the 43.674 m of Rover-Base, a decrease of 30%. The STD of Rover-Roadside is 30.156 m, which is lower than the 40.156 m of Rover-Base, indicating that its error distribution is more concentrated. More importantly, the circular error probability of 95% (CEP95) of Rover-RSG is 66.132 m, which is much lower than the 118.132 m of Rover-Base, which indicates that the probability of extreme errors is significantly reduced. The differential GNSS positioning method (Rover-RSG) with the introduction of RSGs shows significant advantages in terms of positioning accuracy, error fluctuation range, and the probability of extreme errors.

Based on the static experimental results of DD observations between user receivers and RSGs, we can conclude that it can eliminate satellite and receiver clock errors while mitigating atmospheric delay, multipath, and NLOS effects. Unlike traditional wide-area reference networks with longer baselines (4.34 km in this paper), RSGs shorten distances, improving error correlation and reducing multipath and NLOS effects. This proximity enhances the effectiveness of double-differencing, suppressing residual errors and significantly improving positioning accuracy, especially in complex urban environments. In summary, RSG, through its significant geographical advantages, not only enhances the efficiency of utilizing error spatial correlation but also provides more precise correction values for DD pseudorange positioning technology, thereby achieving a significant improvement in positioning accuracy in urban environments. In the following section, we investigate the effectiveness of including RSG measurements in our proposed sensor fusion framework.

B. Dynamic Experimental Evaluation in C-V2X Testbed

1) Results of RSG-GLIO

To evaluate the effectiveness of the proposed RSG-GLIO method, experiments were conducted under challenging urban scenarios. The positioning performance using the listed algorithms is summarized in Table 6.5. The 2D RMSE of GNSS RTK is 5.661 meters, with a standard deviation of 2.662 meters. A 2D RMSE of 9.254 meters was resulted using the popular LIO-SAM. The RMSE of the tightly coupled GLIO method was 1.726 meters, which shows improvements in correcting the accumulated by LiDAR odometry using GNSS measurements in urban areas. With the aid of one single roadside GNSS, the RMSE was reduced to 1.245 meters and 1.153 meters aided by RSG1 and RSG2, respectively. Interestingly, the proposed RSG-GLIO with multi-roadside GNSS enabled outperformed other methods in terms of RMSE of 1.095 meters. The utilization of multiple accurate DD roadside GNSS measurements is reliable for odometry and mapping in urban areas.

Fig. 6.19 illustrates the positioning error while Fig. 6.20 presents the trajecto-

Table 6.5: 2D positioning results of listed methods. RSG-GLIO w/rsg1 and RSG-GLIO w/rsg2 denote the RSG-GLIO method using single roadside GNSS RSG1 and RSG2, respectively.

Method	**RMSE (m)**	**MEAN (m)**	**STD (m)**	**Max (m)**
RTKLIB	5.661	4.999	2.662	12.709
LIO-SAM	9.254	7.258	5.741	16.369
GLIO	1.726	1.551	0.758	3.537
RSG-GLIO w/rsg1	1.245	1.137	0.508	3.334
RSG-GLIO w/rsg2	1.153	1.063	0.758	3.537
RSG-GLIO	**1.095**	**1.025**	**0.385**	**2.230**

ries based on the evaluated methods. The positioning performance of GNSS-RTK is affected by the numerous buildings. The error of the LiDAR-based method is accumulated when it comes to the roundabout area which is marked in grey. By tightly coupled GNSS and LiDAR, the positioning performance can achieve a low drift pose estimation using state-of-the-art GLIO methods. After applying the multi-RSG DD constraint, the positioning performance was significantly improved by 36.6% compared to the GLIO.

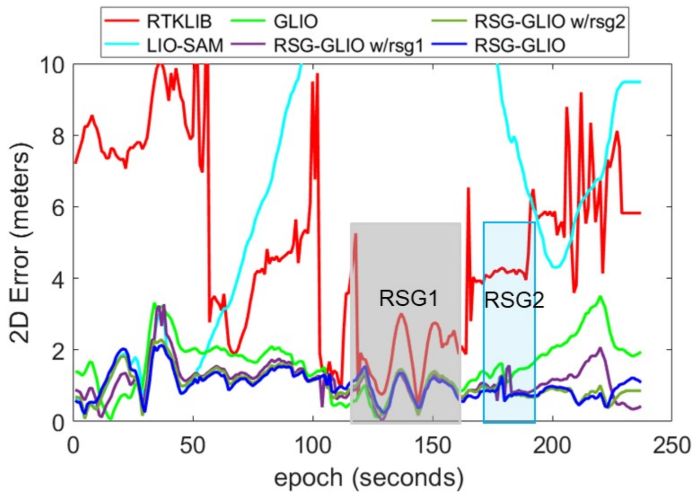


Figure 6.19: 2D Positioning errors of the evaluated methods in the C-V2X testbed. The area marked in grey indicates the error mitigated with the RSG1 while the area marked in blue indicates the error mitigated by the RSG2.

2) Results of Coarse-to-Fine DD Measurement Selection

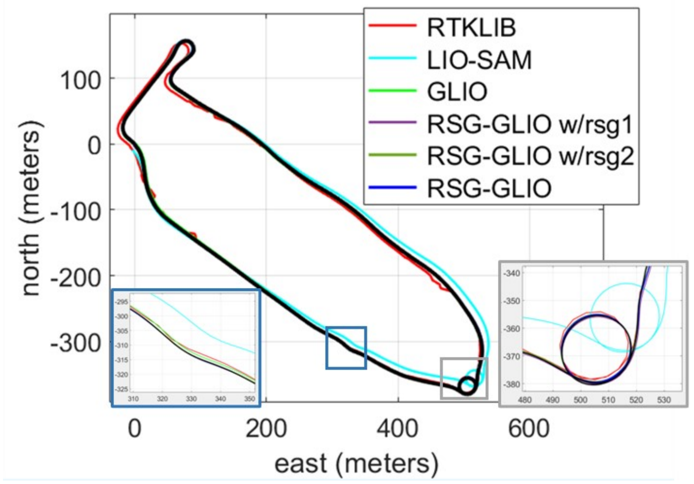


Figure 6.20: 2D positioning of the evaluated methods in the C-V2X testbed. The grey and blue areas correspond to the yellow and grey areas in Fig. 6.19, respectively.

Fig. 6.21 demonstrates the estimated DD residuals and ground truth labeled DD residuals. The initial guess of the state estimation is used to evaluate the DD residual in the coarse stage. It can be seen that our estimated residual is close to the ground truth labeled DD residuals using the coarse-to-fine scheme. We only select the consistent DD measurements and then contribute to the global graph-based optimization. Take PRN G27 as an example (left bottom of Fig. 6.21), the DD residuals are large between epoch 0-5 therefore we excluded the unhealthy DD measurements for the graph-based optimization.

3) Results of Transmission Latency from RSU to OBU The raw measurement of the roadside GNSS is packed in the standard V2X basic safety message (BSM) of 254 bytes per message [151] and broadcast to the OBU which is located in the vehicle platform. The average latency of one message or in one message group sent from RSU and then received by OBU is 21.3 milliseconds. The latency is acceptable [24] for GNSS DD technical with the roadside GNSS measurements.

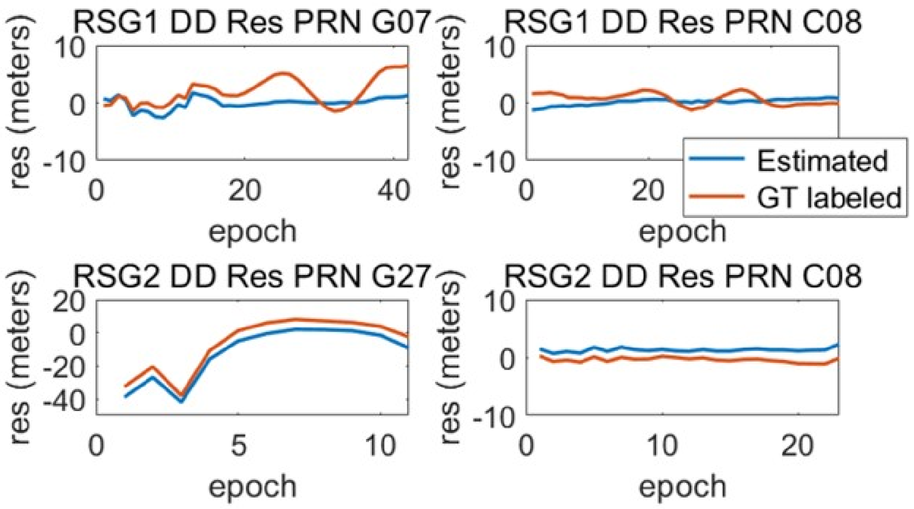


Figure 6.21: DD residual in urban areas. The x-axis indicates the epoch (which might be discontinuous GNSS measurements) while the y-axis represents the DD residuals. RSG1 DD Res PRN indicates the DD residual of PRN number G07 satellite using the RSG1 as the base station.

Applications with Roadside GNSS

As the roadside GNSS is statically installed, it has the potential to serve the RTK correction of the local region since it shares similar NLOS and multipath effects in urban areas. Fig. 6.22 presents the mean error of RTK positioning using the roadside low-cost GNSS is 2.49 meters, which can achieve similar performance compared to using the high-end base station provided by the Hong Kong government (4.3 kilometers away) with the MEAN error is 1.97 meters.

6.4 Continuous Error Map Aided Adaptive Multi-Sensor Integrated Positioning for Connected Autonomous Vehicles in Urban Scenarios

Inspired by the efficient data transmission capabilities of C-V2X technology among CAVs[152], we propose a system for continuous error map estimation and its application in V2X-enabled multi-sensor positioning. The concept behind this system

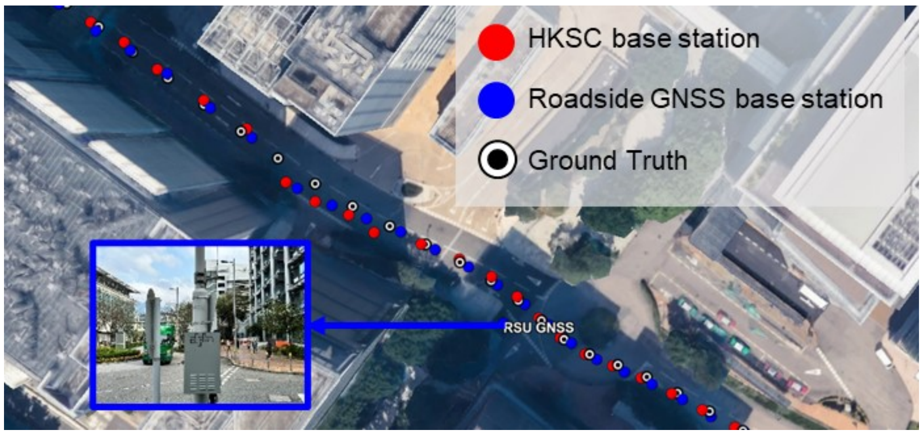


Figure 6.22: RTK positioning results using the HKST base station (nearby station provided by Hong Kong Lands Department) and the proposed roadside GNSS base station.

is illustrated in Fig. 6.23. To create error maps for different sensors in a defined urban area, we utilize the full-sensor-enabled autonomous high-definition (HD) map update [153] vehicle or autonomous bus (CAV1 on the left of Fig. 6.23) periodically collects data and assesses errors. The error data collected by the vehicle is transmitted to a roadside edge system, which integrates this information into a real-time error map. Intelligent vehicles (CAV2 and CAV3 on the left of Fig. 6.23) operating within the same region can then access this error information broadcast by RSUs. Vehicles equipped with a single sensor can optimize their navigation using the error map, while autonomous vehicles with a full sensor suite can improve sensor fusion by incorporating the prior weightings provided by the error maps. These sensor error maps, generated periodically by sensor-rich CAVs, serve as an additional layer upon the base map and point cloud map (right of Fig. 6.23) for intelligent transportation applications.

To evaluate our approach, we collect the multi-sensor data from multi-vehicles, including LiDAR, camera, IMU, and GNSS in urban day and night conditions both in high-fidelity simulation platform [73; 154] and the Hong Kong C-V2X testbed. The contributions of this work are summarized as follows,

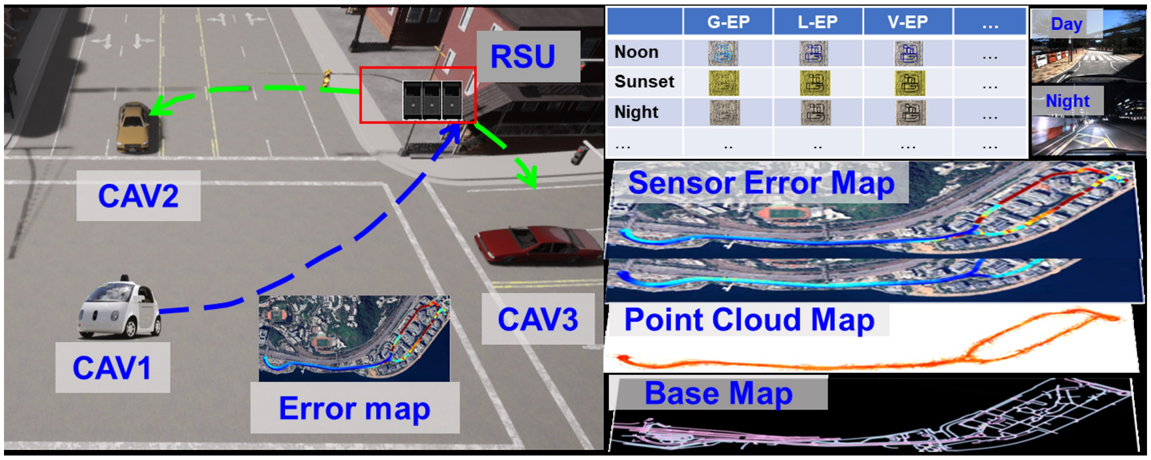


Figure 6.23: *Left*: Illustration of the error map broadcast through the RSU. CAV1 is equipped with a sensor-rich (e.g., LiDAR, camera, GNSS, and high-end devices that can provide ground truth positioning) to periodically evaluate sensor errors. CAV2 and CAV3, the connected autonomous vehicles, receive this error map information to aid their navigation using their available sensors. *Right*: Illustration of the error map serves as an additional layer upon the base map and point cloud map for intelligent transportation applications. These error maps are generated in real time from CAV1.

1. We study the continuous error map estimation based on different onboard sensors with different confidences in different urban environments. The up-to-date error maps can support various downstream navigation tasks for vehicle-infrastructure cooperation.

2. We introduce an adaptive multi-sensor integrated system that benefits from the prior information provided by the error map estimation. The weighting coefficients are computed based on the corresponding error from the error maps.

3. We evaluate the performance of the proposed system using extensive simulated data and Hong Kon C-V2X testbed data. The implementation is open-sourced at Github `https://github.com/DarrenWong/continuous_error_map` to benefit the research community.

6.4.1 Method

Overview of the Proposed System

An overview of the proposed system is illustrated in Fig. 6.24. The pipeline consists of two major modules: (1) error map generation using a sensor-rich map update vehicle, and (2) multi-sensor fusion enhanced by adaptive weighting based on the latest error map. The system inputs include LiDAR point clouds, IMU measurements, camera images, and GNSS data from CAVs. First, sensor data from the map update vehicle is processed to produce the error map. This error map is then sent to the RSU via the V2X network. The error maps are subsequently aggregated and distributed to other CAVs in the same area through the RSU. By incorporating this error data, the system adjusts the weighting of different odometry factors to improve the multi-sensor integration and achieve more robust positioning. The coordinate systems used in this work are depicted in Fig. 6.25. The extrinsic parameters among the LiDAR, camera, and IMU are pre-calibrated, while the extrinsic parameters between the ENU and IMU body frames are estimated using the Attitude and Heading Reference System (AHRS). For details on the conversion between the ECEF and ENU frames, please refer to [15].

Error Map Estimation

This section provides a concise overview of the algorithms used in error map estimation for various sensors.

LOAM [18] is a popular LO method. An illustration of the feature extraction process is presented in Fig. 6.26 (a). The scan-to-scan and scan-to-map operations are then performed to minimize the residuals between point-to-line and point-to-plane correspondences for consecutive LiDAR frames and the map point cloud, respectively.

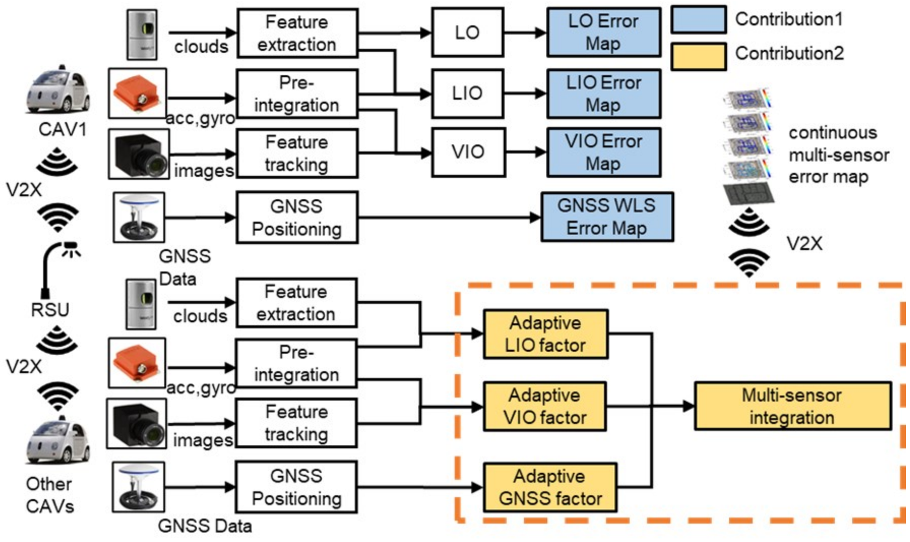


Figure 6.24: Overview of the proposed system. LO, LIO, and VIO represent the LiDAR odometry, LiDAR-inertial odometry, and visual-inertial odometry, respectively. BSM is short for basic safety message which is defined in V2X message.

The cost function for the scan-to-map process can be simplified as follows,

$$\min_{\mathbf{T}_{L_{k+1}}^{W}} \left\{ \sum_{i=1}^{N_{k+1}^{e}} \left\| \mathbf{r}\left(\mathbf{X}_{k+1,i}^{e},\ \mathbf{T}_{L_{k+1}}^{W}\right) \right\|^{2} + \sum_{i=1}^{N_{k+1}^{p}} \left\| \mathbf{r}\left(\mathbf{X}_{k+1,i}^{p},\ \mathbf{T}_{L_{k+1}}^{W}\right) \right\|^{2} \right\} \tag{6.19}$$

where $N_{k+1}^{e}$and N_{k+1}^{p} indicate the numbers of edge and plane points. $\mathbf{r}\left(\mathbf{X}_{k+1,i}^{e},\ \mathbf{T}_{L_{k+1}}^{W}\right)$ and $\mathbf{r}\left(\mathbf{X}_{k+1,i}^{p},\ \mathbf{T}_{L_{k+1}}^{W}\right)$ represent the residuals for the edge $\mathbf{X}_{k+1,i}^{e}$ and planar feature $\mathbf{X}_{k+1,i}^{p}$, respectively.

The LiDAR Inertial Odometry via Smoothing and Mapping (LIO-SAM) algorithm [61] utilizes high-frequency updates from IMU measurements to provide an initial estimate for LiDAR odometry, which is then refined using factor graph optimization.

VINS-Mono [62] is a non-linear optimization estimator designed for monocular visual odometry. It combines data from a monocular camera and an IMU using a sliding window factor graph. Fig. 6.26 (b) shows the feature tracking process used by

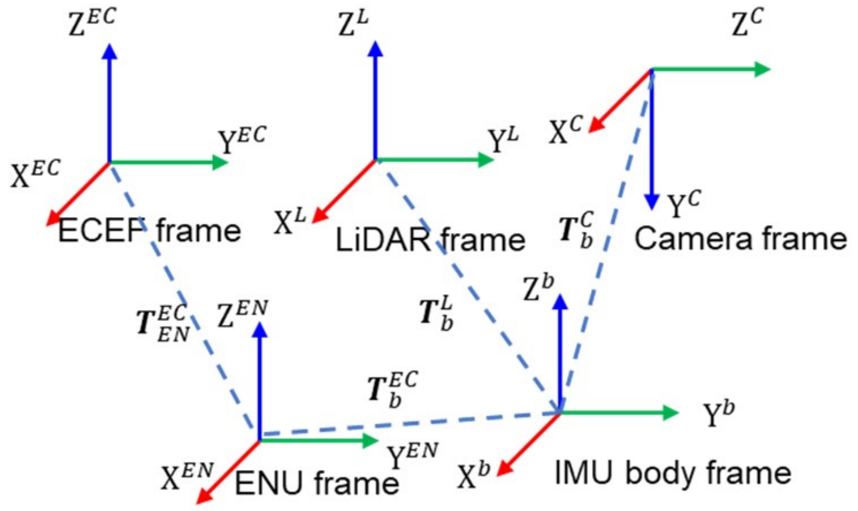


Figure 6.25: Overview of the coordinate system.

VINS-Mono. The cost function aims to minimize residuals from IMU pre-integration, visual measurements, and marginalization.

$$\min_{\chi}\left\{\left\|\mathbf{r}_p-\mathbf{H}_p\chi\right\|^2+\sum_{k\epsilon\mathcal{B}}\left\|\mathbf{r}_{\mathcal{B}}\left(\widehat{\mathbf{Z}}_{b_{k+1}}^{b_k},\chi\right)\right\|_{\mathbf{P}_{b_{k+1}}^{b_k}}^2+\sum_{(l,j)\epsilon C}\left\|\mathbf{r}_C\left(\widehat{\mathbf{Z}}_{\mathbf{l}}^{\mathbf{c_j}},\chi\right)\right\|_{\mathbf{P}_l^{\mathbf{c}_j}}^2\right\} \tag{6.20}$$

where χ is the state vector in the sliding windows. $\mathbf{r}_p$, $\mathbf{r}_{\mathcal{B}}$ and $\mathbf{r}_C$ denotes the residuals for marginalization, IMU pre-integration and visual reprojection, respectively. $\mathbf{H}_p$ represents the marginalization estimation matrix, while $\mathbf{P}\,(.)$ presents the covariance matrix of each term. $\widehat{\mathbf{Z}}_{\mathbf{b}_{k+1}}^{\mathbf{b}_k}$and $\widehat{\mathbf{Z}}_{\mathbf{l}}^{\mathbf{c_j}}$ denotes the observations of IMU and features, respectively.

GNSS WLS [129] is a widely used method that employs WLS to estimate the position of a GNSS receiver. Both pseudorange and carrier phase measurements are utilized to compute the GNSS WLS solution. The objective of the WLS method is to minimize the weighted sum of residuals from the GNSS observations.

$$\mathbf{x}^G=\left(\mathbf{H}^T\mathbf{W}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{z} \tag{6.21}$$

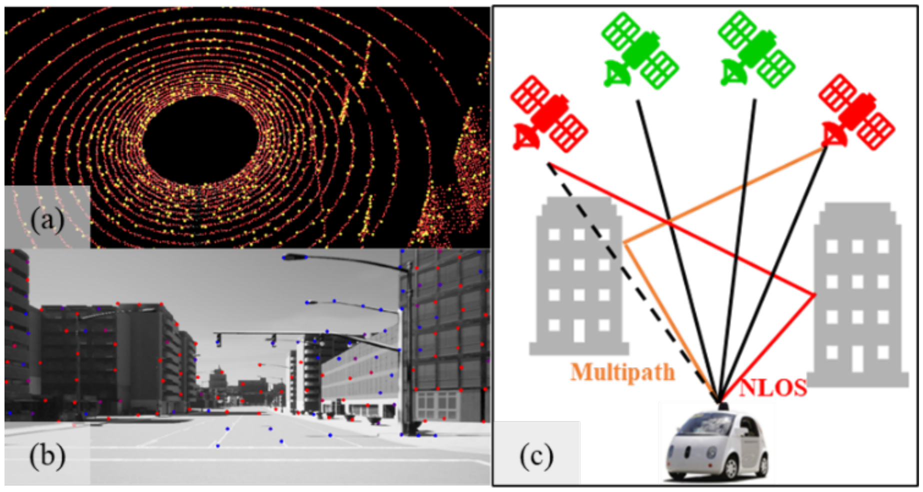


Figure 6.26: Illustration of (a) Edge and planar features extracted by LOAM, with red circles representing planar features and yellow points denoting edge features; (b) VINS-Mono's feature tracking process, where red features indicate longer tracking duration compared to blue features; (c) GNSS NLOS and multipath interference in urban environments, caused by reflections off surrounding buildings.

where $\mathbf{x}^G$ is the GNSS state to be estimated. $\mathbf{H}$ denotes the design matrix. $\mathbf{W}$ is the weighted matrix for observation, which is given based on the standard deviation of the satellite measurement. $\mathbf{z}$ is the measurement vector. GNSS-RTK positioning can further improve the WLS solutions by eliminating systematic errors by adopting the DD operation [129] between the observations received from a reference station and the one from the user. There are two main steps in GNSS-RTK positioning. In the first step, the float solution is estimated by the WLS [129]. Secondly, the integer ambiguity resolution (AR) is performed using the least-squares ambiguity decorrelation adjustment (LAMBDA) [130] algorithm in the second stage. For implementation details of GNSS-RTK positioning, refer to RTKLIB [24].

In summary, the methods mentioned above perform effectively in ideal conditions. However, both LIO and VIO experience degraded positioning accuracy in featureless environments or when confronted with unmodeled outliers such as dynamic objects in urban settings. Similarly, GNSS positioning is affected by non-line-of-sight (NLOS) signals and multipath interference, as illustrated in Fig. 6.26 (c). Since the influence of dynamic objects has been examined in our previous work [1; 2], this study focuses on sensor performance degradation due to varying static environments in urban areas across different sensor types.

Given LiDAR odometry and LIO which provides high-rate relative measurements, LO/LIO error e_L denotes the translational components of the RPE of the ground truth state $\mathbf{T}_{j,k,gt}$ and estimated state $\mathbf{T}_{j,k,est}$ between timestamp j and k.

$$e_L = \left\| trans\left(\left(\mathbf{T}_{L_j,L_k,gt}\right)^{-1}\left(\mathbf{T}_{L_j,L_k,gt}\right)\right)\right\| \tag{6.22}$$

Similarly, the VIO error e_C can be expressed as,

$$e_C = \left\| trans\left(\left(\mathbf{T}_{C_j,C_k,gt}\right)^{-1}\left(\mathbf{T}_{C_j,C_k,gt}\right)\right)\right\| \tag{6.23}$$

Regarding the GNSS positioning which provides the absolute positioning, GNSS error e_G denotes the absolute pose error (ATE) of the timestamp j,

$$e_G = \left\| trans(\mathbf{T}_{j,\ gt}) - trans(\mathbf{T}_{j,est})\right\| \tag{6.24}$$

By utilizing the map update vehicle, which collects data at regular intervals, we can periodically generate error maps for the listed algorithms based on the data gathered from the available sensors. Fig. 6.27 demonstrates the GNSS positioning error map generated in the Hong Kong C-V2X testbed at a single route, which includes the timestamp, geographic information, and corresponding error. The matrix form $\mathbf{M}_t$ of the error maps at timestamp t can be expressed as,

$$\mathbf{M}_t = \begin{bmatrix} s_L & t_1 & x_1 & y_1 & z_1 & e_{L,1} & \dots \\ s_C & t_1 & x_1 & y_1 & z_1 & e_{C,1} & \dots \\ s_G & t_1 & x_1 & y_1 & z_1 & e_{G,1} & \dots \\ s_L & s_L & x_2 & y_2 & z_2 & e_{L,1} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{6.25}$$

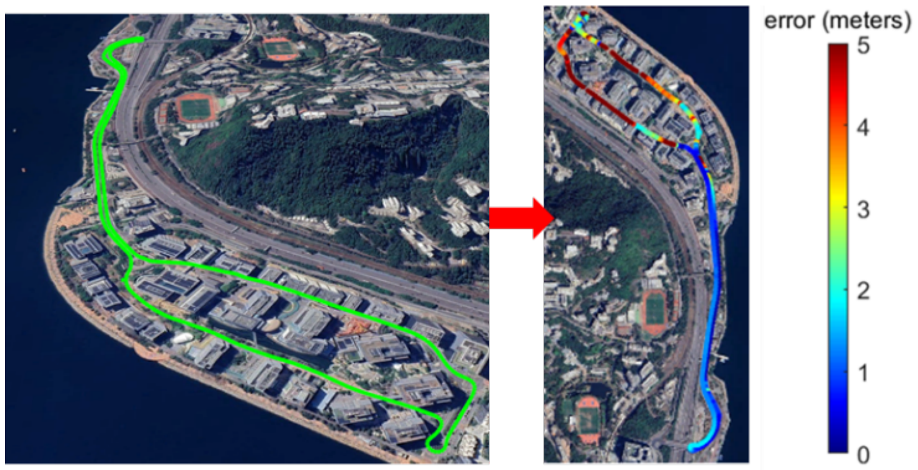


Figure 6.27: Illustration of the generated GNSS error map. The left green curve indicates the ground truth estimation by the sensor-rich map update vehicle while the right color curve indicates the estimated error of GNSS positioning.

where $e_{L,1}$, $e_{C,1}$, $e_{G,1}$ are the corresponding errors derived from different sensors using Equations 6.5, 6.6, and 6.24 at the timestamp t_1 and geographic position $\mathbf{T}_{1,gt} = [x_1,\ y_1,\ z_1\ ]$ in the world frame, respectively.

C-V2X Communication for error maps

C-V2X is designed to enhance road safety, improve traffic efficiency, and enable new applications and services in the context of CAVs and ITS. However, there is currently no existing C-V2X message specifically designed for transmitting error map information. To address this, the SAE J2735-defined basic safety message (BSM) [151] is introduced, as it contains essential data such as the timestamp, vehicle position, accuracy, component status, and other relevant travel information necessary for error maps. Additional information can also be included in the BSM message Part 1 & 2 if required. Fig. 6.23 illustrates the pipeline of C-V2X data communication incorporating error maps. In this system, sensor-rich vehicles transmit packed BSM data through the V2X OBU device to the RSI, which includes the RSU and edge computing capabilities, integrates the error maps and subsequently broadcasts them to nearby connected vehicles that have an OBU installed, utilizing the V2X communication channel.

Error Map-aided Adaptive Sensor Integration

Since the error information is received from the RSU, the nearby connected vehicle can utilize it to aid adaptive multi-sensor integration in urban areas. Fig. 6.28 depicts the process of obtaining the closest error information from the error map, considering the estimated trajectory under the world frame.

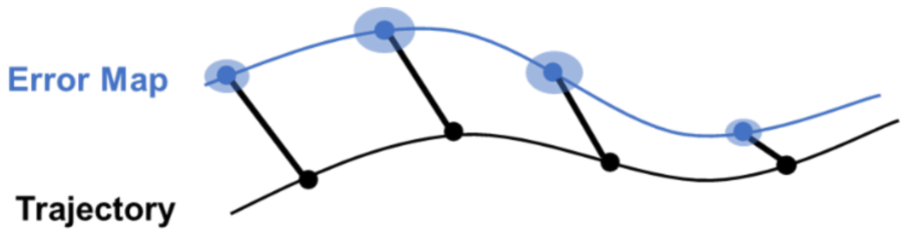


Figure 6.28: Illustration of the error information extracted from the error map under world frame. The error of the nearest point from the error map is assigned for the current pose estimation.

The weighting coefficient of the LIO factor is computed as,

$$\omega_{L_{k,k+1}} = \frac{e_{L_{k,k+1}} + e_{C_{k,k+1}}}{e_{L_{k,k+1}}} \tag{6.26}$$

which $\omega_{L_{k,k+1}}$ refers to the weighting parameters for the LIO factor. $e_{L_{k,k+1}}$ and $e_{C_{k,k+1}}$ indicate the estimated relative translation errors for LIO and VIO, respectively, at the same position.

Similarly, the weighting coefficient of the VIO factor is calculated as,

$$\omega_{C_{k,k+1}} = \frac{e_{L_{k,k+1}} + e_{C_{k,k+1}}}{e_{C_{k,k+1}}} \tag{6.27}$$

which $\omega_{C_{k,k+1}}$ denotes the weighting coefficient associated with the VIO factor, where a higher weight is applied as $e_{C_{k,k+1}}$ decreases. Meanwhile, the absolute GNSS measurement is incorporated into the graph optimization process when the prior positional error falls below an experimental threshold value m.

$$\omega_{G,k+1} = \begin{cases} 1, & e_{G_k} < m \\ 0, & e_{G_k} > m \end{cases} \tag{6.28}$$

Subsequently, we loosely integrate the relative LIO factor, relative VIO factor, and absolute GNSS factor, applying the corresponding weights derived from the error map. The factor graph of the proposed approach is illustrated in Fig. 6.29.

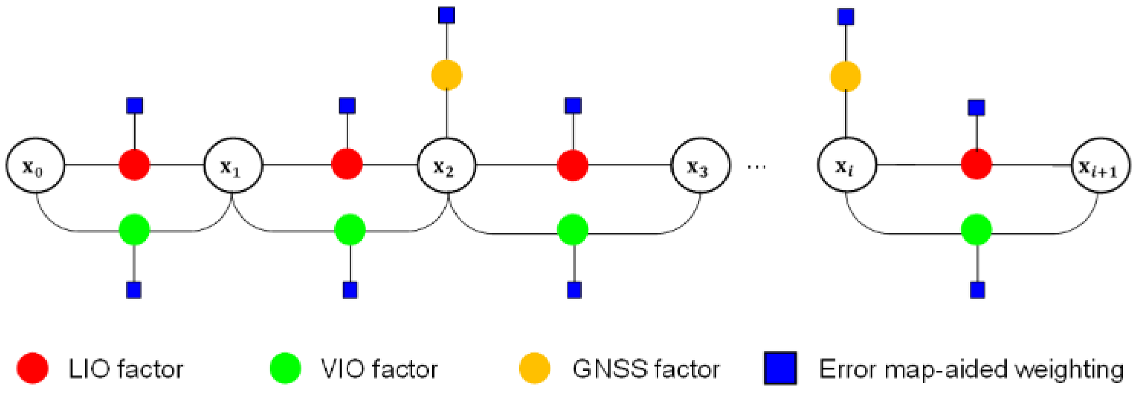


Figure 6.29: Factor graph of the proposed method.

The cost function of the proposed method can be formulated as,

$$\min_{\mathbf{T}^W_{b_{k+1}}} \left(\left\| \mathbf{r}_L \left(\omega_{L_{k,k+1}}, \mathbf{T}^W_{b_{k+1}} \right) \right\|^2_{\mathbf{P}_L} + \left\| \mathbf{r}_C \left(\omega_{C,k,k+1}, \mathbf{T}^W_{b_{k+1}} \right) \right\|^2_{\mathbf{P}_C} + \left\| \mathbf{r}_G \left(\omega_{G,k+1}, \mathbf{T}^W_{b_{k+1}} \right) \right\|^2_{\mathbf{P}_G} \right) \tag{6.29}$$

where $\omega_{L_{k,k+1}}$, $\omega_{C,k,k+1}$, $\omega_{G,k+1}$ indicate the weighting of the LIO factor, VIO factor and GNSS factor, respectively. $\mathbf{r}_C(\bullet)$ and $\mathbf{P}_{(\bullet)}$ present the residual and covariance matrix of each term, respectively. Algorithm 5 summarized the details of the error map-assisted sensor integration.

Algorithm 5: Error map-assisted sensor integrated positioning

```
Input: Point cloud $\hat{\mathcal{P}}^L$, image $\hat{\mathbf{I}}^C$, GNSS measurements $\hat{\phi}^G$, IMU data $\hat{\mathbf{Z}}^b$
Output: The optimal state $\mathbf{T}_{L_k}^W$
1: Step 1: Obtain the factor from different sensors.
2:     Step 1-1: Obtain the LIO factor using $\hat{\mathcal{P}}^L$ and $\hat{\mathbf{Z}}_k^b$ based on (6.19).
3:     Step 1-2: Obtain the VIO factor using $\hat{\mathbf{I}}^C$ and $\hat{\mathbf{Z}}^b$ based on (6.20).
4:     Step 1-3: Obtain the GNSS unary factor using $\hat{\phi}^G$ based on (6.21).
5: Step 2: Extract error information from the error map and compute sensor
   weighting coefficients based on (6.26), (6.27) and (6.28).
6: Step 3: Minimize the cost function to obtain the optimal $\mathbf{T}_{L_k}^W$ based on (6.29).
```

6.4.2 Performance Evaluation

Experiment Setup

Sensor Setups: The effectiveness of the proposed method is validated using simulation and the real data collected in the Hong Kong C-V2X testbed.

The CARLA [73] autonomous driving simulator and the RUMS [154] urban GNSS simulator are utilized to generate realistic sensor data in simulated scenarios. To examine the impact of varying time conditions, we simulated data during different time slots (noon, sunset, and night) within urban environments. Dynamic objects are not included in this research, as their effects have been extensively evaluated in our previous work [2]. We collected data at a frequency of 10 Hz for LiDAR point clouds, 20 Hz for images, 100 Hz for IMU measurements, and 100 Hz for ground truth positioning, all of which explicitly consider real-world noise in urban scenarios (example data are shown in Fig. 6.30) while maintaining a maximum speed of 30 km/h. Additionally, GNSS measurements were simulated using the realistic RUMS simulator, which accounts for the effects of signal reflection and diffraction based on a 3D building model, as CARLA can only simulate GNSS solutions with Gaussian noise. Details of the sensor setup are provided in Table 6.6 and Fig. 6.30.

In real-world evaluation, UrbanNav vehicle platform [3] is employed for onboard data collection, which includes GNSS, INS, cameras, and LiDARs. NovAtel SPAN-

CPT and Inertial Explorer software [107] are used for GT positioning. The accuracy of our ground truth system is centimeter level continuously, plus 1 ppm, with RTK enabled. Each RSI in the Hong Kong C-V2X testbed [137] includes GNSS measurements, high-resolution LiDAR, and low-latency V2X communication. We collected 1Hz U-blox F9P GNSS data, 10 Hz Velodyne HDL 32-line LiDAR data, 20Hz Point Grey camera data and 400 Hz Xsens Mti-10 IMU data from our vehicle platform under a maximum speed of 50 km/h. The RSUs are distributed equally in this urban environment with an effective range of 500 meters. The RSUs can provide the error map service using the direct C-V2X (PC5) with a communication delay of less than 20 milliseconds. The details of the sensor setup are provided in Fig. 6.31. We divided the collected sequences from the experimental vehicles into two sub-sequences to estimate the error map and evaluate the effectiveness of the proposed method. The details of the simulated and real-world sequences are listed in Table 6.7.

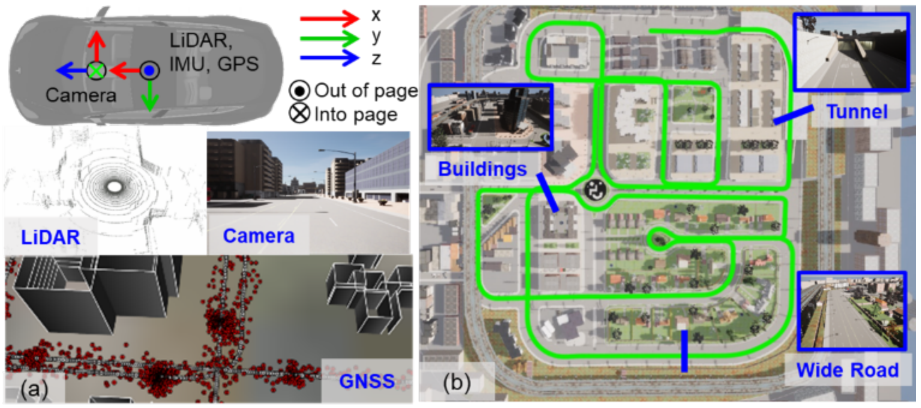


Figure 6.30: (a) Sensor for the simulated platform; (b) The evaluated trajectory and its ground truth trajectory in the simulated urban areas (modified Town03 in CARLA). It contains diverse urban scenarios, including medium-height buildings, wide roads, and tunnels.

Evaluated Methods: To evaluate the performance of algorithms with different types of sensors, we use the following methods. We first analyzed the error matrix of each sensor via the RPE to generate the error maps. The performance of the

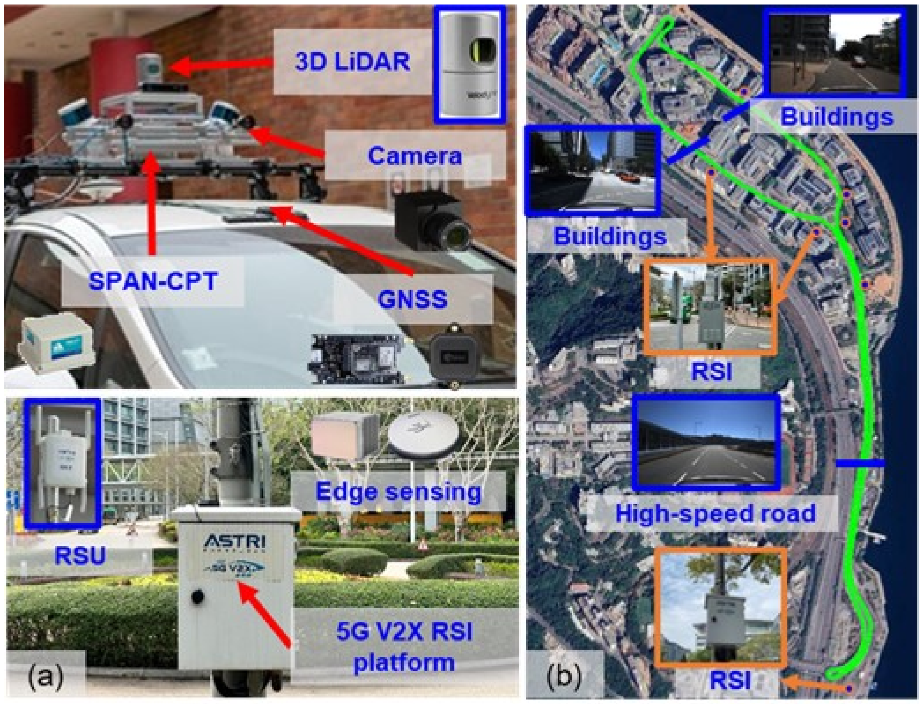


Figure 6.31: (a) Setup for the vehicle-infrastructure platform; (b) The evaluated Hong Kong C-V2X scene and its ground truth trajectory. The distributed RSIs are marked in orange.

proposed method was evaluated via the ATE to investigate the global accuracy of the trajectory.

(1) **LOAM** [18]: The popular LiDAR odometry.

(2) **LIO-SAM** [61]: The state-of-the-art LiDAR-inertial odometry. It can loosely couple with GPS if available.

(3) **VINS-Mono** [62]: The popular visual-inertial odometry.

(4) **GNSS Positioning** [24]: GNSS WLS positioning is adopted for simulated data while GNSS-RTK positioning is adopted for real-world data as the simulated data contains pseudorange measurements only.

(5) **LVI-SAM** [155]: The state-of-the-art tightly coupled LiDAR-visual-inertial

Table 6.6: Sensor description for the simulated data.

Sensor	**Description**
1x LiDAR	360° Horizontal FOV, +10°~-30° vertical FOV, 100 meters, Velodyne[1] HDL 32 noise model, 5cm standard deviation on range measurement
1x RGB Camera	FoV of 90°, 960x600, forward, default noise setting[2] in CARLA [73]
1x IMU	100 Hz, 9-axis, Xsens MTi[3] noise model according to UrbanNav [3]
1x GNSS	10 Hz, Ublox M8T[4] noise model with sign reflection from buildings

1. `https://velodynelidar.com/`
2. `https://carla.readthedocs.io/en/latest/ref_sensors/#rgb-camera`
3. `https://www.movella.com/products/sensor-modules`
4. `https://www.u-blox.com/en/product/neolea-m8t-series`

odometry. The depth information from LiDAR will help recover the scale of visual odometry.

(6) **EMA-MS:** The proposed error map-assisted multi-sensor integrated positioning. The weighting of the different sensors is computed using the continuous error estimated by the map update vehicle periodically.

Experimental Validation in Simulated Urban Areas

The evaluated data was collected in typical urban areas (modified Town03 in CARLA) with a total length of 3.5 km, as shown in Fig. 6.30-(a). The environment includes medium-height buildings, wide roads, and tunnels. As detailed in Table 6.7, we collected three sets of data using a single sensor-rich CAV under different time conditions (*urban-noon1, urban-sunset1,* and *urban-night1*) for error map estimation. Subsequently, we gathered another three sets of data with the tested CAV (*urban-noon2, urban-sunset2,* and *urban-night2*) during the same time slots to verify the effectiveness of the proposed method, which is aided by the error maps generated from the previous data.

Table 6.7: The simulated urban and real-world datasets for evaluation.

Data		Description
Simulated Urban	*urban-noon 1*	data collected at noon is used to generate error maps
	urban-noon 2	data for error map-aided sensor fusion at noon
	urban-sunset 1	data collected at sunset is used to generate error maps
	urban-sunset 2	data for error map-aided sensor fusion at sunset
	urban-night 1	data collected at night is used to generate error maps
	urban-night 2	data for error map-aided sensor fusion at night
Hong Kong C-VX2 Testbed	*testbed-day 1*	data collected at daytime is used to generate error maps
	testbed-day 2	data for error map-aided sensor fusion at daytime
	testbed-night 1	data collected at night is used to generate error maps
	testbed-night 2	data for error map-aided sensor fusion at night

1) Error Map Estimation

Translation error is used as the criterion for error map estimation. Table 6.8 illustrates the positioning performance of the evaluated methods. The performance of LiDAR-only LOAM is inferior to that of other methods, with an error of up to 0.520 meters in *urban-noon1*. With the assistance of inertial sensors, the RPE is reduced to 0.318 meters in *urban-noon1*. However, the RPE and absolute translation error (ATE) are still inadequate for autonomous driving due to degradation in the tunnel (as shown in Fig. 6.32) and accumulated drift. VINS demonstrates better performance regarding RPE and ATE in the absence of dynamic objects. Nevertheless, the RPE increases from 0.066 to 0.077 meters as the illumination worsens from *urban-noon1* to *urban-night1* for the visual-based method. GNSS WLS achieves the best results in terms of ATE since GNSS can provide absolute positioning without drift accumulation. However, noisy GNSS positioning (shown in Fig. 6.32) in urban areas and unavailable measurements in tunnels fail to meet the requirements for CAVs. In summary, each sensor has its limitations, and positioning results could be enhanced through appropriate sensor integration.

Table 6.8: Error evaluation of listed methods using different sensors.

Data	**Methods**	**RPE RMSE (m)**	**ATE RMSE (m)**
urban-noon 1	LOAM	0.520	54.672
	LIO-SAM	0.318	16.935
	VINS	0.045	12.076
	GNSS Positioning	*	1.357
urban-sunset 1	LOAM	0.643	43.619
	LIO-SAM	0.381	26.324
	VINS	0.036	11.312
	GNSS Positioning	*	1.171
urban-night 1	LOAM	0.536	43.982
	LIO-SAM	0.297	17.809
	VINS	0.077	12.914
	GNSS Positioning	*	1.2205

Fig. 6.32 and Fig. 6.33 illustrate the positioning errors and trajectories for the four evaluated methods, respectively. The black curve represents the ground truth. The LiDAR-based methods, LOAM and LIO-SAM, experience degradation in the tunnel (as shown in Fig. 6.32 (A)), making it challenging to achieve satisfactory positioning. The visual-based method, VINS, exhibits smoother performance across different time slots but suffers from decreased accuracy under poor illumination, as demonstrated in Fig. 6.32 (C). The GNSS WLS method provides average absolute performance when compared to both LiDAR-based and visual-based methods; however, it experiences significant errors near tall buildings, as illustrated in Fig. 6.32 (B).

Fig. 6.34 illustrates the results of the error maps from the listed methods using *urban-noon 1*. The error maps for *urban-sunset 1* and *urban-night 1* can be found in the supplementary materials. Similar to Figs. 6.32 and 6.33, we observe signifi-

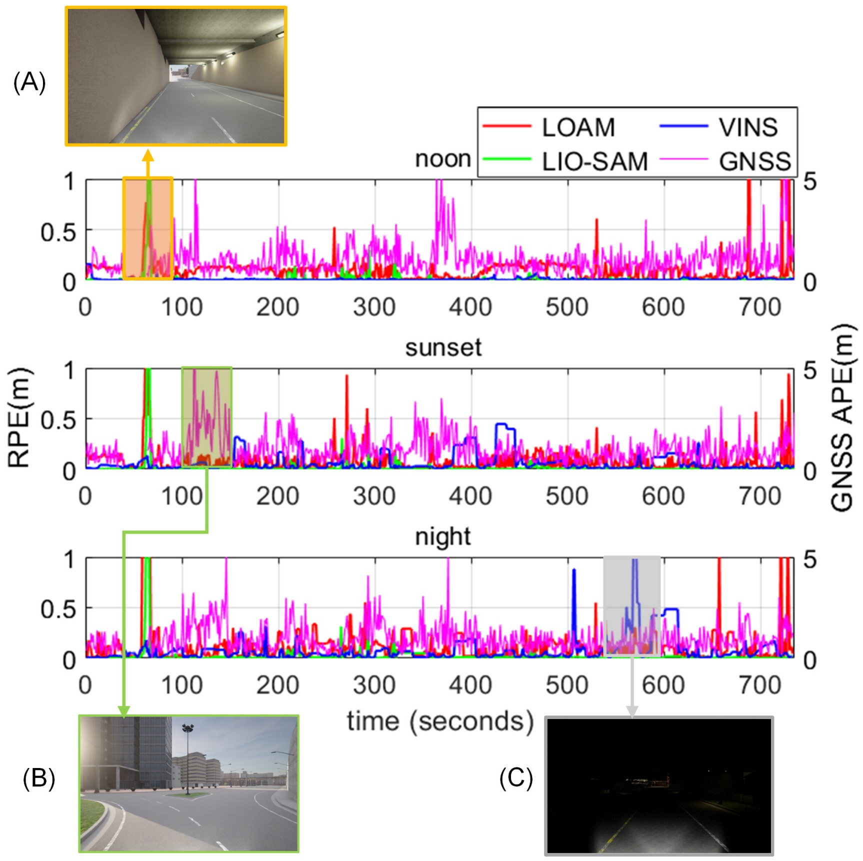


Figure 6.32: Positioning errors of the listed methods (GNSS APE is a 2D error). The x-axis represents the timestamp, while the y-axis indicates the corresponding error. Areas (A), (B), and (C) highlight the challenging scenarios for the LiDAR-based methods, GNSS positioning, and VINS, respectively.

cant errors from LOAM and LIO-SAM in tunnel scenarios. Interestingly, fusing the state estimation from VINS and the LiDAR-based method appears complementary in typical tunnel scenes, as indicated by the error maps of LIO-SAM and VINS. Additionally, the GNSS error map can be utilized to identify areas with minimal errors, which can facilitate sensor integration. Consequently, each CAV equipped with LiDAR, visual, or GNSS sensors can benefit from the real-time error map estimations broadcasted by the RSU. One potential application is that other autonomous vehicles (AVs) could leverage the state estimation errors to optimize planning and control in regions with larger errors. Furthermore, better performance can be achieved by using

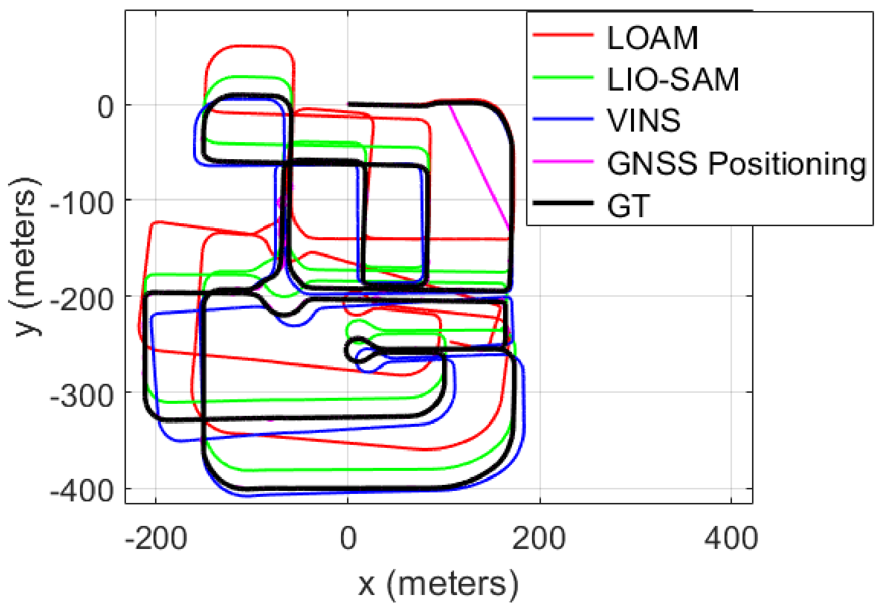


Figure 6.33: Illustration of trajectories of the four methods under *urban-noon 1*. The trajectories of *urban-sunset 1* and *urban-night 1* can be found in the supplementary.

the error map for adaptive weighting of different sensors if the CAV is equipped with a full suite of sensors, as discussed in the following section.

1) Positioning Results of the Proposed Method

The positioning performance for the test data is presented in Table 6.9. The proposed EMA-MS method outperforms state-of-the-art techniques, achieving an ATE RMSE of 1.019 meters in *urban-noon 2*. In the same scene, the existing LIO-SAM recorded an ATE RMSE of 19.749 meters, with a mean error of 17.723 meters. VINS demonstrated a similar performance, yielding an ATE RMSE of 16.103 meters. LVI-SAM performed accurately on the test data due to its tightly coupled LIO and VIO systems. However, the error of LVI-SAM increases as illumination conditions worsen, particularly in *urban-night 2*, where the ATE RMSE escalated from 1.414 meters to 3.658 meters. EMA-MS maintained superior performance by loosely coupling LIO, VIO factors, and GNSS adaptive weighting in *urban-noon 2*.

Fig. 6.35 illustrates the trajectories produced by the five methods. It is evident

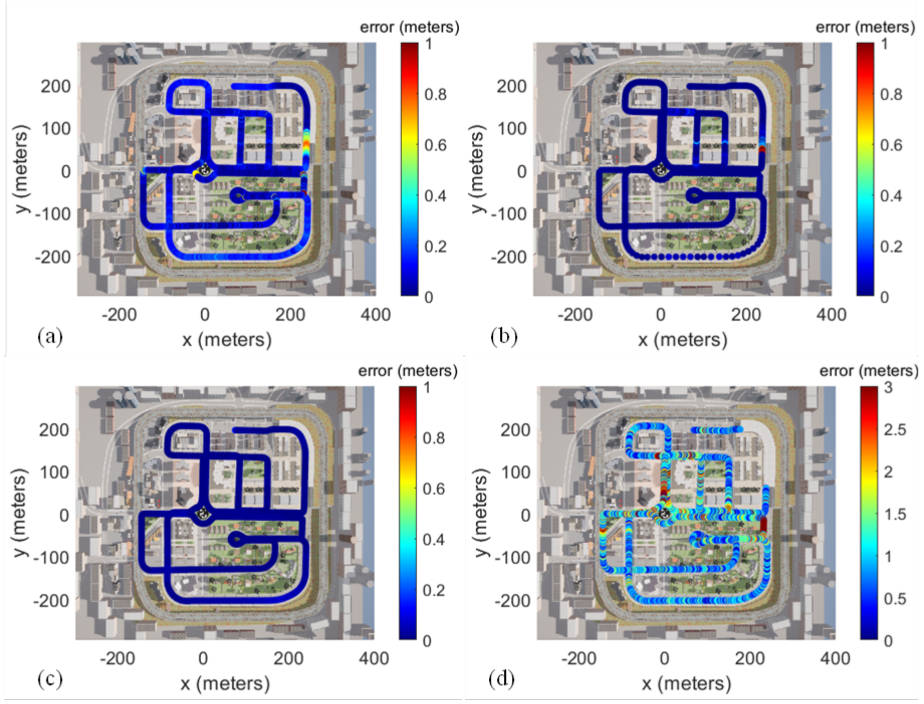


Figure 6.34: Illustration of error map of the listed methods in data *urban-noon 1*. (a) LOAM; (b) LIO-SAM; (c) VINS;(d) GNSS WLS. The error maps of sunset and night can be found in supplymentary[6]. Note that the error color which larger than the color legend upper limit will be the same red color as the upper limit.

that the estimated trajectory of the proposed EMA-MS method closely aligns with the ground truth, particularly in the zoomed-in U-turn area. The trajectories for *urban-sunset 2* and *urban-night 2* can be found in the supplementary materials.

Experimental Validation in Hong Kong C-V2X Testbed

To further evaluate the proposed method in real complex scenes, we conduct another experimental validation in the Hong Kong C-V2X testbed with a total length of 4.19 km, as shown in Fig. 6.31-(b). It involves buildings and high-speed roads. As shown in real-world data descriptions in Table 6.7, we collected two sets of data using one sensor-rich CAV under different time conditions (*testbed-day 1 and testbed-night 1*) for error map estimation. Afterward, we collected another two sets of data (*testbed-day 2 and testbed-night 2*) with the same time slots to verify the effectiveness of the

Table 6.9: Performance evaluation of listed methods. RMSE Improvement is calculated based on the evaluated method compared to the LIO-SAM.

Data	Methods	ATE RMSE (m)	ATE MEAN (m)
urban-noon 2	LIO-SAM	19.749	17.723
	VINS	16.103	15.202
	LVI-SAM	1.414	2.095
	LIO-SAM w/ GNSS	11.077	7.246
	EMA-MS	**1.019**	**0.828**
urban-sunset 2	LIO-SAM	24.172	21.678
	VINS	29.565	28.06
	LVI-SAM	2.549	2.456
	LIO-SAM w/ GNSS	12.040	8.238
	EMA-MS	**1.732**	**1.568**
urban-night 2	LIO-SAM	21.376	19.168
	VINS	12.425	11.382
	LVI-SAM	3.658	3.523
	LIO-SAM w/ GNSS	6.715	4.787
	EMA-MS	**2.909**	**1.852**

proposed method aided by the error map generated by the previous data.

1) Error Map Estimation

Table 6.10 demonstrates the positioning performance of the evaluated methods. The performance of LiDAR-only LOAM is worse than other methods, up to 0.452 meters in terms of RPE in *HK C-V2X testbed day 1.* With the help of the inertial sensors, the RPE of the LIO-SAM is decreased to 0.233 meters. VINS achieves an average performance of 0.438 meters in terms of RPE. The RPE of GNSS positioning is up to 9 meters due to multi-path in urban areas.

Fig. 6.36 and Fig. 6.37 show the positioning error and the trajectories using four methods, respectively. The black curve represents the ground truth. The LiDAR-based method LOAM and LIO-SAM are degenerated in the featureless scene (shown in Fig. 6.36 (A) and (E)). VINS suffers a worse performance if under ill illumination and featureless scenes, as shown in Fig. 6.36 (D) and (E). The GNSS positioning method provided average absolute performance compared to the LIO-SAM. Similar

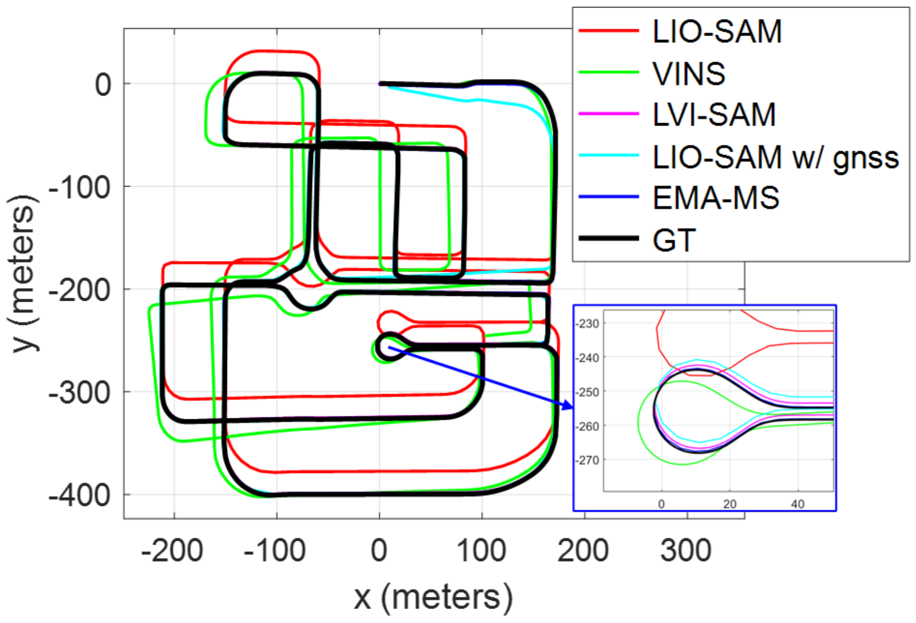


Figure 6.35: Trajectories of the listed methods in Data *urban-noon 2*. The area marked in blue is the zoom-in of the U-turn area. The trajectories of *urban-sunset 2* and *urban-night 2* can be found in supplementary.

to the simulated scene, it suffers from large errors entering the dense urban areas, as shown in Fig. 6.36 (C) and 6.36 (F).

Fig. 6.38 demonstrates the results of the error maps from the listed methods using *testbed-day 1*. Similar to Figs. 6.36 and 6.37, it can observe large errors of VINS in high-speed scenarios. The performance of GNSS positioning is degraded near the high-building areas. Interestingly, it is complementary if we fuse the state estimation of multiple sensors according to the error maps. We can obtain better performance by utilizing the error map for the adaptive weighting of different sensors if the CAV is equipped with all sensor-suit, as shown in the following section.

1) Positioning Results of the Proposed Method

The positioning performance for the test data is shown in Table 6.11. An ATE RMSE of 12.476 meters was obtained using the existing LIO-SAM in the same scene, with a mean error of 10.875 meters. The error of VINS increases as the illumination condition worsens, especially on *testbed-night 2*. The LVI-SAM obtained a similar

Table 6.10: Error evaluation of listed methods using different sensors in terms of RMSE in Hong Kong C-V2X testbed.

Data	**Methods**	**RPE RMSE (m)**	**ATE RMSE (m)**
testbed-day 1	LOAM	0.452	223.142
	LIO-SAM	0.233	9.546
	VINS	0.438	556.614
	GNSS Positioning	*	9.597
testbed-night 1	LOAM	0.159	47.848
	LIO-SAM	0.357	16.501
	VINS	0.134	258.772
	GNSS Positioning	*	9.306

performance as VINS due to the poor performance of VINS in these dynamic illumination scenes. With the aid of the weighted constraint, the error of the proposed method is reduced to 4.743 meters in *HK C-V2X day 2* and 4.120 meters in *HK C-V2X night 2*, respectively. Fig. 6.39 shows the trajectories using five methods. It can be seen that the estimated trajectory of the proposed EMA-MS method aligns well with the ground truth, especially in the zoom-in U-turn area. The trajectories of *HK C-V2X night 2* can be found in the supplementary. In summary, using the continuous error map can significantly improve the positioning performance of the multi-sensor integration in urban areas.

Table 6.11: 2D Positioning evaluation of listed methods.

Data	**Methods**	**ATE RMSE (m)**	**ATE MEAN (m)**
testbed-day 2	LIO-SAM	12.476	10.875
	VINS	424.171	378.136
	LVI-SAM	633.347	575.513
	LIO-SAM w/ GNSS	5.536	4.289
	EMA-MS	**4.743**	**3.911**
testbed-night 2	LIO-SAM	8.871	8.275
	VINS	600.720	422.203
	LVI-SAM	410.649	310.221
	LIO-SAM w/ GNSS	6.091	4.471
	EMA-MS	**4.120**	**3.498**

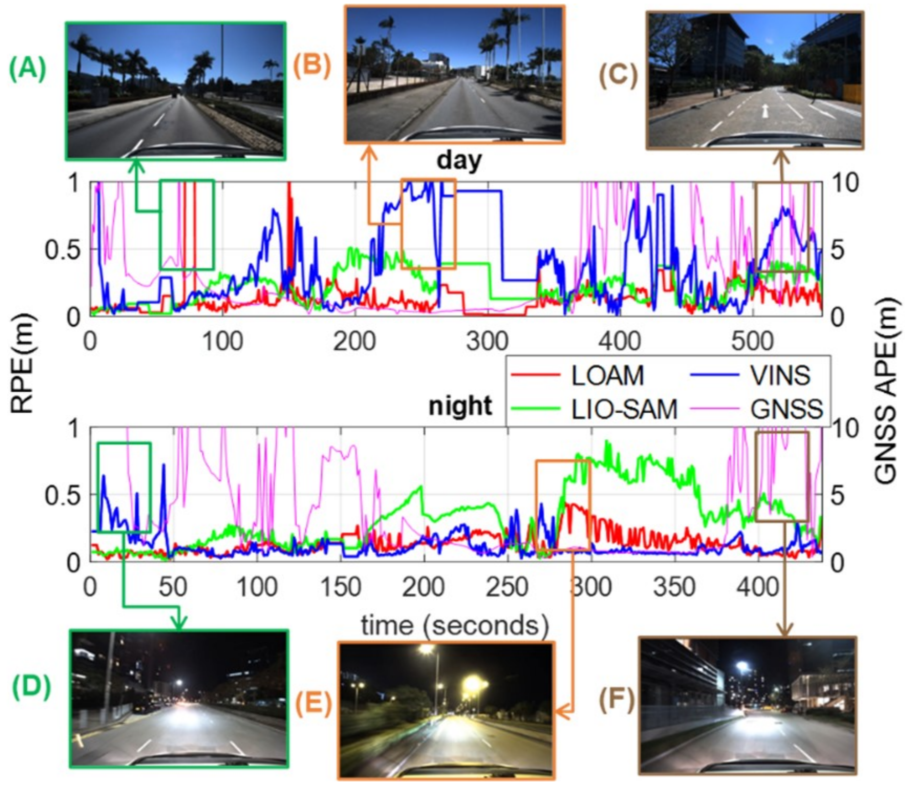


Figure 6.36: 2D Positioning errors of the listed methods. The x-axis denotes the timestamp while the y-axis indicates the error, respectively. Scene (A)-(F) are the marked area of the challenging scenarios for the state estimation.

6.5 Summary

Roadside Infrastructure assisted LiDAR/Inertial-based Mapping for Intelligent Vehicles in Urban Areas: With the growth and development in smart cities, there is a huge demand for vehicle-infrastructure cooperation. This study presented a complete pipeline to enhance the performance of the LIO aided by the global constraint provided by the RSIs. Evaluation results on the Hong Kong C-V2X testbed show that our proposed method outperforms the state-of-the-art method regarding absolute positioning accuracy.

Roadside Infrastructure-Assisted LiDAR/Inertial-Based Mapping for Intelligent Vehicles in Urban Areas: As smart cities continue to evolve, the

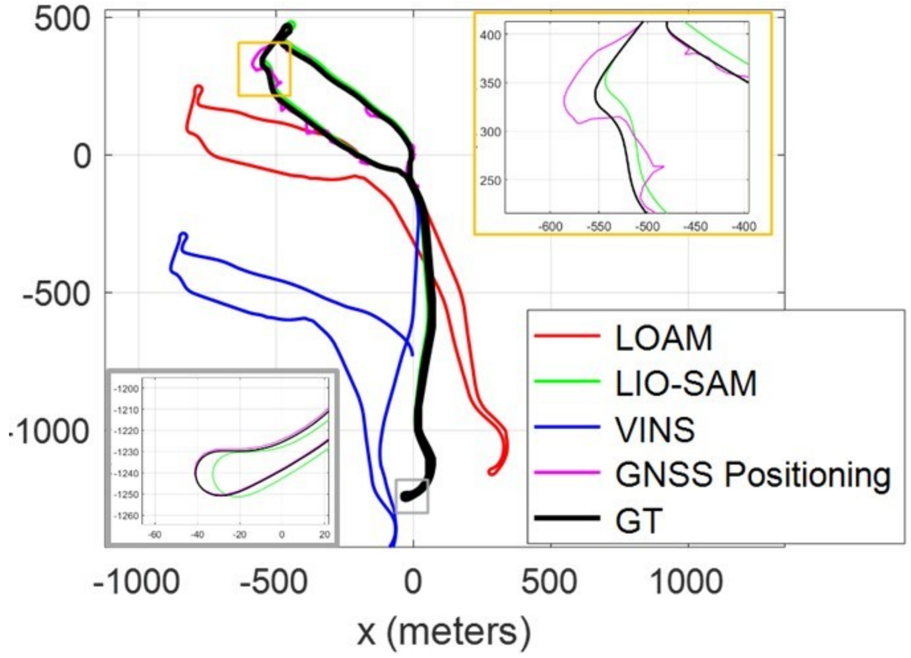


Figure 6.37: Illustration of trajectories of the four methods under *testbed-night 1*. The trajectories of *testbed-night 1* can be found in the supplementary

demand for efficient vehicle-infrastructure cooperation has grown significantly. In this study, we propose a method aimed at improving LIO performance by leveraging global constraints from roadside infrastructures (RSIs). Evaluation results from the Hong Kong C-V2X testbed show that our approach outperforms state-of-the-art methods in terms of absolute positioning accuracy.

Roadside GNSS Aided Multi-Sensor Integrated Positioning for Intelligent Vehicles in Urban Areas: This study presents the RSG-GLIO method, which enhances sensor integration performance by incorporating multiple roadside GNSS DD observations. The results show that roadside GNSS can significantly improve positioning accuracy for autonomous systems and offer potential applications, such as low-cost base stations.

Continuous Error Map Aided Adaptive Multi-Sensor Integrated Positioning for Connected Autonomous Vehicles in Urban Scenarios This study introduces sensor error maps, which can be deployed as an additional layer on top of high-definition maps for V2X applications in urban environments. We demonstrate that the multi-sensor integrated system benefits from prior error information provided by error map estimation, as validated through both a realistic simulator and a real-world C-V2X testbed.

Since the same sensor noise and configuration are used in this study, future work will focus on conducting a detailed analysis of sensor correlations to further enhance the robustness of error maps. This will include:

1. Assessing Cross-Vehicle Sensor Correlations – Investigating how spatial and temporal alignment of sensors across different vehicles affects error propagation.

2. Evaluating the Impact of Sensor Accuracy and Configuration – Analyzing how variations in LiDAR, GNSS, and IMU specifications influence the accuracy of shared error maps.

3. Examining Operational Duration Effects – Exploring the impact of long-term operation and accumulated drift on error consistency across multiple vehicles.

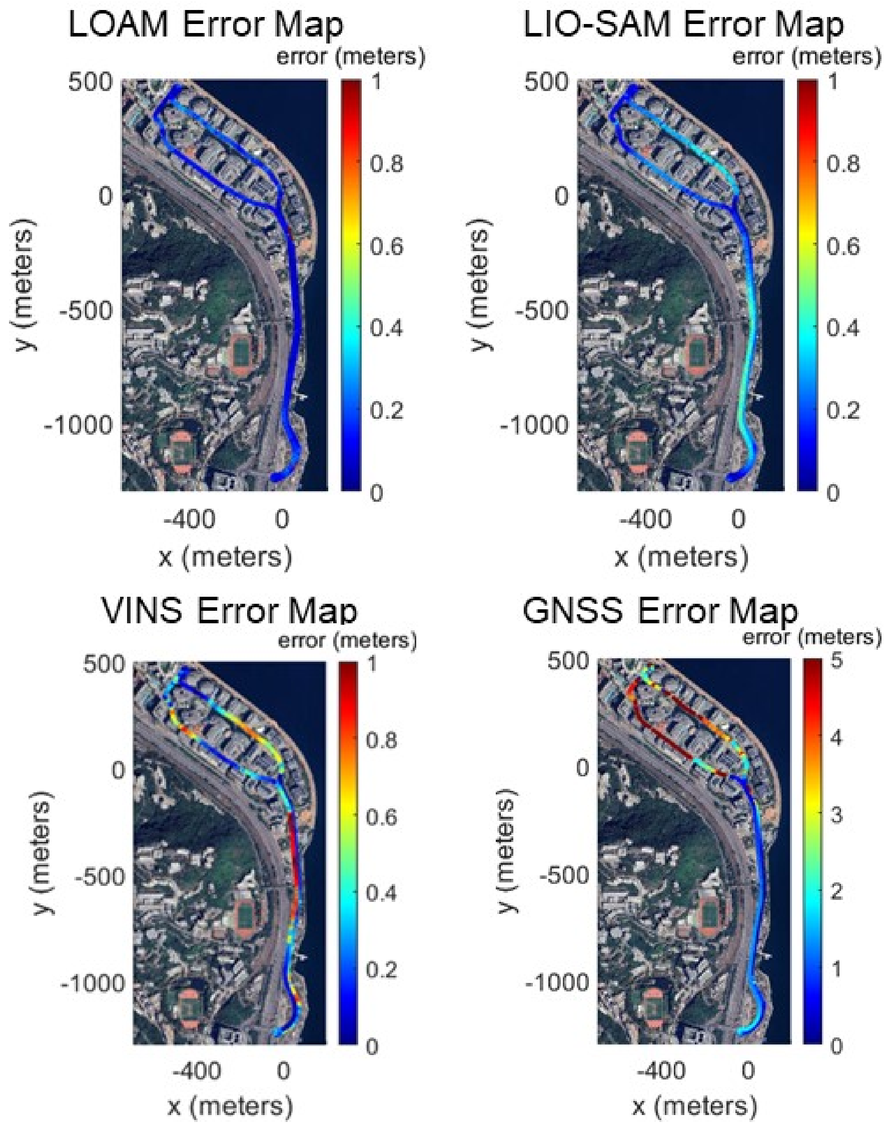


Figure 6.38: Illustration of error map of the listed methods in data *HK C-V2X day 1*. (a) LOAM; (b) LIO-SAM; (c) VINS;(d) GNSS WLS. The error maps of *HK C-V2X night 1* can be found in supplementary. Note that the error color which larger than the color legend upper limit will be the same red color as the upper limit.

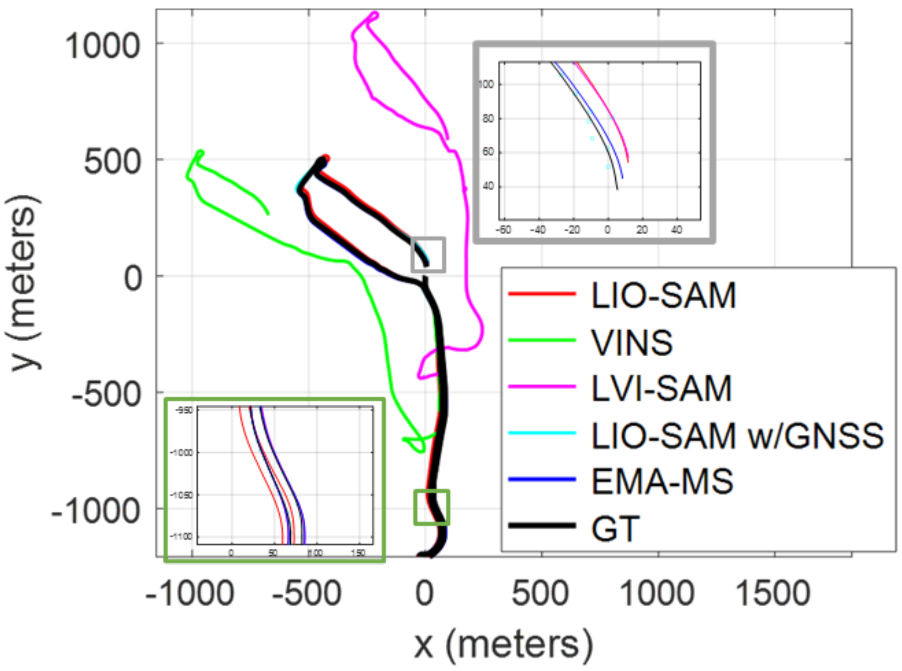


Figure 6.39: Trajectories of the listed methods in Data *HK C-V2X day 2*.

Chapter 7

Conclusions and Future Work

This thesis investigates the dynamic object-aware LiDAR odometry in dense urban areas by developing several new methods. The proposed approaches cover a range of solutions, from single-agent intelligence to vehicle-infrastructure cooperative navigation. This chapter presents the conclusion of the research, based on the four stages of study: (1) identify the limitations of existing LO pipelines in urban areas; (2) develop dynamic object-aware LO to mitigate the impact of moving objects; (3) integrate LiDAR with GNSS-RTK positioning to enhance fixing rates and state estimation; (4) develop roadside infrastructure-assisted navigation systems to address the limitations faced by individual autonomous vehicles in urban areas. Finally, several potential future research directions are discussed at the end of the chapter.

7.1 Conclusion of this Research

In Chapter 3, we conducted a benchmark performance evaluation and analysis of publicly available LO pipelines using two challenging datasets collected from the urban canyons in Hong Kong. The findings indicate that point-wise LO methods can be enhanced by incorporating additional computational steps, such as G-ICP. While voxelization in VG-ICP reduces the computational effort needed for neighboring point searches, it may also compromise accuracy in state estimation. Feature-based methods, on the other hand, demonstrate both accuracy and computational efficiency by utilizing extracted feature points. However, the performance of both types of methods is influenced by the three dominant factors we identified. Our experiments suggest that combining feature-wise and point-wise approaches can be an effective solution. The feature-based method can quickly generate a coarse odometry estimate, which can then serve as an initial guess for point-wise point cloud registration. Since point-wise methods are highly dependent on the accuracy of the initial guess, this strategy creates a coarse-to-fine LO pipeline. However, it is crucial to recognize that feature-wise methods can occasionally produce erroneous estimates. In such cases, fault detection mechanisms, such as the degeneration factor should be employed.

In chapter 4, we aimed to mitigate the impact of dynamic objects on LO. Firstly, we introduced a LiDAR-based pipeline that removes dynamic objects, leading to improved state estimation and reduced mapping residuals in densely populated urban areas. This approach also enables the creation of a clearer point cloud map that more accurately reflects the real world. Secondly, we provided a comprehensive evaluation of how dynamic objects can degrade LO performance. By analyzing various scenarios, this study highlights the challenges posed by dynamic elements in urban environments. In contrast to the work with dynamic object removal, this study proposes a reweighting strategy for dynamic objects, which enables the system to adaptively adjust the influence of these objects during state estimation. Experiments conducted on our *UrbanNav* dataset and the open-source *nuScenes* dataset validate the effectiveness of our proposed method, which outperforms both conventional methods and those relying solely on dynamic object removal [16]. By addressing the challenges introduced by dynamic objects, our approach enhances the robustness and reliability of LO, contributing to more accurate navigation in complex urban environments.

In chapter 5, we explored the complementary relationship between GNSS-RTK and LiDAR odometry. We proposed a LiDAR-aided method for detecting cycle slips in triple-differenced carrier-phase measurements. We resolve the integer ambiguity resolution after identifying the potential cycle slips. The performance of this approach is evaluated using a challenging dataset collected in typical urban canyons of Hong Kong. Step-by-step comparisons with traditional methods demonstrate the effectiveness of the proposed pipeline in terms of fixing rate and state estimation.

In chapter 6, we introduced roadside infrastructure-assisted navigation systems in urban areas, aimed at overcoming the limitations faced by individual intelligent vehicles. First, we propose a method that enhances LIO performance by utilizing global constraints from RSI. Evaluation results from the Hong Kong C-V2X testbed indicate that our approach surpasses state-of-the-art methods in absolute positioning accuracy. Secondly, we present the RSG-GLIO method, which improves sensor integration by incorporating multiple roadside GNSS DD observations. Our findings show that roadside GNSS significantly enhances positioning accuracy for autonomous systems and opens up possibilities for applications like low-cost base stations. Thirdly, we introduce sensor error maps, which can be used as an additional layer over high-definition maps for V2X applications in urban environments. We demonstrate that the multi-sensor integrated system benefits from prior error information provided by error map estimation, as validated through both a realistic simulator and the real-world C-V2X testbed.

7.2 Future Directions

Future research directions will focus on enhancing the integration of vehicle-infrastructure cooperation to enable robust, long-term mapping and meshing across diverse urban environments.

7.2.1 Collaborative high-definition map update for autonomous systems

Intelligent vehicle-infrastructure cooperation has the potential to revolutionize the transportation industry, offering optimized traffic management and a significant leap forward in the development of next-generation intelligent transportation systems. However, one of the major challenges lies in the continuous updating of high-definition (HD) maps, which is both cost-intensive and laborious. This is largely due to the reliance on expensive, sensor-rich map update vehicles, as well as the substantial manpower required to maintain the accuracy and freshness of these maps.

As depicted in Fig. 7.1, one of our key future research directions aims to address these challenges by focusing on three fundamental aspects: First, map change detection, which involves identifying and tracking dynamic changes in the environment, such as road modifications, new buildings, or updated traffic infrastructures, to ensure that maps remain up to date without needing full re-scans of entire areas. Second, we aim to explore massive data compression and restoration techniques. As HD maps consist of large, detailed datasets, compressing these maps efficiently while ensuring minimal loss of critical information is essential for real-time updates and storage efficiency. Finally, the third focus is on developing a globally consistent optimal solution for HD map updates, which entails creating algorithms that ensure all changes are seamlessly integrated into the map, maintaining global consistency across different regions and sensor data inputs, thus improving the reliability and usability of the maps across vast urban environments.

7.2.2 Calibration of onboard and roadside sensors

Our current approach relies on the assumption that both roadside and onboard sensors are pre-calibrated. However, over time, sensor calibration parameters may drift due to environmental factors, sensor wear, or slight shifts in sensor positioning,

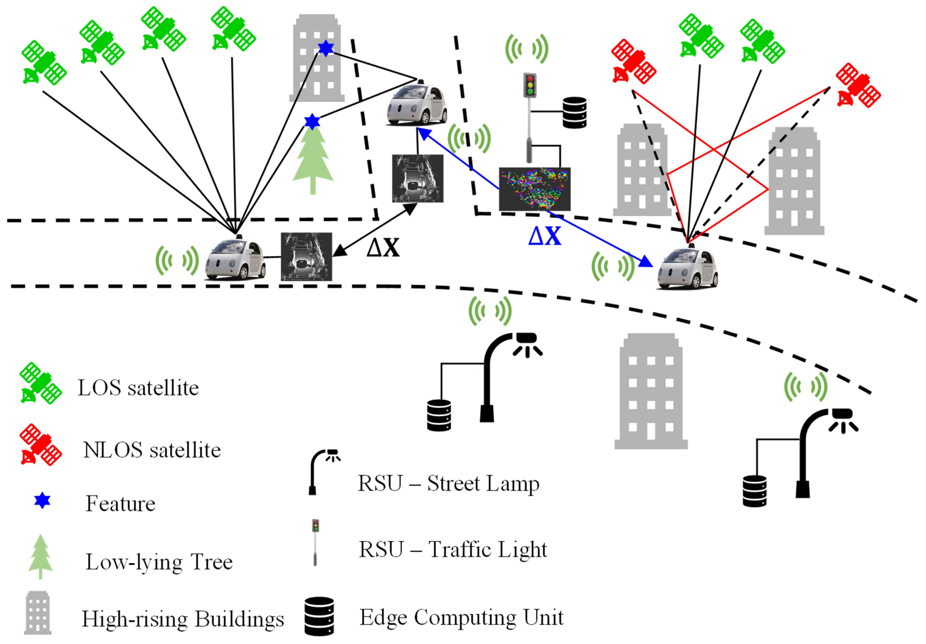


Figure 7.1: Illustration of collaborative mapping with vehicle infrastructure

necessitating periodic recalibration. Manual recalibration, especially for roadside sensors, can be both labor-intensive and costly, as it typically requires the use of static reference markers and often involves shutting down roads or setting up controlled environments. This not only incurs additional costs but also disrupts the normal functioning of transportation networks.

To address these challenges, a potential future direction involves the development of an automatic calibration system for roadside sensors, which leverages the continuous, high-precision global constraints provided by the integration of onboard sensors. As shown in Fig. 7.2, the accurate and real-time positioning data from onboard sensors can be used to continuously monitor and correct the calibration parameters of roadside sensors. This would enable a dynamic and automated recalibration process, eliminating the need for costly and time-consuming manual interventions.

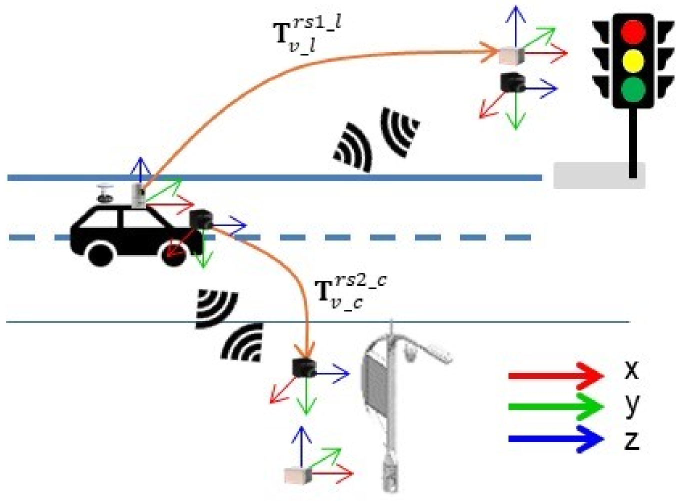


Figure 7.2: Illustration of online calibration of onboard and roadside sensors

7.2.3 Online 3D reconstruction in challenging urban areas

Various 3D reconstruction methods, such as NeRF [156] and 3D Gaussian Splatting [17], have shown remarkable performance in static scenes, as illustrated in Fig. 7.3. These techniques have pushed the boundaries of photorealistic and geometrically accurate reconstructions. However, when applied to complex urban environments characterized by highly dynamic elements—such as moving vehicles, pedestrians, and changing lighting conditions—maintaining the same level of performance becomes a significant challenge.

To address this, we will explore the integration of our navigation system with state-of-the-art 3D reconstruction techniques. By leveraging advanced multi-sensor data, including LiDAR, cameras, and GNSS, combined with robust navigation algorithms, our goal is to enable accurate and resilient 3D reconstruction even in challenging, high-dynamic urban areas. This integration could provide real-time, continuously updated 3D models of urban landscapes, which would be invaluable for applications like autonomous driving, urban planning, and virtual reality environments.

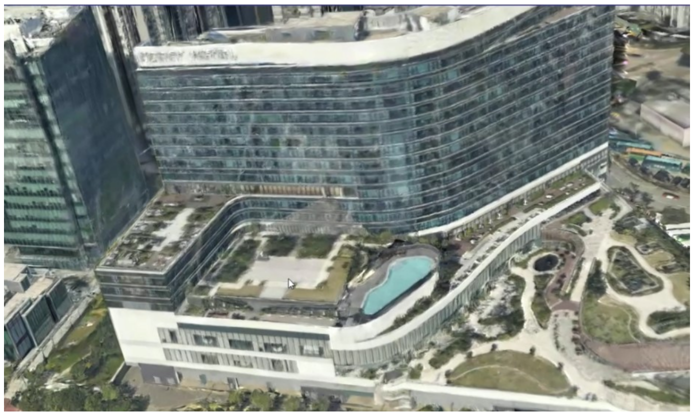


Figure 7.3: Results of 3DGS [17] using Google Map Satellite imagery in Hong Kong

Bibliography

[1] F. Huang, W. Wen, J. Zhang, and L.-T. Hsu, "Point Wise or Feature Wise? A Benchmark Comparison of Publicly Available LiDAR Odometry Algorithms in Urban Canyons," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 155–173, 2022.

[2] F. Huang, W. Wen, J. Zhang, C. Wang, and L.-T. Hsu, "Dynamic object-aware LiDAR odometry aided by joint weightings estimation in urban areas," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, pp. 3345–3359, 2024.

[3] L.-T. Hsu, F. Huang, H.-F. Ng, G. Zhang, Y. Zhong, X. Bai, and W. Wen, "Hong Kong UrbanNav: An open-source multisensory dataset for benchmarking urban navigation algorithms," *NAVIGATION: Journal of the Institute of Navigation*, vol. 70, no. 4, 2023.

[4] P. Chen, W. Guan, F. Huang, Y. Zhong, W. Wen, L.-T. Hsu, and P. Lu, "Ecmd: An event-centric multisensory driving dataset for slam," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 407–416, 2024.

[5] J. Chang, Y. Zhang, S. Fan, F. Huang, D. Xu, and L.-T. Hsu, "An anti-spoofing model based on mvm and mccm for a loosely-coupled gnss/ins/lidar kalman filter," *IEEE Transactions on Intelligent Vehicles*, 2023.

[6] J. Chang, R. Hu, F. Huang, D. Xu, and L.-T. Hsu, "Lidar-based ndt matching performance evaluation for positioning in adverse weather conditions," *IEEE Sensors Journal*, 2023.

[7] J. Chang, F. Huang, L. Zhang, D. Xu, and L.-T. Hsu, "Selection of areas for effective gnss spoofing attacks to a vehicle-mounted msf system based on scenario classification models," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 11, pp. 14 645–14 655, 2023.

[8] Y. Zhong, F. Huang, J. Zhang, W. Wen, and L.-T. Hsu, "Low-cost solid-state LiDAR/inertial-based localization with prior map for autonomous systems in urban scenarios," *IET Intelligent Transport Systems*, vol. 17, no. 3, pp. 474–486, 2023.

[9] J. Zhang, W. Wen, F. Huang, Y. Wang, X. Chen, and L.-T. Hsu, "Gnss-rtk adaptively integrated with lidar/imu odometry for continuously global positioning in urban canyons," *Applied Sciences*, vol. 12, no. 10, p. 5193, 2022.

[10] J. Chang, L. Zhang, L.-T. Hsu, B. Xu, F. Huang, and D. Xu, "Analytic models of a loosely coupled gnss/ins/lidar kalman filter considering update frequency under a spoofing attack," *IEEE Sensors Journal*, vol. 22, no. 23, pp. 23 341–23 355, 2022.

[11] J. Zhang, W. Wen, F. Huang, X. Chen, and L.-T. Hsu, "Coarse-to-fine loosely-coupled lidar-inertial odometry for urban positioning and mapping," *Remote Sensing*, vol. 13, no. 12, p. 2371, 2021.

[12] P. Yan, Z. Li, F. Huang, W. Wen, and L.-T. Hsu, "Fault detection algorithm for gaussian mixture noises: An application in lidar/imu integrated localization systems," *NAVIGATION: Journal of the Institute of Navigation*, vol. 72, no. 1, 2025.

[13] F. Huang, H. Chen, A. Urtay, D. Su, W. Wen, and L.-T. Hsu, "Roadside infrastructure assisted lidar/inertial-based mapping for intelligent vehicles in urban areas," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 5831–5837.

[14] F. Huang, W. Wen, G. Zhang, D. Su, and L.-T. Hsu, "Adaptive multi-sensor integrated navigation system aided by continuous error map from rsu for autonomous vehicles in urban areas," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 5895–5902.

[15] F. Huang, W. Wen, H.-F. Ng, and L.-T. Hsu, "Lidar aided cycle slip detection for gnss real-time kinematic positioning in urban environments," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 1572–1578.

[16] F. Huang, D. Shen, W. Wen, J. Zhang, and L.-T. Hsu, "A coarse-to-fine lidar-based slam with dynamic object removal in dense urban areas," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 3162–3172.

[17] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[18] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous robots*, vol. 41, pp. 401–416, 2017.

[19] C. Wang, L. Meng, S. She, I. M. Mitchell, T. Li, F. Tung, W. Wan, M. Q.-H. Meng, and C. W. de Silva, "Autonomous mobile robot navigation in uneven and unstructured indoor environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 109–116.

[20] P. Yang and W. Wen, "Tightly joining positioning and control for trustworthy unmanned aerial vehicles based on factor graph optimization in urban transportation," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 3589–3596.

[21] W. Ding, S. Hou, H. Gao, G. Wan, and S. Song, "Lidar inertial odometry aided robust lidar localization system in changing city scenes," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4322–4328.

[22] M. Zhao, J. Wang, T. Gao, C. Xu, and H. Kong, "Fmcw-lio: A doppler lidar-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5727–5734, 2024.

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

[24] T. Takasu and A. Yasuda, "Development of the low-cost rtk-gps receiver with an open source program package rtklib," in *International symposium on GPS/GNSS*, vol. 1. International Convention Center Jeju Korea Seogwipo-si, Republic of Korea, 2009, pp. 1–6.

[25] L.-T. Hsu, "Analysis and modeling gps nlos effect in highly urbanized area," *GPS solutions*, vol. 22, no. 1, p. 7, 2018.

[26] L.-T. Hsu, S.-S. Jan, P. D. Groves, and N. Kubo, "Multipath mitigation and nlos detection using vector tracking in urban environments," *Gps Solutions*, vol. 19, pp. 249–262, 2015.

[27] S. Gu, C. Dai, F. Mao, and W. Fang, "Integration of multi-gnss ppp-rtk/ins/vision with a cascading kalman filter for vehicle navigation in urban areas," *Remote Sensing*, vol. 14, no. 17, p. 4337, 2022.

[28] W. W. Wen, G. Zhang, and L.-T. Hsu, "Gnss nlos exclusion based on dynamic object detection using lidar point cloud," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 853–862, 2021.

[29] W. Wen, G. Zhang, and L.-t. Hsu, "Correcting nlos by 3d lidar and building height to improve gnss single point positioning," *Navigation*, vol. 66, no. 4, pp. 705–718, 2019.

[30] A. Y. Alhilal, B. Finley, T. Braud, D. Su, and P. Hui, "Street smart in 5g: Vehicular applications, communication, and computing," *IEEE Access*, vol. 10, pp. 105 631–105 656, 2022.

[31] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan *et al.*, "Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 21 361–21 370.

[32] Transport Department of the Government of Hong Kong, "Smart mobility roadmap for hong kong," https://www.td.gov.hk/filemanager/en/publication/smr_roadmap_hk.pdf, 2023, accessed on April 27, 2023.

[33] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[34] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[35] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized gicp for fast and accurate 3d point cloud registration," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 054–11 059.

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

[37] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419841532, 2019.

[38] "GitHub - HKUST-Aerial-Robotics/A-LOAM: Advanced implementation of LOAM — github.com," https://github.com/HKUST-Aerial-Robotics/A-LOAM, [Accessed 06-01-2020].

[39] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[40] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3144–3150.

[41] H. Wang, C. Wang, C.-L. Chen, and L. Xie, “F-loam: Fast lidar odometry and mapping,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4390–4396.

[42] H. Maron, N. Dym, I. Kezurer, S. Kovalsky, and Y. Lipman, “Point registration via efficient convex relaxation,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.

[43] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, “The trimmed iterative closest point algorithm,” in *2002 International Conference on Pattern Recognition*, vol. 3. IEEE, 2002, pp. 545–548.

[44] J. Serafin and G. Grisetti, “Nicp: Dense normal based point cloud registration,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 742–749.

[45] Y. Chen and G. Medioni, “Object modelling by registration of multiple range images,” *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[46] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk, “Beyond points: Evaluating recent 3d scan-matching algorithms,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3631–3637.

[47] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (fpfh) for 3d registration,” in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 3212–3217.

[48] S. Salti, F. Tombari, and L. Di Stefano, “Shot: Unique signatures of histograms for surface and texture description,” *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[49] Z. Zhang, J. Zheng, H. Xu, X. Wang, X. Fan, and R. Chen, “Automatic background construction and object detection based on roadside lidar,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4086–4097, 2019.

[50] J. Wu, H. Xu, and W. Liu, “Points registration for roadside lidar sensors,” *Transportation research record*, vol. 2673, no. 9, pp. 627–639, 2019.

[51] H. Wang, C. Wang, and L. Xie, “Intensity scan context: Coding intensity and geometry relations for loop closure detection,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2095–2101.

[52] S. Agarwal, K. Mierle *et al.*, “Ceres solver: Tutorial & reference,” *Google Inc*, vol. 2, no. 72, p. 8, 2012.

[53] C. Chen, B. Wang, C. X. Lu, N. Trigoni, and A. Markham, "A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence," *arXiv preprint arXiv:2006.12567*, 2020.

[54] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li, "Lo-net: Deep real-time lidar odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8473–8482.

[55] J. Deng, Q. Wu, X. Chen, S. Xia, Z. Sun, G. Liu, W. Yu, and L. Pei, "Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 8218–8227.

[56] Y. Cho, G. Kim, and A. Kim, "Unsupervised geometry-aware deep lidar odometry," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 2145–2152.

[57] Y. Wu, T. Guadagnino, L. Wiesmann, L. Klingbeil, C. Stachniss, and H. Kuhlmann, "Lio-ekf: High frequency lidar-inertial odometry using extended kalman filters," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 13 741–13 747.

[58] S. Bai, J. Lai, P. Lyu, Y. Cen, and B. Ji, "Improved preintegration method for gnss/imu/in-vehicle sensors navigation using graph optimization," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 11 446–11 457, 2021.

[59] M. Zhao, J. Wang, T. Gao, C. Xu, and H. Kong, "Fmcw-lio: A doppler lidar-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5727–5734, 2024.

[60] C. Xue, Y. Huang, C. Zhao, X. Li, L. Mihaylova, Y. Li, and J. A. Chambers, "A gaussian-generalized-inverse-gaussian joint-distribution-based adaptive msckf for visual-inertial odometry navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 3, pp. 2307–2328, 2022.

[61] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5135–5142.

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

[63] X. Bai, W. Wen, and L.-T. Hsu, "Degeneration-aware outlier mitigation for visual inertial integrated navigation system in urban canyons," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.

[64] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation.* IEEE, 2027, Conference Proceedings, pp. 3565–3572.

[65] T. Li, L. Pei, Y. Xiang, X. Zuo, W. Yu, and T.-K. Truong, "P 3-lins: Tightly coupled ppp-gnss/ins/lidar navigation system with effective initialization," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023.

[66] S. Li, S. Wang, Y. Zhou, Z. Shen, and X. Li, "Tightly coupled integration of gnss, ins, and lidar for vehicle navigation in urban environments," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 24 721–24 735, 2022.

[67] P. Wang, R. Yang, B. Cao, W. Xu, and Y. Lin, "Dels-3d: Deep localization and segmentation with a 3d semantic map," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5860–5869.

[68] L. Han, Y. Lin, G. Du, and S. Lian, "Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2019, pp. 6906–6913.

[69] Y. Jia, H. Luo, F. Zhao, G. Jiang, Y. Li, J. Yan, Z. Jiang, and Z. Wang, "Lvio-fusion: A self-adaptive multi-sensor fusion slam framework using actor-critic method," in *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS).* IEEE, 2021, pp. 286–293.

[70] J. Li, J. Gao, H. Zhang, and T. Z. Qiu, "Rse-assisted lane-level positioning method for a connected vehicle environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2644–2656, 2018.

[71] G. Wang, J. Wu, T. Xu, and B. Tian, "3d vehicle detection with rsu lidar for autonomous mine," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 344–355, 2021.

[72] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2x-vit: Vehicle-to-everything cooperative perception with vision transformer," in *European conference on computer vision.* Springer, 2022, pp. 107–124.

[73] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning.* PMLR, 2017, pp. 1–16.

[74] J. Zhao, H. Xu, H. Liu, J. Wu, Y. Zheng, and D. Wu, "Detection and tracking of pedestrians and vehicles using roadside lidar sensors," *Transportation research part C: emerging technologies*, vol. 100, pp. 68–87, 2019.

[75] W. Zimmer, C. Creß, H. T. Nguyen, and A. C. Knoll, “Tumtraf intersection dataset: All you need for urban 3d camera-lidar roadside perception,” in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 1030–1037.

[76] Y. He, L. Ma, Z. Jiang, Y. Tang, and G. Xing, “Vi-eye: semantic-based 3d point cloud registration for infrastructure-assisted autonomous driving,” in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 573–586.

[77] Y. He, C. Bian, J. Xia, S. Shi, Z. Yan, Q. Song, and G. Xing, “Vi-map: Infrastructure-assisted real-time hd mapping for autonomous driving,” in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023, pp. 1–15.

[78] W. Wen, X. Bai, G. Zhang, S. Chen, F. Yuan, and L.-T. Hsu, “Multi-agent collaborative gnss/camera/ins integration aided by inter-ranging for vehicular navigation in urban areas,” *IEEE access*, vol. 8, pp. 124 323–124 338, 2020.

[79] G. Zhang, H.-F. Ng, W. Wen, and L.-T. Hsu, “3d mapping database aided gnss based collaborative positioning using factor graph optimization,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6175–6187, 2020.

[80] P. Schwarzbach, A. Michler, P. Tauscher, and O. Michler, “An empirical study on v2x enhanced low-cost gnss cooperative positioning in urban environments,” *Sensors*, vol. 19, no. 23, p. 5201, 2019.

[81] L. Zhang, H. Zhao, J. Chen, L. Li, and X. Liu, “Vehicular positioning based on gps/imu data fusion aided by v2x networks,” *IEEE Sensors Journal*, 2024.

[82] G. H. Golub and C. Reinsch, “Singular value decomposition and least squares solutions,” in *Handbook for Automatic Computation: Volume II: Linear Algebra.* Springer, 1971, pp. 134–151.

[83] J. J. Moré, *The Levenberg-Marquardt algorithm: implementation and theory.* Springer, 1978, pp. 105–116.

[84] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2d lidar slam,” in *2016 IEEE international conference on robotics and automation (ICRA).* IEEE, Conference Proceedings, pp. 1271–1278.

[85] L.-T. Hsu, N. Kubo, W. Wen, W. Chen, Z. Liu, T. Suzuki, and J. Meguro, “Urbannav: An open-sourced multisensory dataset for benchmarking positioning algorithms designed for urban areas,” in *Proceedings of the 34th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2021)*, 2021, pp. 226–256.

[86] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.

[87] W. Wen, Y. Zhou, G. Zhang, S. Fahandezh-Saadi, X. Bai, W. Zhan, M. Tomizuka, and L.-T. Hsu, "Urbanloco: a full sensor suite dataset for mapping and localization in urban scenes," in *2020 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, Conference Proceedings, pp. 2310–2316.

[88] E. Li, S. Wang, C. Li, D. Li, X. Wu, and Q. Hao, "Sustech points: A portable 3d point cloud interactive annotation platform system," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1108–1115.

[89] W. Wen, G. Zhang, and L.-T. Hsu, "Exclusion of gnss nlos receptions caused by dynamic objects in heavy traffic urban scenarios using real-time 3d point cloud: An approach without 3d maps," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS).* IEEE, 2018, pp. 158–165.

[90] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[91] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular slam in dynamic environments," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR).* IEEE, 2013, pp. 209–218.

[92] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "Suma++: Efficient lidar-based semantic slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2019, pp. 4530–4537.

[93] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet ++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4213–4220.

[94] P. Pfreundschuh, H. F. Hendrikx, V. Reijgwart, R. Dubé, R. Siegwart, and A. Cramariuc, "Dynamic object aware lidar slam based on automatic generation of training data," in *2021 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, 2021, pp. 11 641–11 647.

[95] D. Duberg, Q. Zhang, M. Jia, and P. Jensfelt, "DUFOMap: Efficient dynamic awareness mapping," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5038–5045, 2024.

[96] R. Qin, J. Tian, and P. Reinartz, "3d change detection–approaches and applications," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 122, pp. 41–56, 2016.

[97] D. Yoon, T. Tang, and T. Barfoot, "Mapless online detection of dynamic objects in 3d lidar," in *2019 16th Conference on Computer and Robot Vision (CRV).* IEEE, 2019, pp. 113–120.

[98] J. Schauer and A. Nüchter, "The peopleremover—removing dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid," *IEEE robotics and automation letters*, vol. 3, no. 3, pp. 1679–1686, 2018.

[99] G. Kim and A. Kim, "Remove, then revert: Static point cloud map construction using multiresolution range images," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2020, pp. 10 758–10 765.

[100] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 529–10 538.

[101] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular slam in dynamic environments," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR).* IEEE, 2013, pp. 209–218.

[102] R. B. Langley *et al.*, "Dilution of precision," *GPS world*, vol. 10, no. 5, pp. 52–59, 1999.

[103] R. H. Rasshofer, M. Spies, and H. Spies, "Influences of weather phenomena on automotive laser radar systems," *Advances in radio science*, vol. 9, pp. 49–60, 2011.

[104] A. Williams, S. Barrus, R. K. Morley, and P. Shirley, "An efficient and robust ray-box intersection algorithm," in *ACM SIGGRAPH 2005 Courses*, 2005, pp. 9–es.

[105] N. Sünderhauf and P. Protzel, "Switchable constraints for robust pose graph slam," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2012, pp. 1879–1884.

[106] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[107] N. Inc, "Waypoint inertial explorer 8.80 post processing software," 2020.

[108] M. Vel'as, M. Španěl, Z. Materna, and A. Herout, "Calibration of rgb camera with velodyne lidar," 2014.

[109] X. Niu, K. Yan, T. Zhang, Q. Zhang, H. Zhang, and J. Liu, "Quality evaluation of the pulse per second (pps) signals from commercial gnss receivers," *GPS solutions*, vol. 19, pp. 141–150, 2015.

[110] P. Zhou, X. Guo, X. Pei, and C. Chen, "T-loam: Truncated least squares lidar-only odometry and mapping in real time," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[111] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, "Factor graph optimization for gnss/ins integration: A comparison with the extended kalman filter," *NAVIGATION: Journal of the Institute of Navigation*, vol. 68, no. 2, pp. 315–331, 2021.

[112] A. Angrisano, S. Gaglione, and C. Gioia, "Performance assessment of gps/glonass single point positioning in an urban environment," *Acta Geodaetica et Geophysica*, vol. 48, pp. 149–161, 2013.

[113] H.-F. Ng and L.-T. Hsu, "3d mapping database-aided gnss rtk and its assessments in urban canyons," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 5, pp. 3150–3166, 2021.

[114] R. Hu, P. Xu, Y. Zhong, and W. Wen, "pyrtklib: An open-source package for tightly coupled deep learning and gnss integration for positioning in urban canyons," *arXiv preprint arXiv:2409.12996*, 2024.

[115] N. C. Talbot, "Centimeters in the field, a users perspective of real-time kinematic positioning in a production environment," in *Proceedings of the 6th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1993)*, 1993, pp. 1049–1057.

[116] B. Li, T. Liu, L. Nie, and Y. Qin, "Single-frequency gnss cycle slip estimation with positional polynomial constraint," *Journal of geodesy*, vol. 93, pp. 1781–1803, 2019.

[117] Z. Liu, "A new automated cycle slip detection and repair method for a single dual-frequency gps receiver," *Journal of Geodesy*, vol. 85, pp. 171–183, 2011.

[118] D. Kim and R. B. Langley, "Instantaneous real-time cycle-slip correction for quality control of gps carrier-phase measurements," *Navigation*, vol. 49, no. 4, pp. 205–222, 2002.

[119] J. Zhao, M. Hernández-Pajares, Z. Li, L. Wang, and H. Yuan, "High-rate doppler-aided cycle slip detection and repair method for low-cost single-frequency receivers," *Gps solutions*, vol. 24, pp. 1–13, 2020.

[120] S. B. Bisnath, "Efficient, automated cycle-slip correction of dual-frequency kinematic gps data," in *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000)*, 2000, pp. 145–154.

[121] Z. L. Yang Gao, "Cycle slip detection and ambiguity resolution algorithms for dual-frequency gps data processing," *Marine Geodesy*, vol. 22, no. 3, pp. 169–181, 1999.

[122] G. Blewitt, "An automatic editing algorithm for gps data," *Geophysical research letters*, vol. 17, no. 3, pp. 199–202, 1990.

[123] G. Xu, Y. Xu, G. Xu, and Y. Xu, "Applications of gps theory and algorithms," *GPS: Theory, Algorithms and Applications*, pp. 313–340, 2016.

[124] C. Altmayer, "Enhancing the integrity of integrated gps/ins systems by cycle slip detection and correction," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No. 00TH8511).* IEEE, 2000, pp. 174–179.

[125] T. Takasu and A. Yasuda, "Cycle slip detection and fixing by mems-imu/gps integration for mobile environment rtk-gps," in *Proceedings of the 21st international technical meeting of the satellite division of the Institute of Navigation (ION GNSS 2008)*, 2008, pp. 64–71.

[126] X. Zhang, F. Zhu, Y. Zhang, F. Mohamed, and W. Zhou, "The improvement in integer ambiguity resolution with ins aiding for kinematic precise point positioning," *Journal of Geodesy*, vol. 93, pp. 993–1010, 2019.

[127] W. Geiger, J. Bartholomeyczik, U. Breng, W. Gutmann, M. Hafen, E. Handrich, M. Huber, A. Jackle, U. Kempfer, H. Kopmann *et al.*, "Mems imu for ahrs applications," in *2008 IEEE/ION Position, Location and Navigation Symposium.* IEEE, 2008, pp. 225–231.

[128] J. A. Slater and S. Malys, "Wgs 84—past, present and future," in *Advances in Positioning and Reference Frames: IAG Scientific Assembly Rio de Janeiro, Brazil, September 3–9, 1997.* Springer, 1998, pp. 1–7.

[129] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems.* Artech House, jan 2008.

[130] P. Teunnissen, "The least-square ambiguity decorrelation adjustment: a method for fast gps integer ambiguity estimation," *J. Geodesy*, vol. 70, no. 1, pp. 65–82, 1995.

[131] W. Gurtner, "Rinex: The receiver-independent exchange format," *GPS world*, vol. 5, no. 7, pp. 48–52, 2000.

[132] K. Chen, G. Chang, and C. Chen, "Ginav: A matlab-based software for the data processing and analysis of a gnss/ins integrated navigation system," *GPS solutions*, vol. 25, no. 3, p. 108, 2021.

[133] W. Gao, Z. Sun, M. Zhao, C.-Z. Xu, and H. Kong, "Active loop closure for osm-guided robotic mapping in large-scale urban environments," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 302–12 309.

[134] L. Chang, X. Niu, T. Liu, J. Tang, and C. Qian, "Gnss/ins/lidar-slam integrated navigation system based on graph optimization," *Remote Sensing*, vol. 11, no. 9, p. 1009, 2019.

[135] C. Xu, H. Zhang, and J. Gu, "Scan context 3d lidar inertial odometry via iterated eskf and incremental k-dimensional tree," in *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2022, pp. 21–27.

[136] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Globally consistent and tightly coupled 3d lidar inertial mapping," in *2022 international conference on robotics and automation (ICRA)*. IEEE, 2022, pp. 5622–5628.

[137] H. K. A. Science and T. R. I. C. Limited, "C-v2x technology," https://www.astri.org/tdprojects/connected-vehicle-v2x-technology/, 2023, accessed: September 15, 2024.

[138] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[139] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[140] Z. Pang, Z. Li, and N. Wang, "Simpletrack: Understanding and rethinking 3d multi-object tracking," in *European Conference on Computer Vision*. Springer, 2022, pp. 680–696.

[141] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.

[142] M. B. Wright, "Speeding up the hungarian algorithm," *Computers & Operations Research*, vol. 17, no. 1, pp. 95–96, 1990.

[143] S. Gu, C. Dai, W. Fang, F. Zheng, Y. Wang, Q. Zhang, Y. Lou, and X. Niu, "Multi-gnss ppp/ins tightly coupled integration with atmospheric augmentation and its application in urban vehicle navigation," *Journal of Geodesy*, vol. 95, no. 6, p. 64, 2021.

[144] X. Liu, W. Wen, and L.-T. Hsu, "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*, 2023.

[145] Deloitte and ASTRI, "Hong kong connected & autonomous vehicle (cav) development study," https://www2.deloitte.com/cn/zh/pages/technology/articles/hk-cav-development-study.html, Deloitte, 2024, accessed: April 18, 2024.

[146] Y. Zhong, W. Wen, and L.-T. Hsu, "Towards accurate vehicle-to-pedestrian relative positioning aided by inter-frame and inter-agent gnss measurement collaboration using factor graph optimization for smart summon," in *Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023)*, 2023, pp. 3104–3116.

[147] P. Tétreault, J. Kouba, P. Héroux, and P. Legree, "CSRS-PPP: An Internet Service for GPS User Access to the Canadian Spatial Reference Frame," *Geomatica*, vol. 59, no. 1, pp. 17–28, 2005.

[148] C. Chi, X. Zhang, J. Liu, Y. Sun, Z. Zhang, and X. Zhan, "Gici-lib: a gnss/ins/camera integrated navigation library," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 7970–7977, 2023.

[149] T. G. S. S. of Survey and H. K. L. D. Mapping Office, "Web download of satref gnss raw data (rinex format)," https://www.geodetic.gov.hk/en/rinex/downv.aspx/, 2025, accessed: March 15, 2025.

[150] Y. Zhong, W. Wen, and L.-T. Hsu, "Trajectory smoothing using gnss/pdr integration via factor graph optimization in urban canyons," *IEEE Internet of Things Journal*, 2024.

[151] M.-S. Kang, J.-H. Ahn, J.-U. Im, and J.-H. Won, "Lidar-and v2x-based cooperative localization technique for autonomous driving in a gnss-denied environment," *Remote Sensing*, vol. 14, no. 22, p. 5881, 2022.

[152] A. Alhilal, B. Finley, T. Braud, D. Su, and P. Hui, "Distributed vehicular computing at the dawn of 5g: A survey," *arXiv preprint arXiv:2001.07077*, 2020.

[153] A. Boubakri, S. M. Gammar, M. B. Brahim, and F. Filali, "High definition map update for autonomous and connected vehicles: A survey," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2022, pp. 1148–1153.

[154] G. Zhang, B. Xu, H.-F. Ng, and L.-T. Hsu, "Gnss rums: Gnss realistic urban multiagent simulator for collaborative positioning research," *Remote Sensing*, vol. 13, no. 4, p. 544, 2021.

[155] T. Shan, B. Englot, C. Ratti, and D. Rus, "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 5692–5698.

[156] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.