3D LIDAR AIDED GNSS POSITIONING AND ITS APPLICATION IN SENSOR FUSION FOR AUTONOMOUS VEHICLES IN URBAN CANYONS

WEISONG WEN

PhD

The Hong Kong Polytechnic University

2021

The Hong Kong Polytechnic University

Department of Mechanical Engineering

3D LIDAR AIDED GNSS POSITIONING AND ITS APPLICATION IN SENSOR FUSION FOR AUTONOMOUS VEHICLES IN URBAN CANYONS

Weisong Wen

A thesis submitted in partial fulfillment

of the requirements for the degree

of

Doctor of Philosophy

July 2020

CERTIFICATE OF ORIGINALITY

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

____________________(Signed)

Wen Weisong________(Name)

ABSTRACT

Autonomous driving is well believed to be the potential solution for reducing excessive accidents and alleviating severe traffic congestions. Localization is the key and fundamental part of the robust operation of autonomous vehicles. Moreover, centimeter-level globally referenced positioning is required. Global Navigation Satellite System (GNSS) is currently an indispensable source that can provide absolute positioning information. Satisfactory performance (5~10 meters) can be obtained in open space if a decent sky view is available. However, its performance can be suffered due to the blockage and reflection from environment features in the super-urbanized area, causing the well-known multipath effects and non-line-of-sight (NLOS) receptions. In this dissertation, we present several 3D light detection and ranging (LiDAR) aided GNSS positioning methods which aims to solve the problems caused by GNSS NLOS receptions. Different from the conventional 3D mapping aided GNSS (3DMA GNSS), we proposed to leverage the onboard sensing based on 3D LiDAR to detect the polluted GNSS NLOS signals. Then the detected NLOS signals are excluded, remodeled or even corrected before its application in GNSS positioning or integration with other sensors. First, we present a novel GNSS NLOS exclusion method caused by dynamic objects using LiDAR perception. The surrounding dynamic objects are detected based on 3D point clouds. Then the NLOS signals caused by the blockage from dynamic objects are identified based on the detected dynamic objects. The proposed method relaxes the drawback of the 3DMA GNSS which can not help to detect the NLOS caused by dynamic objects. Second, instead of excluding all the detected GNSS NLOS signals, we introduced a novel NLOS correction method with the aid of 3D LiDAR and building height information. The proposed method moves forward a step and solves the problem of the distortion of satellite geometry distribution caused by excessive NLOS exclusion. Thirdly, a more general solution to make use of the detected GNSS NLOS is proposed by remodeling the detected GNSS NLOS. In addition, the integration of GNSS positioning with NLOS modeling with the LIDAR odometry is presented. Fourth, a sliding window map (SWM) is proposed to remodel or correct the GNSS NLOS signals. The proposed method alleviates the dependence of building height information. In addition, the field of view (FOV) of 3D LiDAR point clouds is significantly enhanced with the help of the SWM. Moreover, a general GNSS signal validation and calibration pipeline are employed to ensure the feasibility of GNSS measurements before its integration with inertial navigation system using state-of-the-art factor graph optimization (FGO). All of the four proposed methods are validated using real data collected in urban canyons of Hong Kong. The improved accuracy shows the effectiveness of the proposed method in GNSS positioning in urban canyons.

PUBLICATIONS ARISING FROM THE THESIS

JOURNAL PAPERS

1. **Wen Weisong**., Xiwei Bai., Zhang, Guohao., Shengdong Chen., Feng Yuan., Hsu, L.T.*. Multi-Agent Collaborative GNSS/Camera/INS Integration Aided by Inter-ranging for Vehicular Navigation in Urban Areas, *IEEE Access*, 2020,
2. **Wen W**., Hsu, L.T.* (2020) GraphGNSSLib: An Open-source Package for GNSS Single Point Positioning and Real-time Kinematic Using Factor Graph Optimization, *GPS Solutions*. (SCI. 2019 IF. 3.049, Ranking 7.7%) [**To be Submitted**]
3. Jiang Yue, **Wen Weisong***., Jin Han., Hsu, L.T. 3D Point Clouds Data Super Resolution Aided LiDAR SLAM for Vehicular Positioning in Urban Canyons, *IEEE Transactions on Vehicular Technology*, 2020, (SCI. 2019 IF. 3.019, Ranking 10.7%) [**First round revision**]
4. **Wen Weisong**., Tim Pfeifer., Hsu, L.T.* Robust GNSS/LiDAR Integration Aided by Self-adaptive Gaussian Mixture Models for Vehicular Positioning in Urban Scenarios, *IEEE Transactions on Aerospace and Electronic Systems*, 2020, [**Submitted**]
5. **Wen Weisong**., Tim Pfeifer., Xiwei Bai., Hsu, L.T.* Comparison of Extended Kalman Filter and Factor Graph Optimization for GNSS/INS Integrated Navigation System, *The Journal of Navigation*, 2020, (SCI. 2019 IF. 3.019, Ranking 10.7%) [**First round revision**]
6. Xiwei Bai., **Wen Weisong**., Hsu, L.T.* Robust Visual-Inertial Integrated Navigation System Aided by Online Sensor Model Adaption for Autonomous Ground Vehicles in Urban Areas, *Remote Sensing*, 2020, (SCI. 2019 IF. 3.019, Ranking 10.7%)
7. Bai, Xiwei, **Weisong Wen***, and Li-Ta Hsu. "Using Sky-Pointing Fish-eye Camera and LiDAR to Aid GNSS Single Point Positioning in Urban Canyons." IET Intelligent Transport Systems (2020).
8. Zhang, Guohao, Hoi-Fung Ng, **Weisong Wen**, and Li-Ta Hsu. "3D Mapping Database Aided GNSS Based Collaborative Positioning Using Factor Graph Optimization." IEEE Transactions on Intelligent Transportation Systems (2020). (SCI. 2019 IF. 5.744, Ranking 7.7%)
9. Zhang, Guohao, **Weisong Wen**, Bing Xu, and Li-Ta Hsu. "Extending Shadow Matching to Tightly-coupled GNSS/INS Integration System." IEEE Transactions on Vehicular Technology (2020). (SCI. 2019 IF. 5.744, Ranking 7.7%)
10. **Wen Weisong**., Bai Xiwei., Kan Y.C., Hsu, L.T.* Tightly Coupled GNSS/INS Integration Via Factor Graph and Aided by Fish-eye Camera, *IEEE Transactions on Vehicular Technology*, 2019, (SCI. 2019 IF. 5.339, Ranking 7.7%)
11. **Weisong Wen**, Guohao Zhang and Li-Ta Hsu, GNSS NLOS Exclusion Based on Dynamic Object Detection Using LiDAR Point Cloud, *IEEE Transactions on Intelligent Transportation Systems*, 2019, (SCI. 2019 IF. 5.744, Ranking 7.7%)
12. **Weisong Wen**, Guohao Zhang and Li-Ta Hsu, Correcting NLOS by 3D LiDAR and Building Height to Improve GNSS Single Point Positioning, *The Journal of Navigation 61.1 (2019)*, (SCI. 2019 IF. 3.019, Ranking 10.7%)
13. **Weisong Wen**, Guohao Zhang and Li-Ta Hsu, Object Detection Aided GNSS and Its Integration with LiDAR in Highly Urbanized Areas, *IEEE Intelligent Transportation Systems Magazine*, 2019, (SCI. 2016 IF. 3.294, Ranking 10.7%) [**Accepted**]
14. Zhang G., **Wen W**., Hsu, L.T.* (2019) Rectification of GNSS based Collaborative Positioning using 3D Building Models in Urban Areas, *GPS Solutions* 23(3):83. (SCI. 2019 IF. 3.049, Ranking 7.7%)
15. **Wen, Weisong**, Li-Ta Hsu, and Guohao Zhang. "Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong." *Sensors* 18.11 (2018): 3928. **(SCI. 2018 IF. 3.031)**
16. **Wen, Weisong**, Xiwei Bai, Wei Zhan, Masayoshi Tomizuka, and Li-Ta Hsu. "Uncertainty estimation of LiDAR matching aided by dynamic vehicle detection and high definition map." *Electronics Letters* (**SCI. 2017 IF. 1.232**).

INTERNATIONAL CONFERENCE PAPERS

1. Wenqi Fang, Huiyun Li, Shaobo Dang, Hui Huang, Li-TaHsu, **Weisong Wen**, Combining deep gaussian process and rule-based method for decision-making in self- driving simulation with small data, *2019 International Conference on Computational Intelligence and Security (CIS'2019)*, Macau, China, December 13-16, 2019.
2. Xiwei Bai, **Weisong Wen**, Li-Ta Hsu and Huiyun Li, Perception-aided Visual/Inertial Integrated Positioning in Highly Dynamic Urban Areas, *IEEE/ION PLANS 2020*, Portland, Oregon, USA. (To be presented)
3. Jiang Yue, **Weisong Wen**, Jin Han and Li-Ta Hsu, LiDAR Data Enrichment using Deep Leaning based on High-Resolution Image: An approach to achieve High Performance LiDAR SLAM using Low-cost LiDAR, *IEEE/ION PLANS 2020*, Portland, Oregon, USA. (To be presented)
4. **Weisong Wen**, Tim Pfeifer, Xiwei Bai and Li-Ta Hsu, GNSS/LiDAR Integration Aided by Self-adaptive Gaussian Mixture Model in Urban Scenarios: An Approach Robust to Non-Gaussian Noise, *IEEE/ION PLANS 2020*, Portland, Oregon, USA. (To be presented)
5. **Weisong Wen**, Yiyang Zhou, Guohao Zhang, Saman Fahandezh-Saadi, Xiwei Bai, Wei Zhan, Masayoshi Tomizuka, and Li-Ta Hsu, UrbanLoco: A Full Sensor Suite Dataset for Mapping and Localization in Urban Scenes (***submitted***), *ICRA 2020*, Paris, France.
6. **Weisong Wen**, Xiwei Bai, and Li-Ta Hsu, AGPC-SLAM: Absolute Ground Plane Constrained 3D LiDAR SLAM. (***submitted***), *ICRA 2020*, Paris, France.
7. **Weisong Wen**, Kan Y.C and Li-Ta Hsu, Performance Comparison of GNSS/INS Integrations based on EKF and Factor Graph Optimization, *ION GNSS+ 2019*, Miami, Florida, USA.
8. Xiwei Bai, **Weisong Wen***, Li-Ta Hsu. Performance Analysis of Visual/Inertial Integrated Positioning in Typical Urban Scenarios of Hong Kong. *Proceedings of 2019 Asian-Pacific Conference on Aerospace Technology and Science*, August 28-31, 2019, Taiwan.
9. **Weisong Wen**, Li-Ta Hsu. GNSS/INS Integration Based on Graph Optimization for Vehicular Positioning. *Proceedings of 2019 Asian-Pacific Conference on Aerospace Technology and Science*, August 28-31, 2019, Taiwan.
10. **Weisong Wen**, Wei Zhan and Li-Ta Hsu, Robust Localization Using 3D NDT Matching and Beam Model for Autonomous Vehicles in an Urban Scenario with Dynamic Obstacles, *MMT 2019,* Shenzhen, Guangdong, China.
11. Xiwei Bai, **Weisong Wen***, Guohao Zhang, Li-Ta Hsu. " Real-time GNSS NLOS Detection and Correction Aided by Sky-Pointing Camera and 3D LiDAR." *Proceedings of the ION 2019 Pacific PNT Meeting*.
12. **Weisong Wen**, Xiwei Bai, and Li-Ta Hsu, Multi-sensor Integration-based Localization for Autonomous Vehicle in Urbanized Area. *The 8th East Asia Mechanical and Aerospace Engineering Workshop*, December 1-3, 2018, Hong Kong SAR, China.
13. Sai Hu, **Weisong Wen**, Yin-Chiu Kan and Li-Ta Hsu, Low-speed Autonomous Vehicle for Campus Delivery Application. *The 8th East Asia Mechanical and Aerospace Engineering Workshop*, December 1-3, 2018, Hong Kong SAR, China.
14. Guohao Zhang**, Weisong Wen**, and Li-Ta Hsu, Collaborative GNSS positioning with the aids of 3D city models, *ION GNSS+ 2018*, Miami, Florida, USA.
15. **Weisong Wen**, Guohao Zhang and Li-Ta Hsu, Correcting GNSS NLOS by 3D LiDAR and Building Height, *ION GNSS+ 2018*, Miami, Florida, USA.
16. **Weisong Wen**, Guohao Zhang and Li-Ta 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, *IEEE/ION PLANS 2018*, Monterey, CA, USA.
17. Guohao Zhang, **Weisong Wen**, and Li-Ta Hsu, A Novel GNSS based V2V Cooperative Localization to Exclude Multipath Effect using Consistency Checks, *IEEE/ION PLANS 2018*, Monterey, CA, USA.

MAGZINE ARTICLES

1. **Weisong Wen** and Li-ta Hsu, Perceived Environment Aided GNSS Single Point Positioning: An Example using LiDAR Scanner, Inside GNSS Magazine, July-August 2019. (Link)

BOOK CHAPTERS

1. Vehicular Networks: Applications, Performance Analysis and Challenges. Chapter 2. Vehicular Positioning System (Li-Ta Hsu, **Weisong Wen** and Guohao Zhang, Interdisciplinary Division of Aeronautical and Aviation Engineering, Hong Kong Polytechnic University, Hong Kong, China)

Innovation and Entrepreneurship Awards

1. (**First Prize in Hong Kong Section**) Wen W. and Hsu, Li-Ta, 〞Low-cost and Accurate Localization Solution for Autonomous Vehicles〞 in Qianhai-Guangdong-Macao Youth Innovation and Entrepreneurship Competition, Shenzhen, Guangdong, China.
2. (**Excellent Project Award**) Wen W. and Hsu, Li-Ta 〞Lane-level navigation system for autonomous driving vehicle〞 in Songshan Lake Innovation and Entrepreneurship Competition, Dongguan, Guangdong, China.

Number of Papers Published

Academic Conference: 9 **first-authored conference papers**
Journals: **7 first-authored papers** published in Journals/Transactions, **5 co-authored papers** published in Journals/Transactions, **3 first-authored papers** being submitted to Journals/Transactions,

ACKNOWLEDGEMENTS

The three years of Ph.D. study is near the end. At this time, I would like to thank all the teachers, classmates and friends who have helped me in this long journey. The knowledge and encouragement from you have always motivated me toward this step. Firstly, I want to show my sincere appreciation to my supervisor, Dr. Li-ta Hsu. He patiently guides me through the navigation and positioning field and impressed me with professional and logical thinkings. On the one hand, Dr. Hsu taught me how to see a problem from a research aspect and model it mathematically. On another hand, Dr. Hsu taught me how to efficiently share the research achievement with other researchers in the same fields by submitting papers to journals or presenting in top academic conferences globally. All these trainings significantly improved my problem-solving capability. His patient direction and encouragement will affect me throughout my life. Second, I want to thank my colleges, Guohao Zhang, Dr. Yue Jiang and so on, who helped me a lot in my research. Finally, I want to show my sincere appreciation to my family members who gave me the courage to go through diverse difficulties. I love you all!

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Background

At present, autonomous driving poses great attention due to its potential application to improve the quality of intelligent transportation systems (ITS). Localization for autonomous driving has been extensively studied since the Defence Advanced Research Projects Agency (DARPA) urban challenge [1] in 2007. Localization is a significant part of the operation of the safety-critical autonomous driving [1, 2]. In general, globally referenced centimeter-level positioning is required for the robust operation of autonomous vehicles.

Global navigation satellite system (GNSS) is, in fact, currently an indispensable source that can provide absolute positioning relative to the earth. The low-cost GNSS receiver with single point positioning (SPP) can obtain absolute positioning with an accuracy of 3~5 meters in open space [3]. GNSS solution possesses increased popularity because of the availability of multi-constellation global satellite navigation systems (GPS, Beidou, GLONASS, Galileo, and QZSS). GNSS positioning can obtain decent performance if the GNSS receiver receives enough direct signals transmitted from satellites, so-called line-of-sight (LOS) [4]. However, the GNSS signal propagation can be reflected, diffracted or blocked by skyscrapers and moving objects [5] in urbanized areas, such as Hong Kong, which can cause additional signal transmission delay. Thus, it introduces pseudorange errors due to both multipath effects and none-light-of-sight (NLOS) receptions [4] (see Figure 1-1), which can present a positioning error of even more than 50 meters in deep urban canyons [3, 6]. Figure 1-2 shows an example performance test of GNSS SPP using a commercial level GNSS receiver. According to a recent review paper [7], the NLOS is one of the major problems preventing the application of GNSS in urban canyon. Utilizing 3D building models to detect the NLOS is straightforward. NLOS satellites can be detected with the aid of 3D models of buildings, and can then be excluded from use in GNSS positioning [8, 9]. However, NLOS exclusion will distort the geometric distribution of the satellites, degrading accuracy [10], and even resulting in too few satellites for further GNSS calculation [10]. Moreover, these methods require the availability of 3D building models of the environment, and the performance of NLOS detection relies on the accuracy of an initial guess of the GNSS receiver’s position. In addition, NLOS reception caused by dynamic objects cannot be detected as well. A well-known method, GNSS shadow matching, was developed to match measured satellite visibility with the predicted satellite visibility of hypothesized positions [11-13]. However, the performance of shadow matching is dependent upon the quality of satellite visibility classification and the initial guess as to the position of the GNSS receiver. A likelihood-based 3DMA GNSS method, which modeled the measurement uncertainty, and used this value to mitigate NLOS effects, has been proposed to provide accurate positioning in the along-street direction [14]. Due to the complementarity of the shadow matching and likelihood-based 3DMA GNSS, approaches to the integration of these approaches have recently been studied [15]. Another range-based 3DMA GNSS method is to correct the NLOS affected measurements for GNSS positioning [16-19]. These methods were proposed to simulate signal transmission routes, using a ray-tracing method. However, these ray-tracing-based 3DMA GNSS methods have the drawbacks of requiring stringency in 1) the accuracy of 3D mapping databases; 2) the initial guess of receiver positions, and 3) the computational power of the processors due to the ray-tracing process. Recently, a skymask-based NLOS correction method has been proposed [20], which effectively reduces the computational load incurred by ray-tracing-based methods [16-19]. However, these 3D mapping-aided GNSS methods have the drawbacks of 1) reliance on the availability of 3D building models; 2) inability to mitigate NLOS receptions caused by surrounding dynamic objects.

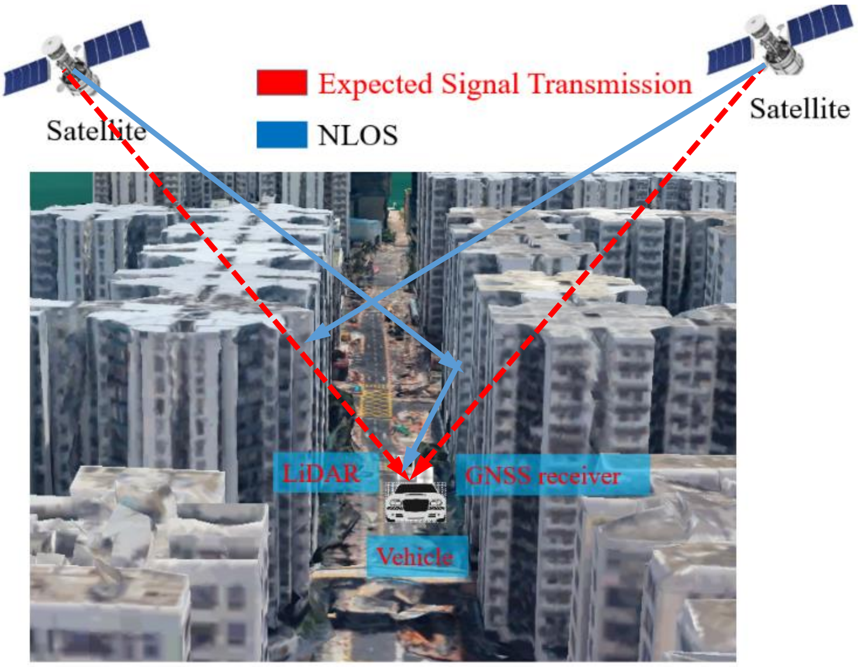

Figure 1-1 Illustration of the GNSS NLOS signals in urban canyon. The red dash line denotes the expected signal transmission route for LOS without any blockage or reflection. The blue curve denotes the reflected signal transmission route for NLOS.

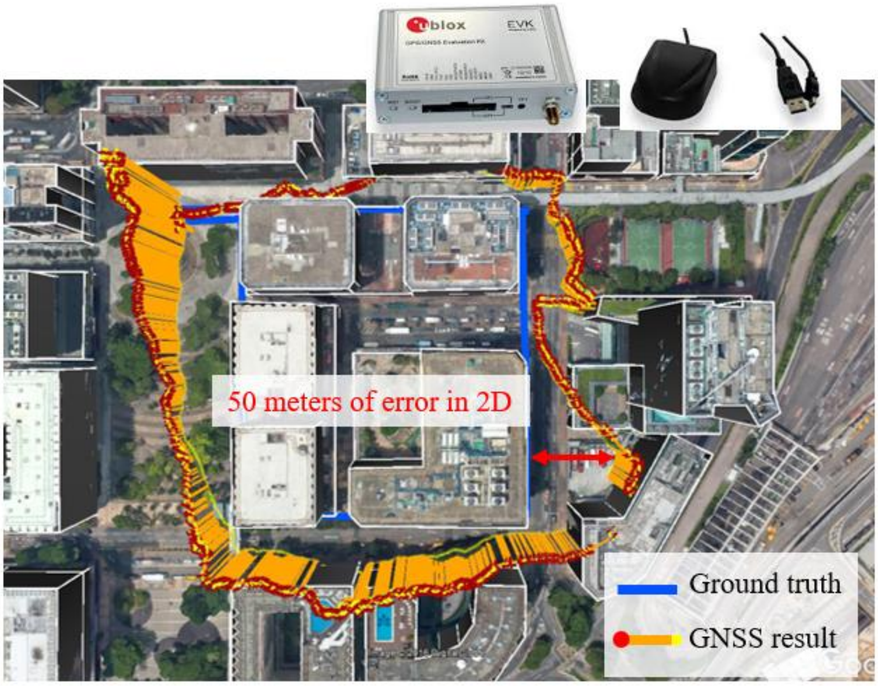

Figure 1−2 Illustration of the GNSS SPP positioning performance in an urban canyon. The 2D mean error can reach 50 meters using a commercial level GNSS receiver.

1.2 Research Objectives and Thesis Contribution

The objective of this project is to solve the problem of GNSS NLOS receptions in urban canyons. As mentioned in Section 1.1, the popular 3DMA GNSS method has two drawbacks of reliance on the availability of the 3D building model and the inability of detecting the NLOS caused by dynamic objects. The detection of GNSS NLOS signals relies on the environment description. Inspired by the fast development of perception capability, which can help to obtain the environmental description in a real-time manner, we proposed to make use of the onboard sensing to help the GNSS positioning in urban canyon. Different from the conventional 3D mapping aided GNSS (3DMA GNSS), we proposed to leverage the environment description derived from 3D LiDAR to detect the polluted GNSS NLOS signals. Then the detected NLOS signals are excluded, remodeled or even corrected before its application in GNSS positioning or integration with other sensors. The contributions of this thesis are as follows:

1) We firstly proposed to make use of the 3D point clouds-based object detection to detect the GNSS NLOS. Then the detected NLOS receptions are excluded from further GNSS positioning. This is currently the first work which proposes to mitigates the effects of NLOS caused by dynamic objects.
2) A GNSS NLOS detection and correction method aided by 3D LiDAR and building heights information is proposed. Instead of directly excluding all the detected GNSS NLOS receptions, a novel GNSS pseudorange correction model is employed to estimate the transmission delay. Therefore, we relaxed the limitation of the previous work where the NLOS receptions are excluded and lead to poor satellite geometry. The proposed method does not rely on the 3D building information.
3) We explore the potential of the proposed method in GNSS/LiDAR integration. The GNSS NLOS receptions are detected and remodeled before its integration with LiDAR odometry. Significantly improved performance is obtained compared with the conventional GNSS/LiDAR integration.
4) A sliding window map (SWM) is proposed to remodel or correct the GNSS NLOS signals. The proposed method alleviates the dependence of building height information. In addition, the field of view (FOV) of 3D LiDAR point clouds is significantly enhanced with the help of the SWM. Moreover, a general GNSS signal validation and calibration pipeline are employed to ensure the feasibility of GNSS measurements before its integration with inertial navigation system using state-of-the-art factor graph optimization (FGO).

2. OBJECT DETECTION AIDED GNSS NLOS EXCLUSION

2.1 Introduction

To achieve fully autonomous driving in highly urbanized areas, absolute lane-level positioning is required. Light detection and ranging (LiDAR), cameras and inertial navigation systems (INS) are usually integrated with GNSS positioning [21-23]. However, the three positioning sources can only conduct relative positioning. The GNSS/INS/LiDAR/ high definition map-based [21] positioning is the potential solution to provide robust and constant localization information for autonomous driving. In this solution, the searching for the initial position in the existing high definition map is essential. GNSS is currently the key source providing the initial absolute positioning information. GNSS solution possesses increased popularity because of the availability of multi-constellation global navigation satellite systems (GPS, BeiDou, GLONASS, Galileo, and QZSS). GNSS positioning can obtain decent performance if the GNSS receiver receives enough direct signals transmitted from satellites, so-called line-of-sight (LOS) [24]. However, the GNSS propagation may be reflected, diffracted or blocked by skyscrapers and moving objects in highly urbanized areas, such as Hong Kong, which can cause additional signal transmission delay. Thus, it introduces pseudorange errors as both multipath effects and non-light-of-sight (NLOS) receptions can cause a positioning error of more than 100 meters in deep urban canyons [3, 6].

According to a recent paper survey, NLOS is the main challenge for GNSS urban localization [7]. In nature, buildings reflect and block the GNSS signal causing NLOS. Thus, the use of a 3D building model to aid GNSS positioning is straightforward [25]. Based on the 3D building model, the NLOS measurement can be excluded from GNSS positioning if the prior knowledge of the user position is available [8, 9, 18]. However, excluding all the NLOS measurements will result in a distortion of satellite distribution. This large dilution of precision (DOP) will enlarge the GNSS positioning error in a lateral direction which is not welcomed for the application of intelligent transportation systems [26, 27]. One of the most famous methods is called GNSS shadow matching, which is to compare the measured satellite visibilities with that of a group of hypothesized positions [12, 28, 29]. This method greatly improves the positioning performance in the lateral direction by using the NMEA level of the GNSS message [13]. To further improve the positioning accuracy, a ray-tracing simulation is conducted to track the signal transmission paths (both direct and reflect) within the 3D building models [17]. By this ray-tracing simulation, the NLOS is not only detected but also corrected [30, 31]. GNSS shadow matching and the range-based 3D maps aided positioning method are combined to optimize the GNSS positioning result in urban canyons [32]. Recently, the researches to waive the 3D building model with different levels of prior information of the city is studied [33, 34].

The 3D laser scanner is also used to construct the point cloud-based 3D geographic information of buildings, so-called 3D point cloud map. The 3D point cloud map is employed to detect the visibilities of satellites [35]. To better model the reliability of GNSS positioning, horizontal dilution of precision (HDOP) is calculated using the remaining satellites and signal noise ratio (SNR) is reconsidered to estimate the noise covariance of GNSS positioning. Deep coupling of GNSS and LiDAR is also researched [36, 37]. Another method to detect the NLOS receptions is the tight coupling of the GNSS receiver and omnidirectional camera [38, 39]. Skyline of buildings in the urbanized area can be detected to identify the NLOS satellites. However, this method cannot get satisfactory improvement in conditions with strong light, as the computer vision-based skyline detection is sensitive to light conditions.

Generally, these 3DMA GNSS can only mitigate multipath effects and NLOS receptions caused by buildings modeled in the 3D city maps. However, moving objects with a tall height, such as the double-decker bus [40] whose height can reach 4.5 meters, can also cause NLOS receptions. In particular, highly urbanized cities such as New York, London and Hong Kong possessing numerous double-decker buses in the streets, which can introduce considerable errors into the GNSS pseudorange measurements. This GNSS positioning error caused by moving objects cannot be mitigated by the novel 3DMA GNSS. To obtain better GNSS positioning performance for autonomous driving, this is a significant issue that needs to be considered.

In this section, we propose to exclude the NLOS receptions caused by moving objects in heavy traffic urban scenarios using real-time 3D point cloud generated by LiDAR. The multiple-channel LiDAR is widely used in autonomous driving vehicles [41, 42] and is employed to provide distance information of surrounding environments. This paper innovatively takes advantage of LiDAR-based perception to assist the performance of GNSS positioning. In general, object detection is consisted of two parts, clustering (segmentation) [43-45] and classification [46, 47]. In this Section, the Euclidean clustering algorithm is employed to do clustering. The parameters-based method is presented for the double-decker bus classification. In this case, the dimension and position of the dynamic object relative to the GNSS receiver are calculated by object detection. Based on the boundaries of the detected object, NLOS exclusion can be implemented with our proposed exclusion algorithm. Moreover, the initial position of the GNSS receiver, which is significant for 3DMA GNSS, is also not required in our proposed NLOS exclusion algorithm. Finally, GNSS positioning result is calculated based on the surviving satellites. To the best of authors' knowledge, this paper is the first attempt to employ the object detection algorithm to depict the dynamic objects in GNSS Skyplot with the aid of 3D LiDAR equipped on the vehicle. In other words, this paper proposes an idea to employ LiDAR perception to aid GNSS absolute positioning.

The remainder of this Section 2 is structured as follows. An overview of the proposed method is given in Section 2.2. Section 2.3 discusses the double-decker bus detection method using the Euclidean clustering algorithm and parameter-based classification method. The coordinate transformation from LiDAR coordinate system to the GNSS Skyplot coordinate system is also presented in Section 2.3. In Section 2.4, the NLOS exclusion criterion is proposed and satellite exclusion is implemented. Then, GNSS WLS positioning is introduced. In Section 2.5, we evaluate the effectiveness of the proposed method utilizing both static and dynamic experiments. Finally, conclusions are presented.

2.2 Overview of the Proposed Method

In this study, we focus on the NLOS reception caused by the double-decker bus, a representative moving object in Hong Kong. Figure 2-1 presents direct propagation routes and potential NLOS reception of the GNSS signal. The double-decker bus (height is 4.5 meters) can block signals transmitted from the satellite. Meanwhile, this GNSS signal can be reflected by nearby buildings and finally received by GNSS receiver equipped on top of the autonomous vehicle resulting in NLOS reception. This magnitude of pseudorange error of this NLOS is subjected to the distance from the GNSS receiver to the reflector and the elevation angle of the satellite [3].

As an essential sensor for positioning and perception of autonomous driving, 3D LiDAR is installed on the top as shown in Figure 2-1. In this study, LiDAR is employed to detect the surrounding double-decker buses. Then, NLOS exclusion is implemented based on detected double-decker boundary parameters which are projected into a

Skyplot, which described the distribution of satellites in terms of elevation and azimuth angles [48]. Finally, GNSS WLS positioning is conducted using the remaining satellites. Figure 2-2 shows the flowchart of the proposed algorithm: improved GNSS positioning by NLOS exclusion based on object detection of LiDAR point cloud. The inputs of the chart include two parts, raw measurements and satellite information from GNSS and 3D point cloud by LiDAR. Moreover, the yaw angle from INS is also an input for coordinate transformation. The output is the GNSS positioning result. The proposed method can be executed as follows:

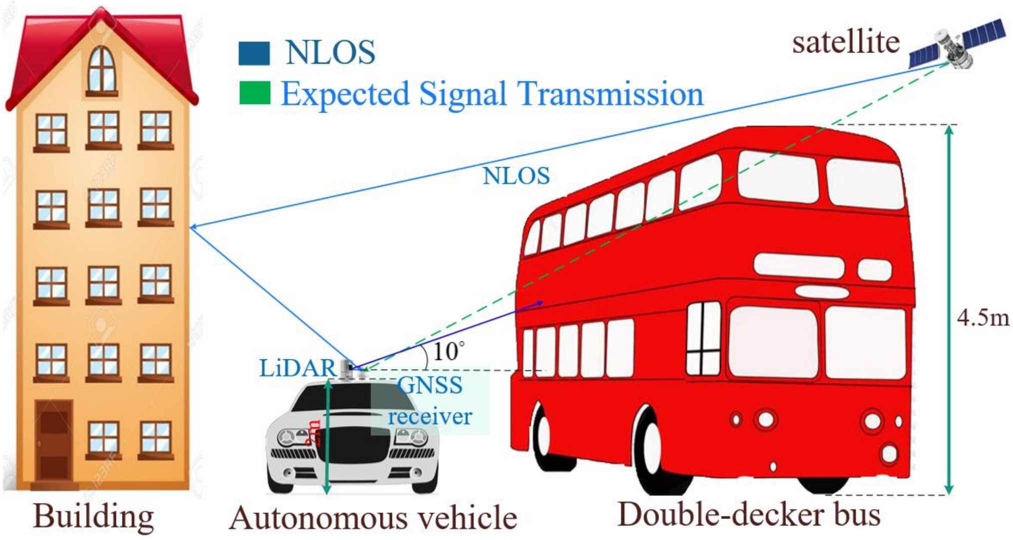

Figure 2-1 Illustration of NLOS receptions caused by a double-decker bus. The double-decker bus causes the GNSS signal blockage. Surrounding objects such as the buildings are the possible reflectors subsequently.

Step I: Euclidean clustering is employed to transfer real-time 3D point clouds into several clusters, so-called point cloud segmentation. The parameters-based classification method is utilized to classify the clusters and identify the double-decker bus from multi-clusters.

Step II: Satellites and the double-decker bus are projected into a Skyplot based on their azimuth and elevation angles relative to the GNSS receiver.

Step III: Considering satellite elevation, azimuth angles, SNR and double-decker bus boundary information (elevation and azimuth angles in Skyplot), satellites that blocked by double-decker bus are excluded.

Step IV: Implementing GNSS WLS positioning using the surviving satellites after the NLOS exclusion by Step III.

The details of the algorithms are introduced in the following sections.

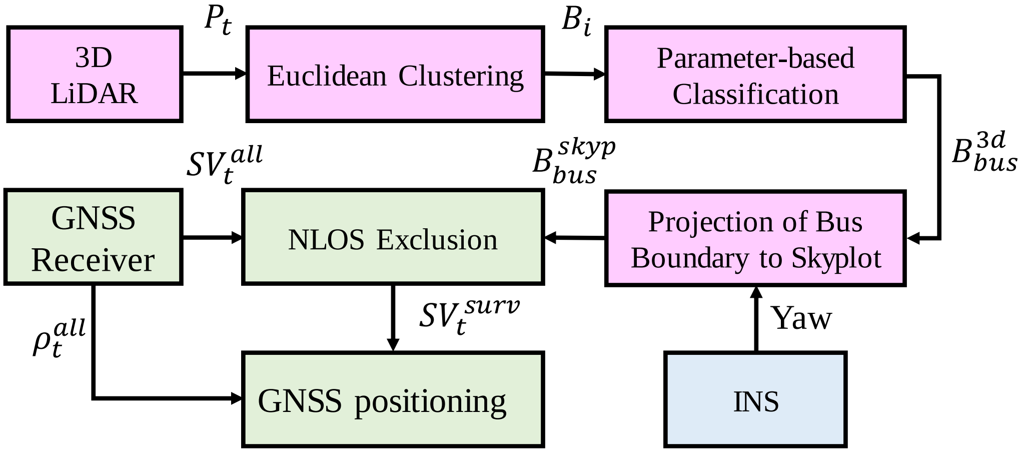

Figure 2-2 Overview of the proposed algorithm of NLOS exclusion and GNSS positioning flowchart. Inputs are the 3D point cloud from 3D LiDAR, yaw angle from INS and raw measurements from GNSS receiver. The output is the GNSS positioning result based on the remaining satellites after NLOS exclusion.

2.3 Double-decker Bus Detection and Transformation

Due to the limited field of view (FOV), for example, +10° to -30° vertical FOV used this paper, the LiDAR can only scan part of the double-decker bus body as shown in Fig. 1. In this section, Euclidean clustering [49] and parameters-based classification methods [46] are employed to detect the double-decker bus. Moreover, the height of the detected double-decker bus is extended to the true height subsequently.

2.3.1 Clustering for Double-decker Bus Detection

From the view of LiDAR, the surrounding environment is represented as numerous points at a given time *t* and the points are considered as a point set $P_t == \{p_1, p_2, \dots, p_i, \dots, p_n, t\}$ where the $p_i = (x_i, y_i, z_i)$ represents a single point at a given time *t* in the LiDAR coordinate system. To give the points set P_t a physical meaning, Euclidean clustering is implemented to divide the point set into several organized sets. The process of the Euclidean clustering algorithm is summarized in Algorithm 2-1.

The output of Algorithm 1 is organized points sets $C_t^{clt} = \{C_1, C_2, \dots, C_i, \dots C_n, t\}$. To better portrait the clusters, each cluster is represented by a descriptor, the bounding box [50]. Based on the principle of the bounding box, each C_i in C_t^{clt} can be transformed to B_i in $B_t^{clt} = \{B_1, B_2, \dots, B_i, \dots B_n, t\}$and is specifically determined by vector B_i as follows:

$$B_i = \left[x_i^c, y_i^c, z_i^c, roll_i^c, pitch_i^c, yaw_i^c, d_i^{len}, d_i^{wid}, d_i^{al}\right] \tag{2–1}$$

where $x_i^c, y_i^c$and z_i^c denote the position of the bounding box in x, y, and z directions respectively in the LiDAR coordinate system. $roll_i^c, pitch_i^c$ and yaw_i^c denote the orientation of the bounding box. d_i^{len} is the length, d_i^{wid} is the width and d_i^{al} is the altitude of the bounding box. The bounding box list B_t^{clt} contains both double-decker buses and other objects.

Algorithm 2-1: Euclidean clustering for points set P_t

```
Input: points set $P_t = \{p_1, p_2, \dots, p_n, t\}$, search radius $r_{search}$
Output: organized points sets $C_t^{clt} = \{C_1, C_2, \dots, C_i, \dots C_n, t\}$
create a Kd-tree representation for the input points set $P_t$
set up an empty clusters list $C_t^{clt}$ and an empty list to save organized point
  sets
for all points $p_i$ in $P_t$ do
  add $p_i$ to the points set $P_t^{check}$
  for all $p_i$ in $P_t^{check}$ do
    search for the points set $N_i$ of point neighbor of $p_i$ in a
    sphere with radius r<$r_{search}$
    for every point $N_i^i$ in points set $N_i$ do
      if $N_i^i$ have not been processed
        add $N_i^i$ to points sets $P_t^{check}$
      end if
    end for the points set $N_i$
  if all the points in $P_t^{check}$ have been processed
    add $P_t^{check}$ to $C_t^{clt}$
    reset $P_t^{check}$ to empty
  end if
  end for $P_t^{check}$
end for $P_t$
```

2.3.2 Parameter-based Double-decker Bus Detection

To determine the double-decker bus clusters in the bounding box list B_t^{clt}, the parameter-based classification method is detailed in Algorithm 2-2. The output of Algorithm 2-2 is the bounding box set indicating the double-decker bus. If one cluster from C_t^{clt} is identified as a double-decker bus, the height for the corresponding bounding box will be extended to the real height d_{bus}^{al}. Dimensions of the bounding box and geometry distribution corresponding to the cluster are all considered for classification. Cubic structure imposes the points in the cluster corresponding to the double-decker bus with specific geometry distribution. Thus, the standard deviation $\sigma_{p,i}^{cluster}$ of all the points is employed as a classification feature which is calculated as follows:

$$\sigma_{p,i}^{cluster} = \sqrt{\frac{\sum_{i=1}^{n}(p_i-\bar{p})^2}{n-1}} \qquad (2–2)$$

$$\bar{p} = \frac{\sum_{i=1}^{n} p_i}{n} \qquad (2–3)$$

where n denotes the total number of points in one cluster. $\bar{p}$ represents the geometry center of the cluster. $(p_i - \bar{p})^2$ means the Euclidean distance between p_i and $\bar{p}$. The inputs of Algorithm 2-2 include bounding Box sets B_t^{clt}, organized point clusters C_t^{clt}, number of points threshold num_{thres}, etc. The function *Num* is used to calculate the number of points in a cluster. len_{min} , and len_{max} are used to identify the bounding box indicating the double-decker bus by the length of the bounding box. The functions of thresholds for length and height are the same as that of the length. As the ranging distance for LiDAR can go up to about 100 meters, thus both width and length can be fully scanned when LiDAR is approaching the double-decker bus.

Algorithm 2-2: Double-decker bus classification

Input: Bounding Box sets $B_t^{clt} = \{B_1, B_2, \dots, B_i, \dots B_n, t\}$, Organized point clusters $C_t^{clt} = \{C_1, C_2, \dots, C_i, \dots C_n, t\}$, point number threshold num_{thres}, true height $d_{bus}^{al}$of the double-decker bus, length threshold len_{min}, len_{max}, width threshold wid_{min}, wid_{max} , height threshold al_{min}, al_{max} and standard deviation threshold PV_{min}, PV_{max}.

Output: Bounding Box set represent double-decker bus $B_t^{clt\_bus} = \{B_1^{bus}, B_2^{bus}, \dots, B_i^{bus}, \dots B_n^{bus}, t\}$

```
set up an empty clusters list $B_t^{clt\_bus}$ to save bounding box
```

```
for all bounding box $B_i$ in $B_t^{clt}$ do
  if $Num(C_i) > num_{thres}$
    $B_i \leftarrow [x_i^c, y_i^c, z_i^c, roll_i^c, pitch_i^c, yaw_i^c, d_i^{len}, d_i^{wid}, d_i^{al}]$
    if $d_i^{len} > len_{min}$ and $d_i^{len} < len_{max}$
      if $d_i^{wid} > wid_{min}$ and $d_i^{wid} < wid_{max}$
        if $d_i^{al} > al_{min}$ and $d_i^{al} < al_{max}$
          $d_i^{al} \leftarrow d_{bus}^{al}$
          $B_i^{bus} \leftarrow B_i$
        end if
      end if
    end if
  end if
end for $B_t^{clt}$
```

2.3.3 Representation of Double-decker Bus Detection

As illustrated previously, only part of the double-decker bus can be scanned by LiDAR which is represented by rectangle ABCD in Figure 2-3. Meanwhile, as the bus surface is partially composed of the glass window, some parts of the bus are vacant from the view of LiDAR which can be seen in Figure 2-3.

Dimension parameters of the bounding box representing the double-decker bus can be extended to the real one. The extended box is represented by rectangle AEFB in Figure 2-3. Then, the boundary parameters for the double-decker bus as shown in Figure 2-2 are denoted by the line segment $\overline{EF}$ denoted as B_{bus}^{3d}, the matrix of the bus boundary. To represent the bus, two points, E and F, are required. The B_{bus}^{3d} is structured as follows:

$$B_{bus}^{3d} = \begin{bmatrix} x_{3dE} & y_{3dE} & z_{3dE} \\ x_{3dF} & y_{3dF} & z_{3dF} \end{bmatrix} \tag{2–4}$$

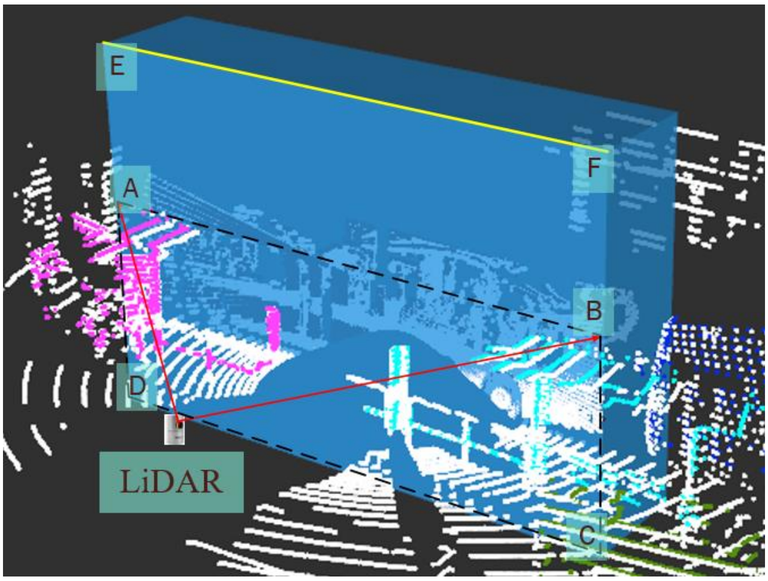

Figure 2-3 Illustration of double-decker bus detection using the Euclidean cluster algorithm and parameters-based classification. Blue box ABCD represents the initially detected double-decker bus. Blue box ABFE represents the extended detected double-decker bus.

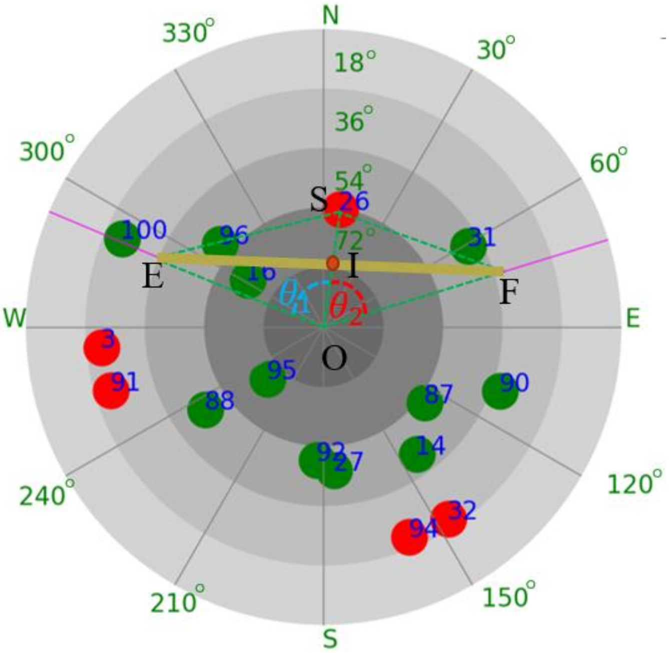

Figure 2-4 Skyplot of GNSS satellites and detected double-decker bus boundary. Green circles and their associated numbers indicate satellites' PRNs. Line segment $\overline{EF}$ indicates the boundary.

Thus, the double-decker bus boundary is detected.

2.3.4 Coordinate Transformation

To implement the algorithm of NLOS exclusion, satellites' visibility needs to be determined based on the boundary of the double-decker bus. Thus, the relative position of the GNSS receiver to satellites and the double-decker bus need to be transformed into the same representation, the Skyplot. In each epoch, satellite information, including azimuth, elevation angles, and SNR, can be acquired from the GNSS receiver. Satellites information can be represented as $SV_t^{all} = \{SV_1, SV_2, \ldots, SV_i, \ldots SV_n\}$. SV_i represents the information for satellite i and $SV_i = \{azim_i, ele_i, SNR_i\}$.

Satellite position can be easily indicated in the Skyplot representation that is 2-dimension coordinate based on corresponding elevation and azimuth angles. Proper transformation matrix should be employed for double-decker bus boundary transformation from 3 dimensions coordinate to 2 dimensions coordinate. The transformation is conducted as the following formula.

$$B_{bus}^{skyp} = B_{bus}^{3d} G_T \tag{2–5}$$

where B_{bus}^{3d} denotes the matrix of bus boundary presented in the previous sub-section. G_T is a 3x2 transform matrix indicating the yaw angle bias between the heading of LiDAR and north of earth, which can be obtained from the INS. The B_{dou}^{skyp} denotes the boundary matrix (2x2) in Skyplot structured as follows:

$$B_{bus}^{skyp} = \begin{bmatrix} x_{skyE} & y_{skyE} \\ x_{skyF} & y_{skyF} \end{bmatrix} \tag{2–6}$$

After the transformation, satellites and the double-decker bus can be presented in the same coordinate system, the Skyplot, as shown in Figure 2-4. Line segment $\overline{EF}$ represents the double-decker bus boundary corresponding to the line segment $\overline{EF}$ as shown in Figure 2-3. Then, the azimuth and the elevation angles for points E, and F can be calculated in the Skyplot respectively.

2.4 Improved GNSS Positioning by NLOS Exclusion

In this section, the NLOS exclusion criterion is proposed based on the detected double-decker bus boundary, satellite elevation, azimuth angles, and SNR. Then, GNSS positioning is conducted by the WLS method.

2.4.1 NLOS Exclusion Based on Double-decker Bus Boundary

To exclude the satellites blocked by the double-decker bus, the relative position between each satellite and the detected bus boundary needs to be calculated. As shown in Figure 2-4, line segment $\overline{EF}$ represents the boundary of a double-decker bus. Satellite 26 (PRN 26) is located at point S. The azimuth and elevation angles are 8° and 54°, respectively. The satellite exclusion procedure is summarized in detail as shown in Algorithm 2-3. Inputs of Algorithm 2-3 are the satellite information SV_t^{all} , bus boundary matrix B_{bus}^{skyp}, the threshold of triangle area $S_{threshold}$, the threshold of SNR $SNR_{threshold}$ and threshold of boundary uncertainty θ_{thres}. Outputs of Algorithm 2-3 are the survived satellites after NLOS exclusion. Firstly, the angle θ_1 and θ_2 shown in Figure 2-4 are estimated. Then areas of the triangle $S_{\Delta SEO}$, $S_{\Delta SFO}$, $S_{\Delta SEF}$ and $S_{\Delta EOF}$ are calculated and ΔS can be estimated subsequently. The ΔS is positive, it means that the satellite is outside the triangle ΔEOF and vice versa. The $(\theta_1 + \theta_2) < 180$ is employed to check if the satellite and the bus are on the same side with respect to the ego-vehicle. If the satellite is outside the triangle ΔEOF, it means that the satellite is possibly blocked by the double-decker bus and vice versa.

Secondly, GNSS measurement that SNR is larger than $SNR_{threshold}$ will not be excluded. To avoid the faulty exclusion, a heuristically determined threshold $S_{threshold}$ is set. Satellites whose positions are quite near the extended edge beam ($\theta_1 < \theta_{thres}$ or $\theta_2 < \theta_{thres}$) also should not be excluded, such as the satellite 100 in Figure 2-4. Satellites whose positions are quite near the double-decker bus boundary should not be excluded which can be judged by ΔS, such as the satellites 31 and 96 in Figure 2-4.

Algorithm 2-3: NLOS exclusion based on bus detection

Input: Satellites information set $SV_t^{all} = \{SV_1, SV_2, \ldots, SV_i, \ldots SV_n\}$, bus boundary matrix B_{bus}^{skyp}, area threshold $S_{threshold}$, SNR threshold $SNR_{threshold}$, the threshold of boundary uncertainty θ_{thres}

Output: surviving satellites set after NLOS exclusion: $SV_t^{surv} = \{SV_1, SV_2, \ldots, SV_i, \ldots SV_m\}$

```
for all satellites $SV_i$ in $SV_t^{all}$do
estimate $\theta_1$, $\theta_2$ as shown in Fig. 4
Get triangle area $S_{\Delta SEO}$ of triangle SEO from $B_{bus}^{skyp}$
Get triangle area $S_{\Delta SFO}$ of triangle SFO from $B_{bus}^{skyp}$
Get triangle area $S_{\Delta SEF}$ of triangle SEF from $B_{bus}^{skyp}$
Get triangle area $S_{\Delta EOF}$ of triangle EOF from $B_{bus}^{skyp}$
$\Delta S = S_{\Delta SEO} + S_{\Delta SFO} + S_{\Delta SEF} - S_{\Delta EOF}$
if $(SNR_i > SNR_{threshold}) or\ (\theta_1 < \theta_{thres})\ or\ (\theta_2 < \theta_{thres})$
    add $SV_i$ to satellites set $SV_t^{surv}$ and break
if $\Delta S > S_{threshold}$ and $((\theta_1 + \theta_2) < 180°)$
    break
else
    add $SV_i$ to satellites set $SV_t^{surv}$
end if
end for satellites set $SV_t^{all}$
```

Finally, all the satellites in SV_t^{all} are indexed and the satellites that should not be excluded will be added to SV_t^{surv}. According to the proposed NLOS exclusion algorithm in Algorithm 2-3, satellites 3, 26, 91,94 and 32 are going to be excluded.

2.4.2 GNSS Positioning Based on Surviving Satellites

Measurements with low elevation angle and SNR are more likely to be a contaminated GNSS signal, such as the multipath or NLOS, due to the reflection, blockage, and diffraction. Thus, proper thresholds need to be set to exclude unhealthy measurements. For satellite SV_i, if ele_i of i_{th} satellite is less than a threshold ele_{thres} , it should be excluded from GNSS positioning.

The clock bias between the GNSS receiver and satellites is usually represented by the pseudorange measurements. The equation linking the receiver position and satellite can be structured as the following formula using least square (LS) method:

$$\hat{x} = (\boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T\rho \tag{2-7}$$

where $\boldsymbol{G}$ represents the observation matrix and is structured by unit LOS vectors between GNSS receivers' position and satellite position. $\hat{x}$ indicates the estimated receiver position and ρ denotes the pseudorange measurements.

To better represent the reliability of each measurement based on the information measured by the receiver, the weightings of each satellite are needed. Function to calculate the weighting by integrating the measurement SNR and satellite elevation is expressed as [51]:

$$W^{(i)}(ele_i, SNR_i) = \frac{1}{sin^2 ele_i}\left(10^{-\frac{(SNR_i - T)}{a}}\left(\left(\frac{A}{10^{-\frac{(F-T)}{a}}} - 1\right)\frac{(SNR_i - T)}{F - T} + 1\right)\right) \quad (2\text{-}8)$$

where $W^{(i)}(ele_i, SNR_i)$ denotes the weighting for satellite SV_i. Parameter a, A and F in (8) are experimentally determined. Then, the weighting matrix W is a diagonal matrix constituted by the weighting $W^{(k)}(ele_i, SNR_i)$. Finally, the GNSS receiver position can be estimated using the WLS method as:

$$\hat{x} = (\boldsymbol{G}^T\boldsymbol{W}\boldsymbol{G})^{-1}\boldsymbol{G}^T\boldsymbol{W}\rho \quad (2\text{-}9)$$

Note that both LS (2-7) and WLS (2-8) positioning methods are compared in the experiment section.

2.5 Experimental Evaluation and Conclusions

2.5.1 Experiment Setup

A static experiment is firstly conducted near a bus stop in Hong Kong with a double-decker bus around to evaluate the magnitude of NLOS errors caused by the double-decker bus. By selecting an obvious landmark in the vicinity of the testing environment, we got the actual position of tested static position by referencing the landmark position labeled in Google Maps. In our experience, the accuracy is within about 1 meter. The dynamic experiment is implemented in urban scenarios of Hong Kong to demonstrate the effectiveness of the proposed method. We use the RTK GNSS/INS integrated system (NovAtel SPAN-CPT, RTK/INS integrated navigation system with fiber optics gyroscopes) to provide high accuracy ground truth. This device is commonly used for ground truth of positioning among academic fields and industry fields. The u-blox M8T receiver is used to collect raw GPS and BeiDou measurements. 3D LiDAR sensor, Velodyne 32, is employed to provide the real-time point cloud.

For the static experiment, both u-blox receiver and 3D LiDAR are installed on a fix position near a static double-decker bus during the experiment. The data were collected at a frequency of 1 Hz. For the dynamic experiment, both u-blox receiver and 3D LiDAR are installed on the top of a vehicle running on an urbanized road.

To verify the effectiveness of the proposed method, four methods were compared, the "EF" represents the "elevation filter". This means that the elevation angle threshold is applied.

(1). LS positioning (LS)

(2). LS positioning + ele_{thres} (LS-EF)

(3). WLS positioning + ele_{thres} (WLS-EF)

(4). WLS positioning + ele_{thres} + NLOS exclusion (WLS-EF-NE)

In this experiment section, the setting of the parameters mentioned above can be found in TABLE 2-1.

TABLE 2-1 PARAMETER VALUES USED IN THIS PAPER

Parameters	$S_{threshold}$	$SNR_{threshold}$	ele_{thres}	θ_{thres}
Value	10	45 dB-Hz	20°	5°
Parameters	a	A	F	
Value	30	32	10	

2.5.2 Comparison of Different GNSS Positioning Methods Using Static Data

The static experimental scenario is shown in Figure 2-5. As the double-decker bus is near the LiDAR sensor, boundary matrix B_{bus}^{skyp} is always available throughout the static test. The experiment results of GNSS positioning using four methods are shown in TABLE 2-2.

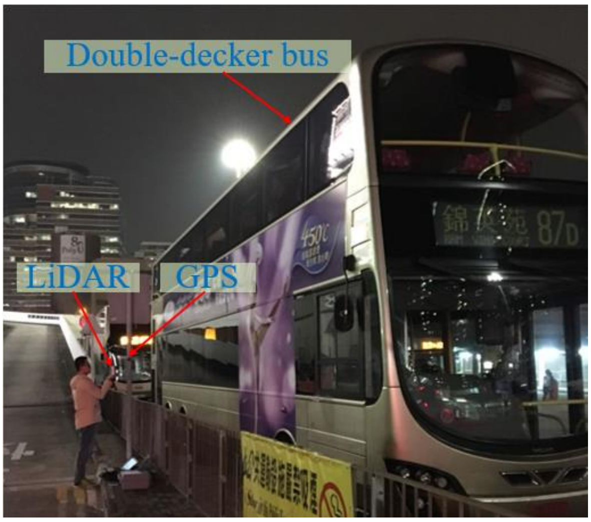

Figure 2-5 An environment that the data were collected at a bus stop. Satellites can be blocked by the double-decker bus.

TABLE 2-2 POSITIONING PERFORMANCE OF THE FOUR METHODS NEAR A BUS STOP (IN THE UNIT OF THE METER)

All data	LS	LS-EF	WLS-EF	WLS-EF-NE
MAE	70.59m	51.91m	47.16m	22.76m
Std	26.0m	29.4m	32.34m	18.59m
RMSE	81.53m	62.34m	57.09m	27.96m
Percentage (MAE<15 meters)	5.81%	11.35%	14.58%	38.00%
Percentage (MAE<30 meters)	9.12%	28.11%	34.46%	77.61%
Percentage (MAE>40 meters)	88.29%	63.24%	50.14%	8.83%
Mean number of excluded satellites		3.13	3.13	5.98

The LS method can achieve only 70.59 meters of mean absolute error (MAE) and 81.53 meters of root mean square error (RMSE) among the test. Approximately 88.29 % of the MAE results have a positioning error of more than 40 meters. With the aid of elevation angle filters, the MAE and RMSE of LS-EF are decreased to 51.91 and 62.34 meters, respectively. About 63.24 % of the results of MAE possess a large error (> 40 meters). Meanwhile, the percentage of MAE results in less than 20 meters is improved from 5.81 % to 11.3 %. This indicates that the elevation filter can enhance the positioning by excluding the unhealthy measurements. The reason behind this improvement is the exclusion of measurements from satellite 3, 91 and 22, which can be seen in Figure 2-6. Those satellites possess low elevation, about 19°, are suffering from severe NLOS and/or multipath effects, thus introducing considerable positioning errors. The mean number of NLOS satellites is 3.13 throughout the test. A slight improvement is obtained using WLS-EF comparing with that of the LS-EF method. The MAE is decreased to 47.16 meters. This enhanced result indicates that weighting shown in (2-8) can effectively represent the health level for each measurement.

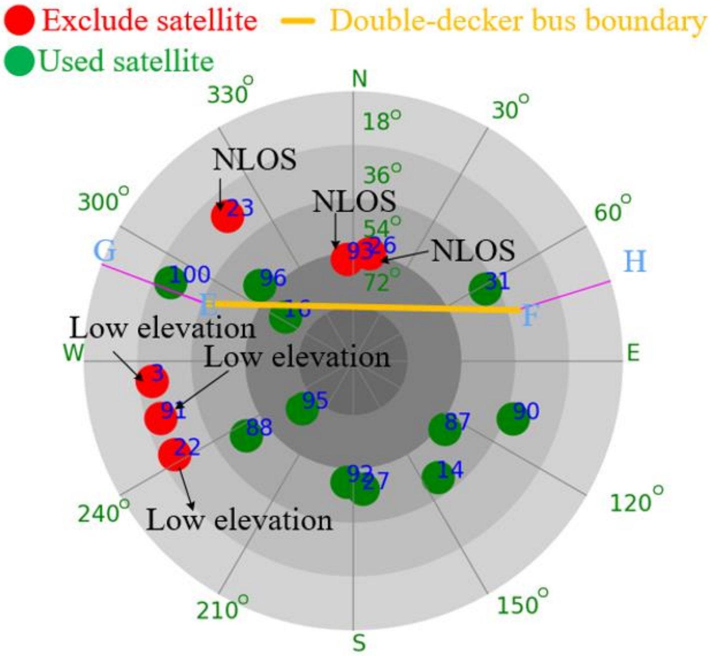

Figure 2-6 Skyplot indicating the satellite distribution during the static experiment. The green circle represents the healthy satellites, which will be used in GNSS positioning. The red circle denotes the excluded satellites. The yellow line indicates the double-decker bus boundary.

With the proposed NLOS exclusion method, the positioning results are considerably improved. Firstly, the positioning error and standard deviation (Std) of WLS-EF-NE are reduced to 22.76 and 18.59 meters, respectively, comparing to that of the WLS-EF method. Secondly, almost 38 % of the results have a small positioning error (<15 meters). Moreover, only 8.83 % of the results possess an error of more than 40 meters. The reason for this improvement is the proposed NLOS exclusion as shown in Figure 2-6. Satellites 23, 26 and 93 are excluded using the proposed algorithm 3. Though the three satellites are blocked by double-decker bus, GNSS signals from them are reflected by surrounding buildings in the double-decker bus station, thus causing the erroneous NLOS receptions. HDOP, positioning error and the numbers of measurement used in the WLS-EF-NE and WLS-EF method are shown in Figure 2-7. The total satellites are over 10 all through the test, thus the availability of GNSS positioning solution is 100 %. After the NLOS exclusion, the HDOP value shown in the second panel is slightly increased, due to the change in the geometry distribution of satellites.

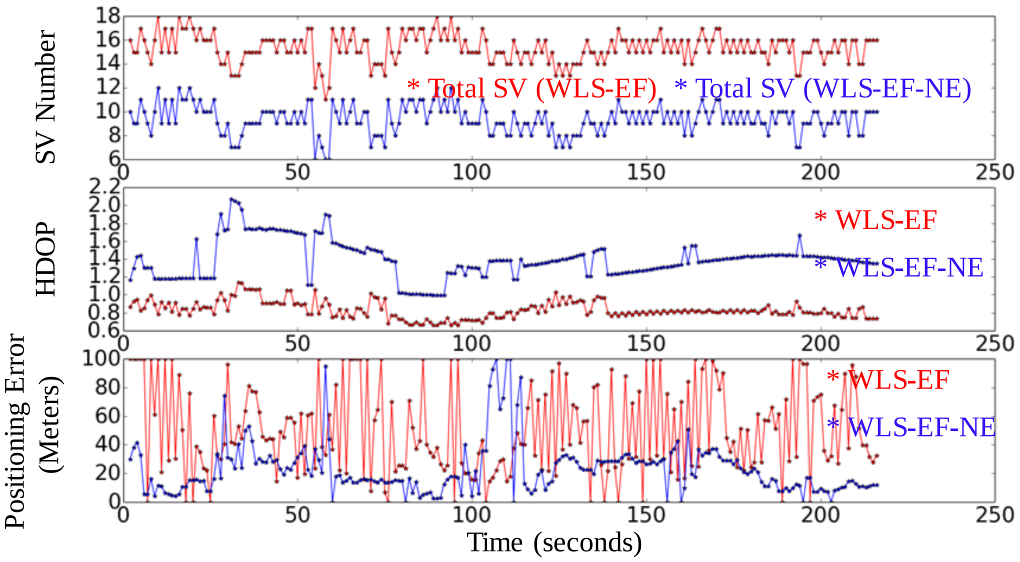

Figure 2-7 Experimental results of WLS-EF and WLS-EF-NE, which depicted in red and blue dots, respectively. The top panel indicates the number of satellites used. Middle panels indicate the HDOP values. Botton panel indicates the 3D positioning errors.

2.5.3 Evaluation of WLS-EF Positioning with Manual Exclusion Using Static data

This sub-section presents the results of WLS-EF with a manual exclusion, meaning a specific measurement is excluded before using the WLS-EF method. TABLE 2-3 shows the results of four separate exclusion tests. Exclusion of satellite 23 introduces a slight improvement in positioning performance with an MAE of 42.5 meters, comparing to the MAE of 47.16 meters using the WLS-EF method without exclusion. As the GNSS signal received from satellite 23 is NLOS. Similarly, exclusion of satellites 26 and 93 also obtain improvements with an MAE of 32.31 meters and 46.51 meters respectively. The reason for this improvement's distinction is that satellite 26 suffered larger NLOS errors comparing to satellites 93 which is subjected to the environment's features. According to [3], the NLOS delay in the pseudorange domain is related to the ground distance from the receiver to the surrounding objects that reflect the signal, and the satellite elevation angle. The satellite with larger ground distance and lower elevation angle is more likely to cause larger NLOS error. Signals from satellites 26 and 93 can be reflected by different objects, respectively. However, the ground distances, denoted by α_1 and α_2, between the receiver and the two separate reflectors can be distinct (α_1 for satellite 26, α_2 for satellite 93). α_1 can be considerably larger than α_2, therefore causing greater positioning error. On the contrary, greater improvement will be introduced if satellite 26 is excluded from GNSS positioning comparing with satellite 93.

After the exclusion of satellite 26, 67.72 % of the results possess an MAE less than 30 meters. However, the exclusion of satellite 100 introduces larger MAE comparing to the no exclusion situation. The MAE increases to 55.08 meters and approximately 59.16 % of the results possess an MAE of more than 40 meters. The reason for this worsen performance is that satellite 100 is not blocked by double-decker bus though it is quite near the extended edge beam (line segment $\overline{EG}$ in Figure 2-6). Thus, excluding satellites 23, 26 and 93 can all obtain improvements in GNSS positioning due to the double-decker bus blockage and subsequent NLOS receptions.

TABLE 2-3 POSITIONING PERFORMANCE OF WLS-EF WITH SINGLE SATELLITE EXCLUSION (IN THE UNIT OF THE METER)

All data	PRN23	PRN26	PRN93	PRN100
MAE	42.5m	32.31m	46.51m	55.08m
Std	27.53m	26.67m	30.01m	30.28m
RMSE	47.65m	39.50m	56.94m	67.75m
Percentage (MAE<15 meters)	15.33%	18.51%	6.38%	4.12%
Percentage (MAE<30 meters)	37.71%	67.72%	38.29%	25.13%
Percentage (MAE>40 meters)	45.90%	23.28%	45.74%	59.16%
Improvement (Mean error)	4.66m (9.8%)	14.8m (31.38%)	0.65m (1.42%)	-7.92m (worsen: -16.79%)

2.5.4 Comparison of Different GNSS Positioning Methods Using Dynamic Data

The dynamic experimental scenario is shown in Figure 2-8 and Figure 2-11, with a GNSS receiver and LiDAR sensor installed on a vehicle driven in urban areas.

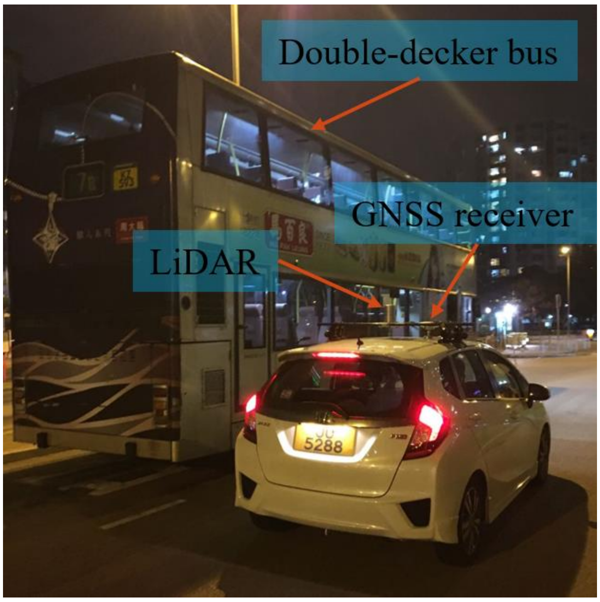

Figure 2-8 The environment that the data were collected in an urban scenario in Hong Kong. Satellites can frequently be blocked by the double-decker.

As the testing vehicle is driven on the road, the boundary matrix B_{bus}^{skyp} is frequently available throughout the dynamic test. Similar to Figure 2-6, the Skyplot representing one epoch in the test can be seen in Figure 2-9. Satellite 6, 30 and 88, with elevations of 41°, 23°, and 46° respectively, are excluded, due to the blockage from the double-decker bus.

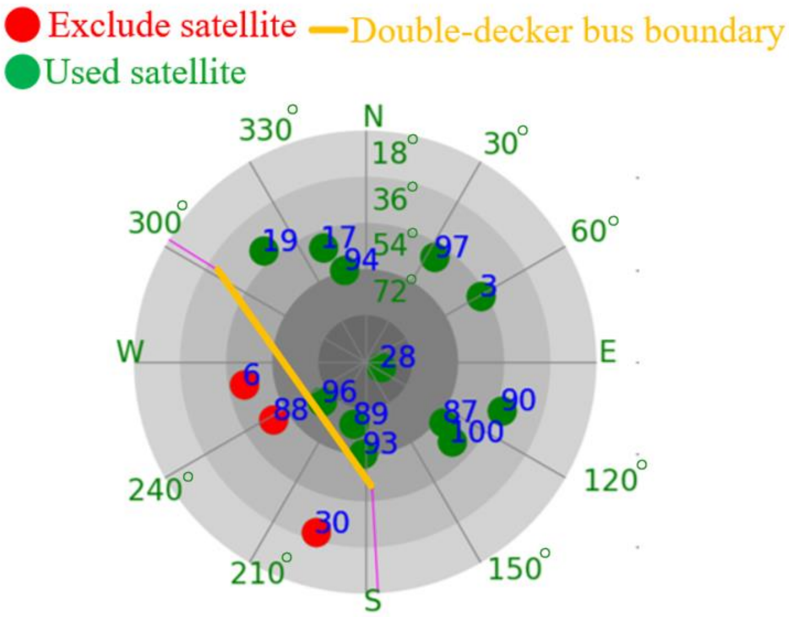

Figure 2-9 Skyplot snapshot indicating the satellite distribution during the dynamic experiment. The green circle represents the satellites that are healthy, which will be used in GNSS positioning. The red circle denotes the excluded satellites. The yellow line indicates the double-decker bus boundary.

There are still about 12 satellites including GPS and BeiDou remained which are enough for further GNSS positioning. The experiment results of GNSS positioning using WLS-EF and WLS-EF-NE are shown in Figure 2-10 and the only discrepancy of the two methods is NLOS exclusion proposed in this paper.

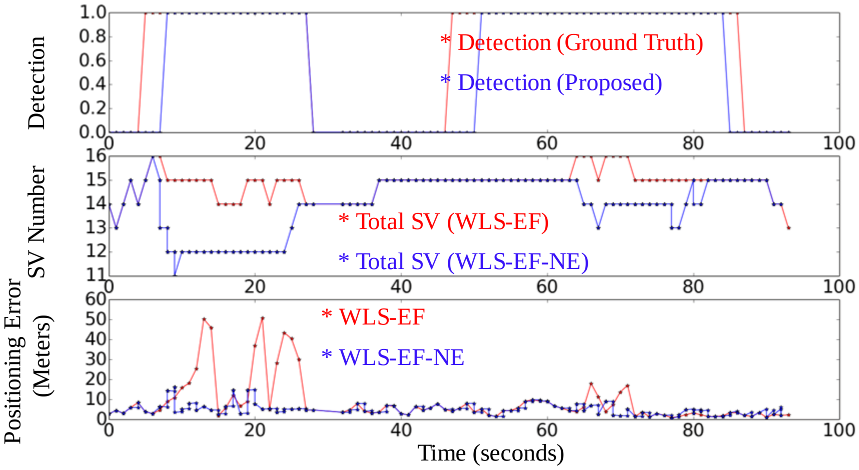

Figure 2-10 Experimental results of WLS-EF and WLS-EF-NE in the dynamic experiment, which depicted in red and blue dots, respectively. The top panel indicates the double-decker bus detection result. The middle panel indicates the numbers of satellites used. The bottom panel indicates the 3D positioning errors.

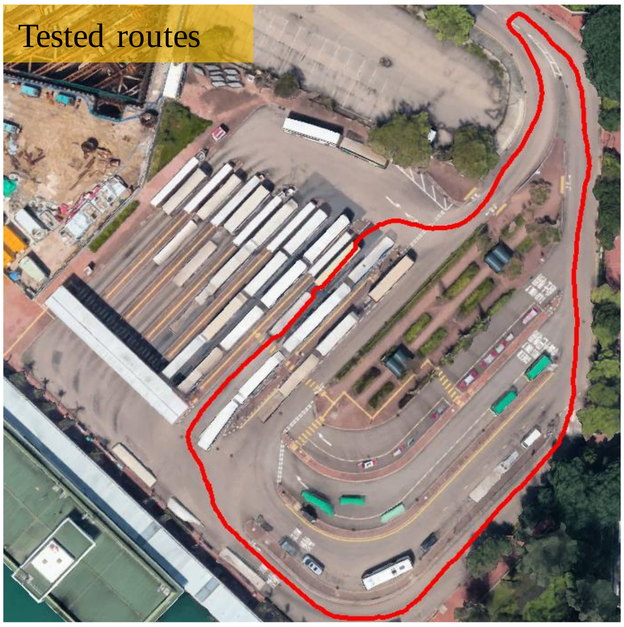

Figure 2-11 Illustration of tested routes in the dynamic experiment (plotted in Google Earth). The red curve indicates the ground truth of tested routes provided by NovAtel SPAN-CPT.

The top panel denotes the double-decker bus detection performance through the test. Detection accuracy of 90.43 % is obtained with the proposed double-decker bus detection method. The failure in double-decker-bus detection usually occurs when the

LiDAR is approaching or leaving the double-decker bus. This is because of the limited FOV of LiDAR and only a few parts of the bus can be scanned which only provides few points. The middle panel indicates the satellite numbers using WLS-EF and WLS-EF-NE throughout the test. Approximately 2 to 4 satellites are excluded during the test when there is a double-decker bus. We can see from Figure 2-10 that there two times that the double-decker bus causes NLOS. The bottom panel indicates the GNSS positioning accuracy through the dynamic test. The positioning errors can be effectively reduced with the NLOS exclusion of those blocked satellites. The positioning errors are all reduced to less than 20 meters with the proposed method, compared with the conventional method (WLS-EF) whose positioning errors can go up to even 50 meters in Figure 2-10. TABLE 2-4 shows the GNSS positioning performance using four positioning methods.

TABLE 2-4 Positioning Performance of the four methods in the dynamic test (in the unit of the meter)

All data	**LS**	**LS-EF**	**WLS-EF**	**WLS-EF-NE**
MAE	47.59m	44.98m	13.19m	5.04m
Std	42.07m	40.99m	14.67m	2.87m
RMSE	58.61m	52.17m	16.58m	6.29m
Percentage (MAE<5 meters)	29.71%	30.23%	46.51%	53.49%
Percentage (MAE<10 meters)	37.16%	41.86%	60.47%	95.35%
Percentage (MAE>15 meters)	60.35%	55.81%	30.23%	0%

The LS method can achieve only 47.59 meters of MAE and 58.61 meters of RMSE among the dynamic test. Approximately 29.71 % of the result possess an error smaller than 5 meters and 60.35 % of the results have a positioning error of more than 15 meters. With the aid of elevation filters, the MAE and RMSE of LS-EF are decreased to 44.98 and 52.17 meters, respectively. About 55.81 % of the results are larger than 15 meters, which indicates the improvements introduced by the elevation angle filters. Considerable improvements are provided by the WLS-EF method with the $w^{(i)}(ele_i, SNR_i)$. The MAE is decreased to 13.19 meters and about 46.51 % of the results have an MAE of less than 5 meters. Moreover, only 30.23 % of the results possess large errors (> 15 meters) and the standard deviation drops dramatically comparing with the LS-EF method.

With the proposed method, decent improvements are obtained. Firstly, the positioning error declines from 13.19 meters to 5.04 meters in the tested dynamic experiment, which obtains about 61.79 % of the improvements, compared with the WLS-EF method. The standard deviation drops from 14.67 to 2.87 meters. Interestingly, 53.49 % of the results have an MAE of less than 5 meters and approximately 95.35 % of the results possess errors less than 10 meters. Both GNSS positioning accuracy and standard deviation are improved by using the proposed method.

2.5.5 Conclusions

With the rise of GNSS, more satellites are available including GPS, BeiDou, GLONASS and Galileo. The number of visible satellites is still enough for GNSS positioning even after NLOS exclusion. This study firstly employs an object detection algorithm to detect a double-decker bus and extend its dimensions to a real one. Then, the proper coordinate transformation is utilized to project the double-decker bus boundary into GNSS Skyplot. NLOS exclusion criterion integrating elevation angle, SNR and bus boundary is proposed. According to the experiment result, the proposed method obtains the best performance among the four conventional GNSS positioning methods. Positioning error of NLOS receptions caused by a double-decker bus can reach 24 meters in the static experiment and 8 meters improvement in the evaluated dynamic experiment. The magnitude of the improvement is subjected to environmental features. Finally, we conclude that the exclusion of NLOS receptions is necessary for obtaining better GNSS positioning accuracy in urbanized cities.

3. GNSS NLOS CORRECTION AIDED BY 3D LIDAR AND BUILDING HEIGHT

3.1 Introduction

Positioning in urban environments is becoming essential due to the increasing need for autonomous vehicles. To achieve L4 [52] autonomous driving in all scenarios, centimeter-level absolute positioning is required. The 3-dimensional (3D) light detection and ranging (LiDAR) is widely used in autonomous driving vehicles [41, 42]. LiDAR, camera, and inertial navigation system (INS) only provide relative positioning. Thus, these sensors are usually integrated with the global navigation satellite system (GNSS) [22, 53-55], because the GNSS is one of the indispensable sources that can provide absolute positioning. A GNSS/INS/LiDAR/high definition (HD) map integrated system can provide satisfactory localization service in sub-urban [56, 57] areas. The suburban areas can be regarded as GNSS-friendly area since the GNSS receiver can receive sufficient direct signals transmitted from multi-constellation GNSS [24]. In urbanized cities, such as Tokyo, Hong Kong, and New York, the signals from satellites can be reflected, blocked, and diffracted by surrounding buildings before they are received by the receiver. If the direct light-of-sight (LOS) is blocked and reflected signals from the same satellite are received, the notorious non-light-of-sight (NLOS) receptions occur. This NLOS is the dominant GNSS positioning error in the cities mentioned above [7]. As a result, the positioning error can go up to even 100 meters [3, 6].

According to a recent review paper [7], NLOS is currently the major difficulty in the use of GNSS in the applications of the intelligent transportation system. Due to NLOS, the performance of GNSS positioning becomes highly related to environmental features, such as buildings. Utilizing the 3D building model to detect the NLOS is straightforward. NLOS can be detected with the aid of a building model and then be excluded from GNSS positioning [8, 9]. However, the NLOS exclusion will distort the geometric distribution of the satellites. In urban canyons, the distortion results in large positioning errors in the across-street direction because only the measurements from the satellites located at the along-street direction are not excluded. Moreover, identifying NLOS measurements based on the 3D building model relies heavily on the initial guess of the GNSS receiver [33].

Figure 3-1 shows the numbers of GNSS satellites (GPS and BeiDou) received by a commercial GNSS receiver in an urban canyon in Hong Kong. We can see from Figure 3-1 that the number of satellites is dramatically decreased after applying NLOS exclusion. The horizontal dilution of precision (HDOP) is increased (shown in Figure 3-1) distinctly. Thus, it is not preferable to exclude all the NLOS measurements in such an area (i.e., a narrow urban canyon). A smart approach, GNSS shadow matching, is proposed to match the measured satellite visibility (classifying into LOS and NLOS) with the predicted satellite visibility of hypothesized positions [58]. This method makes use of the NLOS to improve the positioning accuracy in the cross-street direction [59]. A likelihood-based 3DMA GNSS method, which models the measurement uncertainty to mitigate the NLOS receptions, is also proposed to provide accurate positioning in the along-street direction [14]. Due to the complementariness of the shadow matching and the likelihood-based 3DMA GNSS, their integration has been studied recently [15]. Another stream of range-based 3DMA GNSS methods is to correct the NLOS affected measurement for GNSS positioning [16-19]. These methods are proposed to simulate the signals' transmission routes using the ray-tracing method [26]. However, the drawbacks of these ray-tracing-based 3DMA GNSS methods are the stringent requirements on 1) the accuracy of the 3D mapping database, 2) the initial guess of receiver positions, and 3) the computational power of the processors.

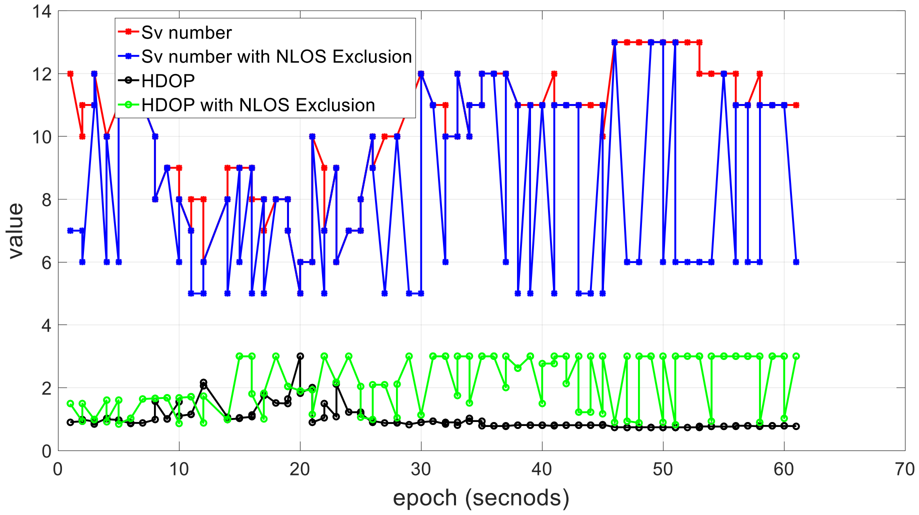

Figure 3-1 Demonstration of numbers of satellite (GPS/BeiDou) measurement before (red) and after (blue) NLOS exclusion in an urban canyon in Hong Kong. The horizontal dilution of precision (HDOP) is also given before (black) and after (green) exclusion.

Instead of using only the 3D mapping data, other scanning sensors, including cameras and LiDAR, can also be employed to sense the surrounding environments of the receiver in real-time operation. To detect the visibility of satellites, omnidirectional and fisheye cameras [38, 39, 60] are used to detect the skylines of buildings in the urbanized area. NLOS receptions can be detected with the detected skylines, and some improvements are obtained. However, this method can suffer from strong light or night scenarios, as computer vision is employed to detect the skylines. The constructed map of the environment using 3D LiDAR is employed to classify the visibility of satellites. A study then modeled the GNSS noise covariance by NLOS detection based on a LiDAR-constructed map [36]. Research incorporating the LiDAR map and 3D city model to exclude NLOS is conducted in the application of unmanned aerial vehicles [61]. However, these methods still tend to exclude the NLOS receptions from further GNSS positioning, which is not applicable in the deep urban areas.

In this section, we propose to improve the GNSS single point positioning (SPP) by detecting and correcting the NLOS receptions based on the environment features perceived by real-time 3D point clouds generated by 3D LiDAR. The perceived environment features refer to the surrounding buildings of the receiver in this paper. Dimension and pose (including the position and orientation) of the building wall relative to the GNSS receiver are calculated by the point cloud-based segmentation. Due to the limited field of view (FOV), tall buildings cannot be fully scanned. Thus, the height of the detected building wall is extended to the exact height provided by a building height list obtained from Google Earth. Then, the satellites and detected TEBs are projected into the Skyplot. To implement the projection, the globally referenced yaw angle of the vehicle is needed and is provided by an RTK GNSS/INS integrated system (SPAN-CPT). Based on the detected TEBs, NLOS measurement can be identified. Previously, our team proposed an NLOS error model based on two parameters: the distance between the GNSS receiver and NLOS reflector, and the elevation and azimuth angle of the satellite [3]. In this method, innovatively, the distance between the GNSS receiver and reflectors can be obtained by the LiDAR scanner. Thus, the correction of NLOS-affected pseudorange measurements can be calculated. Finally, GNSS SPP is calculated using both the corrected NLOS measurements and LOS visible measurements.

To the best of the authors' knowledge, this is the first attempt to aid GNSS SPP by employing the real-time 3D point clouds to detect and correct the NLOS measurements. This is important because GNSS is usually integrated with dead-reckoning (e.g., INS, odometer, visual odometry, LiDAR odometry, etc.) for various applications. Tightly-coupled integration is one of the most popular existing solutions for the integrated navigation system. Our proposed GNSS SPP with NLOS correction can easily fit into the tightly-coupled integration scheme. In other words, it can be easily implemented in many existing navigation systems, especially those used in autonomous driving.

The remainder of this paper is structured as follows. An overview of the proposed method is given in Section 3-2. Section 3-3 discusses a method to detect TEBs from LiDAR point clouds. The coordinate transformation from LiDAR to GNSS Skyplot coordinate system is also presented in this section. In Section 3-4, the criterion of NLOS detection is proposed and the NLOS correction model is introduced. In Section 3-5, we evaluate the effectiveness of the proposed method by two-vehicle experiments in two typical urban canyons in Hong Kong. Finally, conclusions are drawn.

3.2 Overview of the Proposed Method

In this study, we focus on the NLOS receptions caused by surrounding buildings. Figure 3-2 presents direct propagation routes, multipath and potential NLOS receptions of GNSS signals. The buildings, of which height is indicated by H, can block a signal transmitted from a satellite, for example, satellite 1 in Figure 3-2. Meanwhile, this GNSS signal is reflected by the other nearby building and finally received by the GNSS receiver equipped on top of the autonomous vehicle, which results in NLOS receptions. This kind of scenario is a regular case in Hong Kong. In this case, the number of satellites visible to the GNSS receiver is related to the height of buildings and the distance from the receiver to the building (α_i in Figure 3-2).

As a significant sensor for positioning and perception of autonomous driving [62], 3D LiDAR is installed on the top, as shown in Figure 3-2. In this paper, LiDAR is employed to detect the surrounding building surfaces and obtain the distance from the GNSS receiver to the building surface, and then the TEBs can be identified consequently. Then, NLOS detection and correction is implemented based on detected TEBs, which are projected into a Skyplot, and the distance from GNSS receiver to buildings. Finally, GNSS positioning is performed using both the corrected and healthy pseudorange measurements. Figure 3-3 shows the flowchart of the proposed method. The proposed method can be executed as follows:

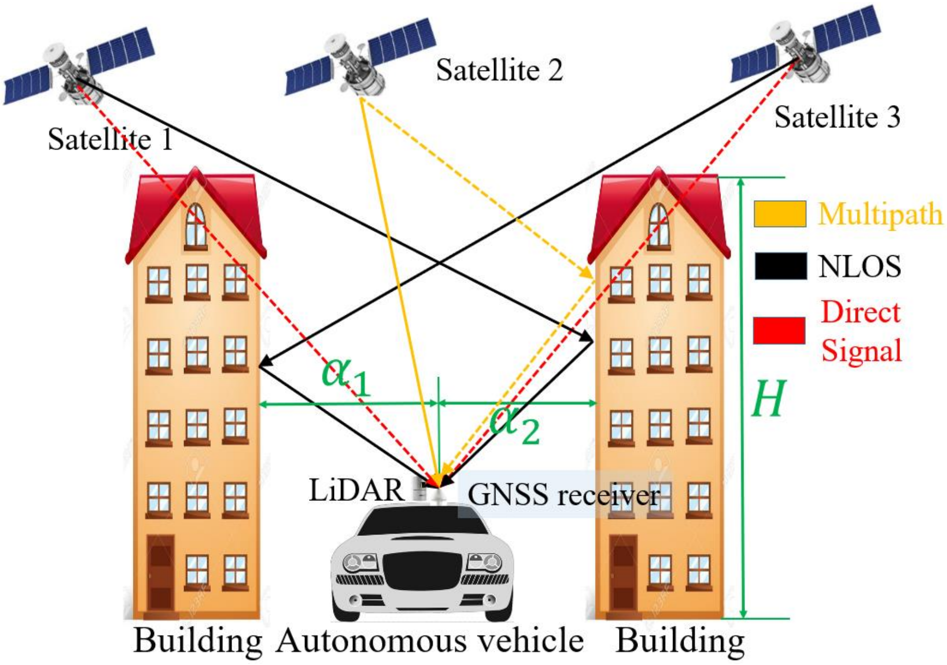

Figure 3-2 Illustration of GNSS signal transmission routes in the urbanized area in Hong Kong. NLOS/multipath can be caused by surrounding buildings.

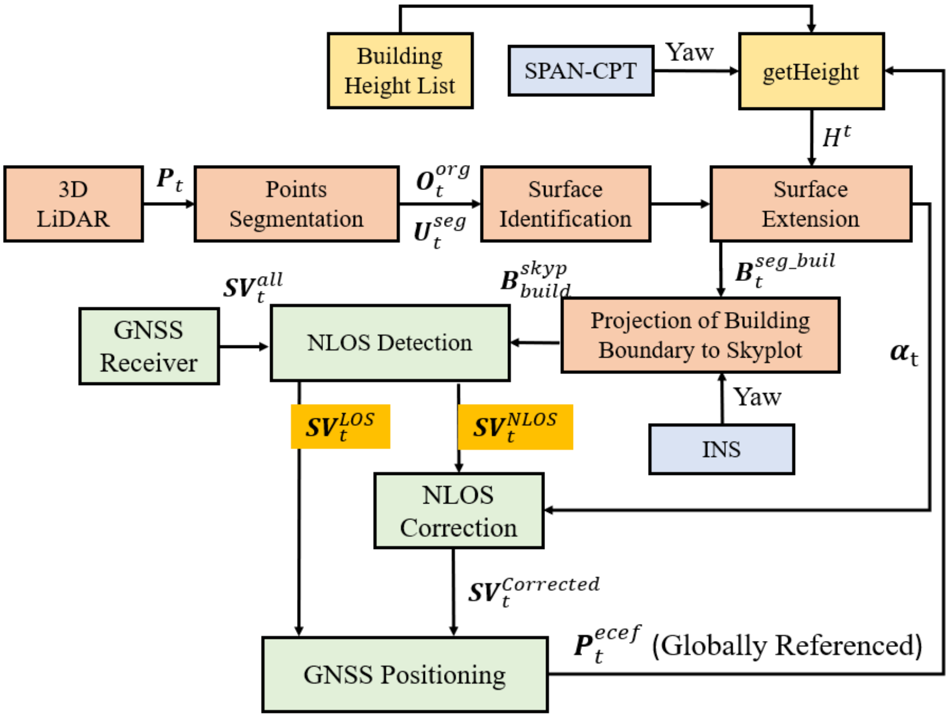

Figure 3-3 Flowchart of the proposed method of GNSS SPP with NLOS correction. The inputs are the 3D LiDAR, yaw angle, building height list, and GNSS raw measurements.

Step I: The Point cloud segmentation method is employed to detect the building surface. The pose of TEBs relative to the GNSS receiver is calculated. The distance between the GNSS receiver and the buildings can be obtained subsequently. Moreover, building height list from Google Earth is employed to extend the detected building height to the exact height.

Step II: The TEBs are projected into a GNSS Skyplot based on their estimated poses relative to the GNSS receiver, and yaw angle provided by the SPAN-CPT (RTK GNSS/INS integrated navigation system).

Step III: Considering the satellite's elevation angle, azimuth angle, SNR, and TEB information (elevation and azimuth angles in Skyplot), satellites blocked by buildings are detected. Consequently, NLOS correction is implemented with an NLOS error model.

Step IV: Implementing GNSS weighted least squares (WLS) based on the corrected pseudorange measurements and healthy pseudorange measurements.
The details of the algorithms are introduced in the following sections.

3.3 Building Surface Detection and Transformation

To detect the surface of the building and obtain the corresponding distances between the GNSS receiver and buildings, a point cloud segmentation method is employed to implement the building surface detection in this section.

3.3.1 TEBs Detection and Transformation

The surrounding environment is expressed as points set $\boldsymbol{P}_t = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_n, t\}$ at a given time, t, where $\boldsymbol{p}_i = (x_i, y_i, z_i)$ represents a single point in the LiDAR coordinate system. To distinguish the building surface from the unordered points set and determine the distance from GNSS receiver to the building surface, three steps are needed: 1) Segmentation of point clouds, 2) Identification of buildings from segmented objects, and 3) Extension of top edges of buildings (TEBs).

The point cloud segmentation is summarized in detail in Algorithm 3-1. Inputs of Algorithm 3-1 are: points set (3D point clouds) and search radius r_{search}, which is the variable constraining the searching area in the *KD*-tree [63]. Outputs include the bounding box [50] sets ($\boldsymbol{U}_{\mathbf{t}}^{\mathbf{seg}}$) and organized point clusters ($\boldsymbol{O}_{\mathbf{t}}^{\mathbf{org}}$), which indicate different objects, such as buildings and vehicles. The definitions of applied variables and functions in Algorithm 3-1 are listed as follows:

- $\boldsymbol{P}_{\mathbf{t}}$**:** input 3D point clouds. $\boldsymbol{U}_{\mathbf{t}}^{\mathbf{seg}}$: segmented bounding box sets. $\boldsymbol{O}_{\mathbf{t}}^{\mathbf{org}}$: segmented point clusters.
- $\boldsymbol{P}_{\mathbf{t}}^{\mathbf{check}}$**:** a middle variable that contains checked points. $\boldsymbol{N}_{i}$: a neighboring points set given a searching radius. r_{search}: the radius of the neighboring points searching area.

The *BoundingBox* mentioned in Algorithm 1 is a function to get the bounding box that represents the organized point cluster. Bounding box $\boldsymbol{U}_{\boldsymbol{i}}$ is specifically determined by a vector, $\boldsymbol{U}_{i}$, as follows:

$$\boldsymbol{U}_{\mathbf{i}} = \left[x_i^c, y_i^c, z_i^c, roll_i^c, pitch_i^c, yaw_i^c, d_i^{len}, d_i^{wid}, d_i^{hei}\right]^T \tag{3-1}$$

where x_i^c, y_i^c, and z_i^c denote the position of the bounding box in *x*, *y*, and *z* directions in the LiDAR coordinate system, respectively. $roll_i^c$,$pitch_i^c$, and yaw_i^c denote the orientation of the bounding box in the LiDAR coordinate system. d_i^{len} is the length, d_i^{wid} is the width, and d_i^{hei} is the height of the bounding box.

The principle of Algorithm 3-1 is also shown on the left side of Figure 3-4. The colored points represent raw 3D point clouds. After applying the Algorithm 3-1, two clusters are detected which are annotated by the two black 2D bounding box ($\boldsymbol{U}_{\mathbf{t}}^{\mathbf{seg}}$). However, we do not know which belongs to the building class.

Algorithm 3-1: Segmentation for points set $\boldsymbol{P}_t$

```
Input: points set $\boldsymbol{P}_{\mathbf{t}} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_n, t\}$, search radius $r_{search}$
Output: Bounding box sets $\boldsymbol{U}_{\mathbf{t}}^{\mathbf{seg}} = \{\boldsymbol{U}_1, \boldsymbol{U}_2, \dots, U_i, \dots \boldsymbol{U}_m, t\}$, Organized point clusters $\boldsymbol{O}_{\mathbf{t}}^{\mathbf{org}} = \{\boldsymbol{O}_1, \boldsymbol{O}_2, \dots, \boldsymbol{O}_i, \dots \boldsymbol{O}_m, t\}$
create a KD-tree representation for the input points set $\boldsymbol{P}_t$
setup an empty list to save point sets $\boldsymbol{P}_{\mathbf{t}}^{\mathbf{check}}$
for all points $p_i$ in $\boldsymbol{P}_{\mathbf{t}}$ do
  add $p_i$ to the points set $\boldsymbol{P}_t^{check}$
  for all $p_i$ in $\boldsymbol{P}_{\mathbf{t}}^{\mathbf{check}}$ do
    search for the points set $\boldsymbol{N}_i$ of point neighbor of $\boldsymbol{p}_i$ in a
    sphere with radius $r<r_{search}$
    for every point $\boldsymbol{N}_i^i$ in points set $\boldsymbol{N}_{\boldsymbol{i}}$ do
      if $\boldsymbol{N}_i^i$ have not been processed
```

```
                add $\boldsymbol{N}_i^i$ to points sets $\boldsymbol{P}_t^{check}$
            end if
        end for the points set $\boldsymbol{N}_i$
    if all the points in $\boldsymbol{P}_{\mathbf{t}}^{\mathbf{check}}$ have been processed
        add $\boldsymbol{P}_{\mathbf{t}}^{\mathbf{check}}$ to $\boldsymbol{O}_{\mathbf{t}}^{\mathbf{org}}$ as an organized point set
        add BoundingBox($\boldsymbol{P}_{\mathbf{t}}^{\mathbf{check}}$) to $\boldsymbol{U}_{\mathbf{t}}^{\mathbf{seg}}$ as a bounding box
        reset $\boldsymbol{P}_{\mathbf{t}}^{\mathbf{check}}$ to empty
    end if
    end for $\boldsymbol{P}_{\mathbf{t}}^{\mathbf{check}}$
end for $\boldsymbol{P}_{\mathbf{t}}$
```

To effectively identify the bounding box ($\boldsymbol{U}_{\mathbf{t}}^{\mathbf{seg}}$) representing the building surface which can result in GNSS signal reflections and subsequent NLOS receptions, a surface identification method is needed; this is summarized in detail in Algorithm 3−2. The objective of Algorithm 3−2 is to: 1) identify the buildings, shown in the middle side of Figure 3-4 and 2) extend its heights to the exact one (right side of Figure 3-4). The inputs of this algorithm are $\boldsymbol{U}_t^{seg}$ and $\boldsymbol{O}_t^{org}$ obtained from Algorithm 3-1, and some experimentally determined thresholds.

The definitions of applied variables and functions in Algorithm 3-2 are listed as follows:

- num_{thres}**:** The number of points that the cluster belongs to the building class should contain.
- len_{thres}: minimum length of a 2D bounding box that belongs to the building class.
- hei_{thres}: minimum height of a 2D bounding box that belongs to the building class.
- building height list, H_{build}, receiver position $\boldsymbol{P}_r^{ecef}$, and yaw angle, Yaw_r.

The output is the bounding box set $\boldsymbol{B}_t^{seg\_buil}$ specifically representing the building surface. The function *Num* mentioned in Algorithm 3−2 is used to calculate the number of points in each cluster, $\boldsymbol{O}_i$. The function $getHeight$ is used to search the height information from a saved building height list, which contains the height information. To determine the actual height of the identified building surface, $\boldsymbol{P}_r^{ecef}, \boldsymbol{U}_i$, and Yaw_r are also needed. $\boldsymbol{P}_r^{ecef}$ indicates the GNSS position given by the previous-epoch positioning result. The relative position between the GNSS receiver and detected building can be obtained from $\boldsymbol{U}_i$. For each bounding box, $\boldsymbol{B}_i$, the distance, α_i, from the receiver to the detected building surface can be calculated as follows:

$$\alpha_i = \sqrt{((x_i^c)^2 + (y_i^c)^2 + (z_i^c)^2)} \tag{3-2}$$

Algorithm 3-2: Building surface identification from bounding box sets and height extension of TEBs

Input: Bounding Box sets $\boldsymbol{U}_{\mathbf{t}}^{\mathbf{seg}} = \{\boldsymbol{U}_1, \boldsymbol{U}_2, \dots, U_i, \dots \boldsymbol{U}_m, t\}$, Organized point clusters $\boldsymbol{O}_{\mathbf{t}}^{\mathbf{org}} = \{\boldsymbol{O}_1, \boldsymbol{O}_2, \dots, \boldsymbol{O}_i, \dots \boldsymbol{O}_m, t\}$, point number threshold num_{thres}, length threshold len_{thres} and height threshold hei_{thres}, building height list $\boldsymbol{H}_{\boldsymbol{build}}$, receiver position $\boldsymbol{P}_{\mathbf{r}}^{\mathbf{ecef}}$, yaw angle Yaw_r

Output: Bounding Box set represents building surfaces $\boldsymbol{B}_{\mathbf{t}}^{\mathbf{seg\_buil}} = \{\boldsymbol{B}_1, \boldsymbol{B}_2, \dots, \boldsymbol{B}_i, \dots \boldsymbol{B}_m, t\}$

```
setup an empty clusters list $\boldsymbol{B}_{\mathbf{t}}^{\mathbf{seg\_buil}}$ to save bounding box
for all bounding box $\boldsymbol{U}_i$ in $\boldsymbol{U}_{\mathbf{t}}^{\mathbf{seg}}$ do
  if $Num(\boldsymbol{O}_i) > num_{thres}$
    $\boldsymbol{U}_{\mathbf{i}} \leftarrow [x_i^c, y_i^c, z_i^c, roll_i^c, pitch_i^c, yaw_i^c, d_i^{len}, d_i^{wid}, d_i^{hei}]$
    if $d_i^{len} > len_{thres}$ and $d_i^{hei} > hei_{thres}$
      $d_i^{hei} \leftarrow getHeight(\boldsymbol{H}_{\mathbf{build}}, \boldsymbol{P}_{\mathbf{r}}^{\mathbf{ecef}}, \boldsymbol{U}_{\mathbf{i}}, Yaw_r)$
      $\boldsymbol{B}_i \leftarrow \boldsymbol{U}_{\mathbf{i}}$
    end if
  end if
end for $\boldsymbol{U}_{\mathbf{t}}^{\mathbf{seg}}$
```

Thus, the bounding box with extended height representing the building surface can be identified with Algorithm 3-2. The height of the bounding box representing the building surface can be extended to the real one. The bounding box is extended from rectangle ABCD to rectangle CDEF, as can be seen on the right side of Figure 3-4. Then, the parameters of TEBs for the bounding box, B_i, corresponding to building surface is denoted by the line segment $\overline{EF}$, denoted as $\boldsymbol{B}_{build}^{3d}$, the matrix of the building boundary [58]. To represent the building boundary, two points, E and F, are required. The B_{build}^{3d} is structured as follows:

$$\boldsymbol{B}_{\mathbf{build}}^{\mathbf{3d}} = \begin{bmatrix} x_{3dE} & y_{3dE} & z_{3dE} \\ x_{3dF} & y_{3dF} & z_{3dF} \end{bmatrix} \tag{3–3}$$

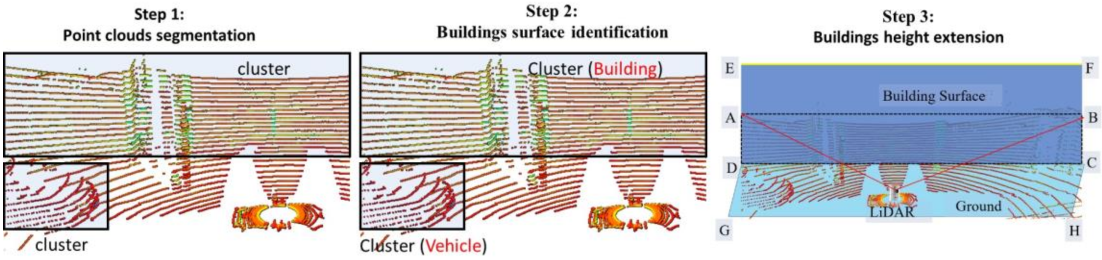

Figure 3-4 Illustration of point sets segmentation and TEBs identification, extension. Box ABCD represents the initially detected building surface. Box CDEF represents the extended building surface. Box CDGH represents the ground. The color points denote the point clouds from 3D LiDAR.

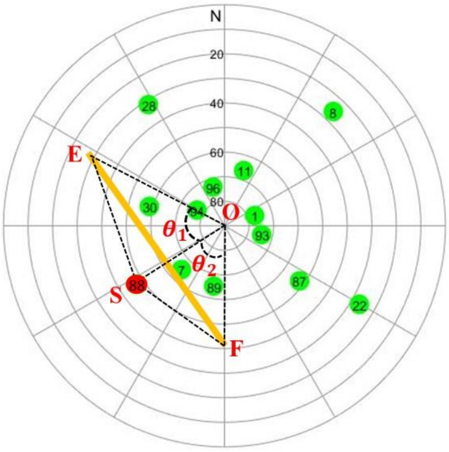

Figure 3-5 Skyplot of GNSS satellites and detected TEBs. Green and red circles and the numbers indicate satellites and corresponding PRNs. Line segment $\overline{EF}$ indicates the TEBs.

3.3.2 Coordinate Transformation

To implement the algorithm of NLOS detection and subsequent correction, satellites' visibility must be determined based on the extended TEBs. Thus, the relative poses of the GNSS receiver to satellites and to building surfaces need to be transformed into the same representation, the Skyplot. In each epoch, information from satellites, including azimuth, elevation angles, and SNR, can be obtained from the GNSS receiver. Part of satellite information can be represented as $\boldsymbol{SV}_{\mathbf{t}}^{\mathbf{all}} = \{\boldsymbol{SV_1}, \boldsymbol{SV_2}, \dots, \boldsymbol{SV_i}, \dots \boldsymbol{SV_n}\}$. n represents the number of satellites received. $\boldsymbol{SV_i}$ represents the information for satellite i, and $\boldsymbol{SV_i} = \{az_i, el_i, SNR_i, \rho_i\}$. az_i denotes the satellite azimuth angle. el_i represents the satellite elevation angle. SNR_i indicates satellite signal-to-noise ratio (SNR) and ρ_i denote the pseudorange measurement.

Satellite positions can be easily indicated in the Skyplot which is a 2-dimension coordinate based on corresponding elevation and azimuth angles. A transformation matrix should be employed for TEBs transformation from a 3D coordinate to a 2D coordinate. The transformation is conducted as per the following formula.

$$\boldsymbol{B}_{\text{build}}^{\text{skyp}} = \boldsymbol{B}_{build}^{3d}\boldsymbol{G}_{T}^{build} \tag{3–4}$$

where $\boldsymbol{B}_{build}^{3d}$ denotes the matrix of the building boundary presented in the previous sub-section. $\boldsymbol{G}_{T}^{build}$ is a 3x2 transform matrix. $\boldsymbol{B}_{build}^{skyp}$ denotes the boundary matrix (2x2) in Skyplot structured as follows:

$$\boldsymbol{B}_{\text{build}}^{\text{skyp}} = \begin{bmatrix} x_{skyE} & y_{skyE} \\ x_{skyF} & y_{skyF} \end{bmatrix} \tag{3–5}$$

After the transformation, satellites and building surface boundaries can be presented in the same coordinate frame, the Skyplot, as shown in Figure 3-5. Bounding box set $\boldsymbol{B}_{\text{t}}^{\text{seg\_buil}} = \{\boldsymbol{B}_1, \boldsymbol{B}_2, \dots, \boldsymbol{B}_i, \dots \boldsymbol{B}_m, t\}$ can be transformed into $\boldsymbol{B}_{\mathbf{t}}^{\mathbf{skyp}} = \{\boldsymbol{B}_1^{skyp}, \boldsymbol{B}_2^{skyp}, \dots, \boldsymbol{B}_i^{skyp}, \dots \boldsymbol{B}_m^{skyp}, t\}$, where $\boldsymbol{B}_{\text{i}}^{\text{skyp}}$ indicates the i_{th} *the* boundary in the Skyplot. Moreover, the distance list representing the distances from GNSS the receiver to the detected surfaces can also be obtained as $\boldsymbol{\alpha}_{\mathbf{t}}^{\mathbf{seg\_buil}} = \{\alpha_1, \alpha_2, \dots, \alpha_i, \dots \alpha_m, t\}$, where α_i is associated with $\boldsymbol{B}_{\mathbf{i}}^{\mathbf{skyp}}$. Line segment $\overline{EF}$ represents the building surface boundary corresponding to the line segment $\overline{EF}$ shown in Figure 3-4. Then, the azimuth and the elevation angles for point E and F can be calculated in the Skyplot, respectively.

3.4 Improved GNSS Positioning With NLOS Correction

In this section, an NLOS error model is presented first. Then, the NLOS detection criterion is proposed based on the detected TEBs, satellite elevation angle, azimuth angle, and SNR. NLOS error correction is then implemented. Finally, the GNSS positioning is conducted by applying the WLS method using the LOS and corrected NLOS pseudorange measurements.

3.4.1 NLOS Correction Based on Detected Building Boundary

In terms of the measurements from the GNSS receiver, each pseudorange measurement, ρ_n, is written as follows [64].

$$\rho_n = R_n + c(\delta t^{\text{r}} - \delta t_n^{\text{sv}}) + I_n + T_n + \varepsilon_n \tag{3-6}$$

where R_n is the geometric range between the satellite and the GNSS receiver. δt_n^{sv} denotes the satellite clock bias. δt^{r} indicates the receiver clock bias. I_n represents the ionospheric delay distance; T_n indicates the tropospheric delay distance. ε_n represents the errors caused by the multipath effects, NLOS receptions, receiver noise, antenna delay. In this paper, we focus on mitigating the NLOS errors caused by surrounding buildings.

The NLOS error model proposed in [3] is expressed in Figure 3-6. The expected signal transmission route is expressed as a dashed blue line in Figure 3-6. α represents the distance from the receiver to the building. θ_{ele} represents the elevation angle of the GNSS signal. We assume that:

(1) The building is vertical to the ground.

(2) GNSS signal reflection satisfies the law of reflection.

Thus, we can get $\theta_a = \theta_b$. Moreover, the direction of real signal transmission is parallel to the direction of expected signal transmission. Finally, we have $\theta_a = \theta_b = \theta_0 = \theta_{ele}$ .The route distance difference, γ, between the reflected signal and the expected signal is indicated as follows:

$$\gamma = \gamma_1 + \gamma_2 \tag{3–7}$$

$$\gamma_1 = \alpha sec\theta_{ele} \tag{3–8}$$

$$\gamma_2 = \gamma_1 \cos{(2\theta_{ele})} \tag{3–9}$$

Thus, the NLOS error can be calculated based on the azimuth angle, elevation angle, and the distance from the receiver to the building causing the reflection. In general, two steps are needed to proceed with the NLOS correction: NLOS detection and NLOS error calculation. The process of NLOS correction is summarized in detail in Algorithm 3-3.

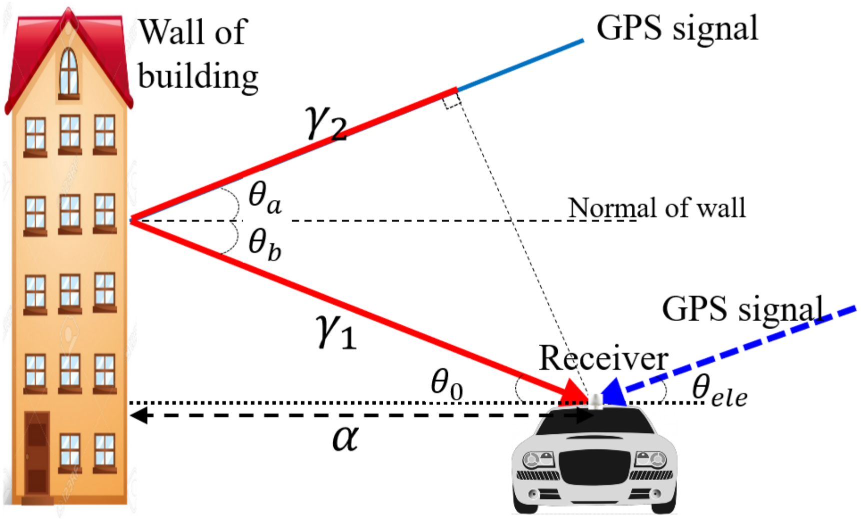

Figure 3-6 NLOS correction model. The signal is reflected by the building and subsequently received by the receiver [3].

Algorithm 3-3: NLOS detection and correction

Input: Satellites information set$\boldsymbol{SV_t^{all}} = \{\boldsymbol{SV_1}, \boldsymbol{SV_2}, \dots, \boldsymbol{SV_i}, \dots \boldsymbol{SV_n}\}$, building boundary matrix $\boldsymbol{B_t^{skyp}} = \left\{\boldsymbol{B_1^{skyp}}, \boldsymbol{B_2^{skyp}}, \dots, \boldsymbol{B_i^{skyp}}, \dots \boldsymbol{B_m^{skyp}}, t\right\}$, distance list $\boldsymbol{\alpha_t^{\mathrm{seg\_buil}}}$, area threshold $S_{threshold}$, SNR threshold $SNR_{threshold}$, the threshold of boundary uncertainty θ_{thres}

Output: corrected satellites information set after NLOS identification: $\boldsymbol{SV_t^{cor}} = \left\{\boldsymbol{SV_1^{cor}}, \boldsymbol{SV_2^{cor}}, \dots, \boldsymbol{SV_i^{cor}}, \dots \boldsymbol{SV_j^{cor}}\right\}$. LOS satellite assembles $\boldsymbol{SV_t^{los}} = \left\{\boldsymbol{SV_1^{los}}, \boldsymbol{SV_2^{los}}, \dots, \boldsymbol{SV_i^{los}}, \dots \boldsymbol{SV_k^{los}}\right\}$.

```
for all boundary $\boldsymbol{B_i^{skyp}}$ in $\boldsymbol{B_t^{skyp}}$do
  for all satellites $\boldsymbol{SV_i}$ in $\boldsymbol{SV_t^{all}}$do
    estimate $\theta_1, \theta_2$ as shown in Fig. 5
    Get triangle area $S_{\Delta SEO}$ of triangle SEO from $\boldsymbol{B_i^{skyp}}$
    Get triangle area $S_{\Delta SFO}$ of triangle SFO from $\boldsymbol{B_i^{skyp}}$
    Get triangle area $S_{\Delta SEF}$ of triangle SEF from $\boldsymbol{B_i^{skyp}}$
    Get triangle area $S_{\Delta EOF}$ of triangle EOF from $\boldsymbol{B_i^{skyp}}$
    $\Delta S = S_{\Delta SEO} + S_{\Delta SFO} + S_{\Delta SEF} - S_{\Delta EOF}$
    if $(SNR_i > SNR_{threshold}) or\ (\theta_1 < \theta_{thres})\ or\ (\theta_2 < \theta_{thres})$
      break
    if $\Delta S > S_{threshold}$ and $((\theta_1 + \theta_2) < \angle EOF < 180°$
      $\boldsymbol{SV_i^{los}} \leftarrow \boldsymbol{SV_i}$ // LOS
    else // NLOS
      $\boldsymbol{SV_i} \leftarrow \{az_i, el_i, SNR_i, \rho_i\}$
      $\boldsymbol{SV_i}(\rho_i) \leftarrow \boldsymbol{SV_i}(\rho_i) - (\gamma_1 + \gamma_2)$ // $\alpha_i$ from $\boldsymbol{\alpha_t^{\mathrm{seg\_buil}}}$
      $\boldsymbol{SV_i^{cor}} \leftarrow \boldsymbol{SV_i}$
    end if
  end for satellites set $\boldsymbol{SV_t^{all}}$
end for a boundary set $\boldsymbol{B_t^{skyp}}$
```

The inputs of Algorithm 3-3 include satellites information, SV_t^{all}, building surface boundaries information, $\boldsymbol{B_t^{skyp}}$, distance list, $\boldsymbol{\alpha_t^{\mathrm{seg\_buil}}}$ and some experimentally determined thresholds. The definitions of applied variables and functions in Algorithm 3-2 are listed as follows:

- $S_{threshold}$: Used to determine whether the satellite is inside the triangle. For example, if satellite 88 is inside triangle EOF.
- $SNR_{threshold}$: if the SNR for a certain satellite is more than this threshold, we treat it as LOS.
- θ_{thres}: threshold of boundary uncertainty.

The outputs are the corrected satellites information set $\boldsymbol{SV}_t^{cor}$ and LOS satellite assembles $\boldsymbol{SV}_t^{los}$. Firstly, geometry angle θ_1 (∠EOS) and θ_2 (∠FOS) shown in Figure 3-5 are estimated. Then areas of triangles $S_{\Delta SEO}$, $S_{\Delta SFO}$, $S_{\Delta SEF}$, and $S_{\Delta EOF}$ are calculated and ΔS can be estimated subsequently. Secondly, GNSS measurement whose SNR is larger than $SNR_{threshold}$ will not be excluded, as signals with strong SNR is not considered to be reflected by buildings.

Satellites whose positions are quite near the extended edge beam ($\theta_1 < \theta_{thres}$ or $\theta_2 < \theta_{thres}$) also should not be excluded, such as satellite 7 in Figure 3-5; thus, the angle threshold θ_{thres} is set. To avoid a faulty exclusion, a heuristically determined threshold $S_{threshold}$ is set. Satellites whose positions are quite near the TEBs of building surface should not be identified as NLOS, which can be determined by comparing ΔS and $S_{threshold}$, such as from satellite 7 in Figure 3-5. Finally, the pseudorange measurements from NLOS receptions can be corrected using the NLOS error model in formula (3-7).

In this case, these NLOS satellites can be detected, and corresponding pseudorange measurements are corrected.

3.4.2 GNSS Positioning Based on Corrected and Healthy Pseudorange Measurements

Measurements with low elevation angles are more likely to be a contaminated GNSS signal in an urban canyon, such as the multipath or NLOS, due to the reflection, blockage, and diffraction. Thus, the proper threshold must be set to exclude the unhealthy measurements. For satellite SV_i, if ele_i is less than ele_{thres}, it should be excluded from GNSS positioning. Pseudorange measurements in $\boldsymbol{SV}_t^{cor}$ and $\boldsymbol{SV}_t^{los}$ will be employed for GNSS positioning calculation.

The clock bias between the GNSS receiver and satellites is usually represented by the pseudorange measurements. The equation linking the receiver position and satellite can be structured as per the following formula using least squares (LS) method:

$$\hat{\boldsymbol{x}} = (\boldsymbol{G}^{\boldsymbol{T}}\boldsymbol{G})^{-\mathbf{1}}\boldsymbol{G}^{\boldsymbol{T}}\boldsymbol{\rho} \tag{3-10}$$

where $\boldsymbol{G}$ represents the observation matrix and is structured by unit LOS vectors between GNSS receivers' position and satellite position. $\hat{\boldsymbol{x}}$ indicates the estimated receiver position and ρ denotes the pseudorange measurements.

To better represent the quality of each measurement based on the information measured by the receiver, the weightings of each satellite are needed. The weightings for each satellite are calculated using the formulation in [65] by integrating the SNR and satellite elevation. Finally, the GNSS receiver position can be estimated using the WLS method as:

$$\hat{\boldsymbol{x}} = (\boldsymbol{G}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{G})^{-\mathbf{1}}\boldsymbol{G}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{\rho} \tag{3-11}$$

The weighting is given as follows [65]:

$$\boldsymbol{W}^{(i)}(ele_i, SNR_i) = \frac{1}{sin^2 ele_i}\left(10^{-\frac{(SNR_i - T)}{a}}\left(\left(\frac{A}{10^{-\frac{(F-T)}{a}}} - 1\right)\frac{(SNR_i - T)}{F - T} + 1\right)\right) \tag{3-12}$$

where $\mathbf{W}^{(i)}(ele_i, SNR_i)$ denotes the weighting for satellite $\boldsymbol{SV}_i$ .The parameter T indicates the threshold of SNR and is equal to $SNR_{threshold}$. Parameter a, A and F in (12) are experimentally determined. Then, the weighting matrix $\boldsymbol{W}$ is a diagonal matrix constituted by the weightings $\boldsymbol{W}^{(k)}(ele_i, SNR_i)$.

3.5 Experimental Evaluation and Conclusions

To evaluate the performance of the proposed method, two experiments conducted in two separate scenarios, are presented in this section. Firstly, the experimental setup is introduced in subsection 1. Experimental validations in two typical urban canyons are presented in subsection 3 and 4, respectively. The relationship between the satellite elevation angle and NLOS error is presented in subsection 4 before the discussion is given in subsection 5.

3.5.1 Experiment Setup

Experiments are conducted in two typical urban canyons (urban canyon 1 and urban canyon 2) of Hong Kong, and the experimental scenes are shown in Figure 3-7. The Skymask on the right-hand side demonstrates the degree of urbanization.

In both experiments, a u-blox M8T receiver is used to collect raw GPS and BeiDou measurements. A 3D LiDAR sensor, Velodyne 32, is employed to provide the real-time 3D point clouds scanned from the surroundings. Both the u-blox M8T receiver and the 3D LiDAR are installed on the top of an experiment vehicle, which can be seen on the left-hand side of Figure 3-7. The data were collected at a frequency of 1 Hz for GNSS and 10 Hz for the 3D LiDAR.

Besides, the NovAtel SPAN-CPT, GNSS RTK/INS (fiber optic gyroscopes) integrated navigation system is used to provide the ground truth of positioning. All the data are collected and synchronized using the Robot Operation System (ROS) [66]. Moreover, the coordinate systems of all the sensors are calibrated before the experiments.

The parameters used in this paper, which are experimentally determined, are shown in TABLE 3-1. Three GNSS positioning methods are compared:

(1) **WLS**: GNSS positioning with the WLS.

(2) **WLS-NE**: WLS with NLOS exclusion.

(3) **WLS-NC**: WLS with NLOS correction.

TABLE 3-1
PARAMETER VALUES USED IN THIS PAPER

Parameters	$S_{threshold}$	$SNR_{threshold}$	ele_{thres}	θ_{thres}
Value	10	45 dB-Hz	20°	5°
Parameters	a	A	F	
Value	30	32	10	

3.5.2 Evaluation of the Proposed Method in Urban Canyon 1

Figures 3-8 and TABLE 3-2 show the positioning results comparison of the conventional WLS, WLS-NE and the proposed method.

As can be seen from Figure 3-8, the total satellite numbers fluctuate between 5 and 13, with a mean satellite number of 10 during the experiment. With the aid of the proposed NLOS correction method, the positioning performance is improved at most of the epochs, which is indicated by the blue curve in the bottom panel of Figure 3-8. 30.29 meters of mean positioning error and 19.86 meters of standard deviation were obtained using the WLS method without any NLOS exclusion or correction. After the NLOS exclusion (all the NLOS are excluded), the mean error goes up to 35.25 meters. The main reason for this increase is due to the distortion of satellites' geometric distribution. In other words, the HDOP increases accordingly. According to the experiment, approximately 2–6 satellites are classified as NLOS due to the blockage from surrounding buildings. Therefore, the availability of GNSS positioning is decreased to about 92.5% due to the lack of satellites (at least five satellites are needed for GPS/BeiDou-based positioning calculation). The positioning error decreases to 22.86 meters using the proposed NLOS correction method. Moreover, the availability of GNSS positioning is also guaranteed. This result shows that the proposed NLOS correction model can obtain improved GNSS positioning performance

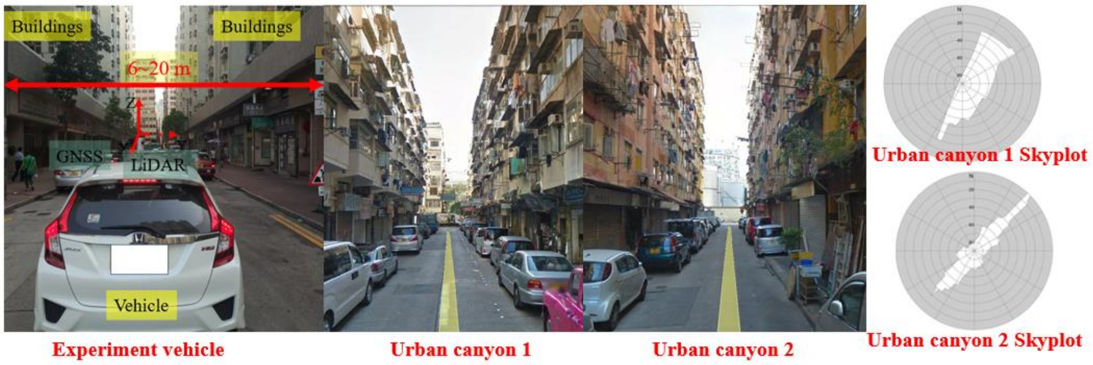

Figure 3-7. The sensors setup of the vehicle and tested environment: GNSS and LiDAR sensors are installed on the top of the vehicle shown in the left side of the figure. The two tested urban scenarios are shown in the middle of the figures. The Skyplot of the two experiments is shown on the right side of the figure.

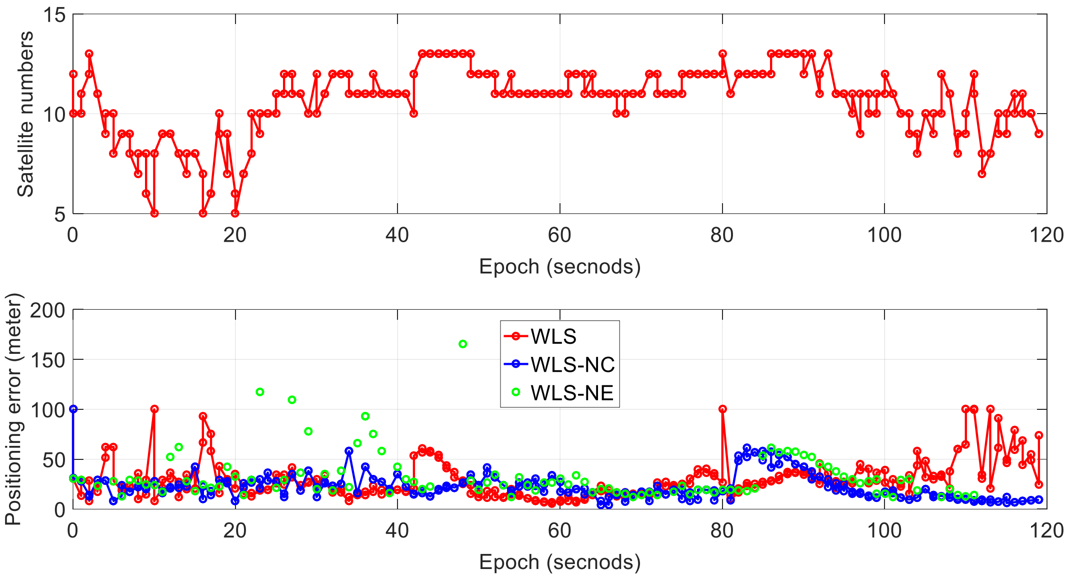

Figure 3-8. Positioning error of the GNSS before and after adding the NLOS correction, and NLOS exclusion in the urban canyon 1. The top panel indicates the satellite numbers. The bottom panel shows the positioning error: the red curve indicates the positioning error using WLS, the blue curve denotes the positioning based on proposed NLOS correction. The green curve shows the result of using WLS-NE.

TABLE 3-2
POSITIONING PERFORMANCE OF THE TWO METHODS IN URBAN CANYON 1 (IN THE UNIT OF METER)

All data	WLS	WLS-NE	WLS-NC
Mean error	30.29	35.25	22.86
Std	19.86	57.49	13.17
Availability	100%	92.5%	100%

3.5.3 Evaluation of the Proposed Method in Urban Canyon 2

Figure 3-9 and TABLE 3-3 show the positioning results comparison of the discussed three methods.

As can be seen from Figure 3-9, the total satellite numbers fluctuate between 8 and 15 with a mean satellite number of 11 during the experiment. With the aid of the proposed NLOS correction method, the positioning performance is improved through almost all the experiments. 42.15 meters of mean positioning error and 21.29 meters of standard deviation are obtained using the WLS method without any NLOS exclusion or correction. After the NLOS exclusion (all the NLOS are excluded), the mean error goes up to 47.74 meters. According to the experiment, approximately 3–7 satellites are classified as NLOS due to the blockage from surrounding buildings. Therefore, the availability of GNSS positioning is decreased to 88.69%. The mean positioning error is decreased to 26.7 meters using the proposed NLOS correction method. Moreover, the availability of GNSS positioning is also guaranteed. Interestingly, we can find that the variation trends of positioning error using two separate solutions are quite similar although the experiment.

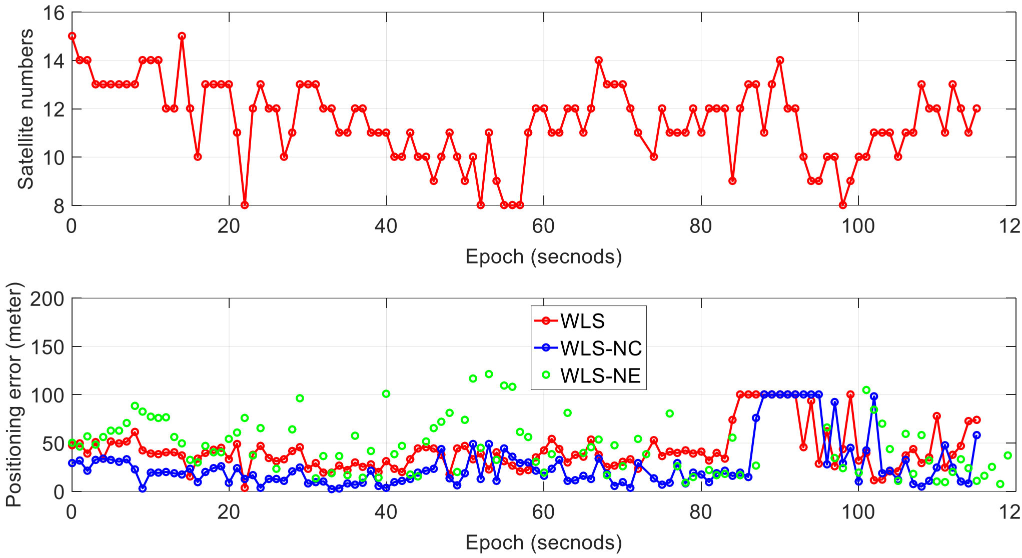

Figure 3-9. Positioning error of the GNSS before and after adding the NLOS correction in the urban canyon 2. The top panel indicates the satellite numbers. The bottom panel shows the positioning error: the red curve indicates the positioning error using WLS, the blue curve denotes the positioning based on proposed NLOS correction. The green curve shows the result using WLS-NE

TABLE 3-3
POSITIONING PERFORMANCE OF THE TWO METHODS IN URBAN CANYON 2 SCENARIO (IN THE UNIT OF METER)

All data	WLS	WLS-NE	WLS-NC
Mean error	42.15	47.74	26.70
Std	21.29	29.34	24.32
Availability	100%	88.69%	100%

3.5.4 Analysis of Satellite Elevation Angles versus NLOS Errors

Our previous work in [3] shows that the potential NLOS error is positively correlated to the satellite elevation angle. In other words, the NLOS satellite with higher elevation can cause a larger GNSS signal transmission delay. To show the relationship of GNSS positioning error and satellite elevation angle, we apply the manual correction in the collected data. Only satellites whose elevation angles are in a certain elevation angle range are corrected. The objective is to analyze the percentages of NLOS errors contributed by each elevation angle range of satellites.

TABLE 3-4 shows the results of three separate NLOS correction tests of urban canyon 1. Three satellite elevation angle ranges are given. If satellites 2, 5 and 24 are corrected whose elevation angles are between 20° and 36°, 5.5 meters of improvement are obtained with a mean positioning error of 24.79 meters. More than half of the results possess errors less than 15 meters. If satellites 29 and 88 are corrected whose elevation angles are between 36° and 54°, 1.65 meters of improvement are obtained with a mean positioning error of 28.64 meters. Only 17.01% of the results possess errors of less than 15 meters. There is almost no improvement if only satellite 13 is corrected whose elevation angle is between 54° and 72°.

TABLE 3-5 shows the results of three separate NLOS correction tests of urban canyon 2. Firstly, if the satellites 8, 17, 22 and 28, whose elevation angles are between 18°~36°, are corrected with the proposed method, the mean positioning error is decreased from 42.15 meters to 29.93 meters compared with the WLS method. 12.22 meters of improvement is obtained. Interestingly, the corresponding standard deviation also increases slightly. Approximately 79.64% of the positioning results have an error which is less than 30 meters. Secondly, only one satellite, satellite 88, possesses an elevation angle which is between 36°~54° and is NLOS. A slight improvement is introduced after the correction with a mean positioning error of 41.95 meters and a standard deviation of 21.80 meters respectively. 0.2 meters of improvement is obtained. Moreover, the percentage of positioning results which is more than 40 meters is similar to the results from WLS. Thirdly, two satellites, satellites 30 and 99 with an elevation which is between 54°~72°. A slight improvement (0.14 meters) is obtained with the proposed NLOS corrections. The corresponding percentages are similar to the result of the NLOS correction of the elevation range (36°~54°). In summary, the NLOS satellites with lower elevation (18°~36°) introduce larger positioning errors, compared with the NLOS satellites with higher elevation (36°~72°).

TABLE 3-4
POSITIONING PERFORMANCE OF WLSP-NC WITH MANUAL SATELLITE CORRECTION (IN THE UNIT OF METER) IN URBAN CANYON 1

All data	**Elevation (20°~36°)**	**Elevation (36°~54°)**	**Elevation (54°~72°)**
Mean error	24.79	28.64	30.1
Std	17.18	15.8	16.24
Percentage	51.62%	17.01%	16.69%

(<15 meters)			
Percentage (<30 meters)	84.66%	43.80%	40.43%
Percentage (>40 meters)	10.46%	25.70%	29.57%
Improvement	5.5	1.65	0.19
NLOS Satellites PRN	2,5,24	29,88	13

TABLE 3-5
POSITIONING PERFORMANCE OF WLSP-NC WITH MANUAL SATELLITE CORRECTION (IN THE UNIT OF METER) IN URBAN CANYON 2

All data	**Elevation (20°~36°)**	**Elevation (36°~54°)**	**Elevation (54°~72°)**
Mean error	29.93	41.95	42.01
Std	24.62	21.80	21.81
Percentage (<15 meters)	51.32%	7.96%	8.03%
Percentage (<30 meters)	79.64%	43.36%	39.29%
Percentage (>40 meters)	15.04%	28.32%	30.36%
Improvement	12.22	0.2	0.14
NLOS Satellites PRN	8,17,22,28	88	30,99

3.5.5 Discussion

1. When comparing the results from the urban canyon 1 and urban canyon, we can find that the improvement in urban canyon 2 is more distinct (from 42.15 meters to 26.70, 36.7% of improvement in total) than that in urban canyon 1 (from 30.29 meters to 22.86, 24.5% of improvement in total).
2. The remaining GNSS positioning errors still 22.86 meters in the middle urban and 26.70 meters, respectively. The major reason behind this is that the potential GNSS multipath contributes to the remaining error. On the other hand, the performance of the NLOS correction relies on the performance of distance α_i (from GNSS receiver to the reflector) estimation, as the signal from the NLOS satellite, can be reflected by different buildings. As shown in Figure 3-11, the NLOS can be caused by both building A and building B (reflector), thus causing different α_i. The proposed NLOS correction method can misidentify the reflector in some ways. In this case, this misidentification can result in positioning error.

Thus, the reflector detection will be studied in future work to improve the performance of the proposed NLOS correction method.

3. According to our previous research [67], the dynamic objects (such as double-decker bus) on the road can also cause NLOS receptions. The effects from the dynamic objects are not modeled which can also contribute to the remaining GNSS positioning error.

4. Performance sensitivity of the proposed method against building heights errors:

As the proposed method employs the building height list to extend the detected TEBs to the exact height. We implement an offline simulation to analyze the performance sensitivity of the proposed method against building heights errors. We collect 6 hours of satellite ephemeris data and do the LOS/NLOS classification based on 3D building models in Hong Kong. The building models are manually added with heights noise which is subject to Gaussian distribution ($\boldsymbol{N}(\mu, \delta^2)$). The offline processing setup is as follows:

(1) **Applied satellites**: GPS/BeiDou/GLONASS/Galileo.

(2) **Ground truth for LOS/NLOS classification**: the LOS/NLOS classification based on original 3D building models are treated as ground truth.

(3) **Accuracy of LOS/NLOS classification**: detected NLOS satellites number denoted by $N_{NLOS}^{Detected}$, ground truth number of NLOS satellites denoted by N_{NLOS}^{Label}. Accuracy (P_s) is calculated by $P_s = N_{NLOS}^{Detected}/N_{NLOS}^{Label} * 100\%$.

(4) The position of the assumed GNSS receiver is accurately set.

The detailed result is shown in Figure 3-10 over the 6 hours of ephemeris. 4 kinds ($\delta = 2, 4, 6, 8$) of building heights noise model (mean μ equals to zero) are applied. The percentages are shown in TABLE 3-6. About 99.07 % of the LOS/NLOS classification is obtained. With the increased noise standard deviation, the accuracy decreases gradually. However, even when the building heights noise standard deviation reaches 8 meters, accuracy still can reach 95.47%.

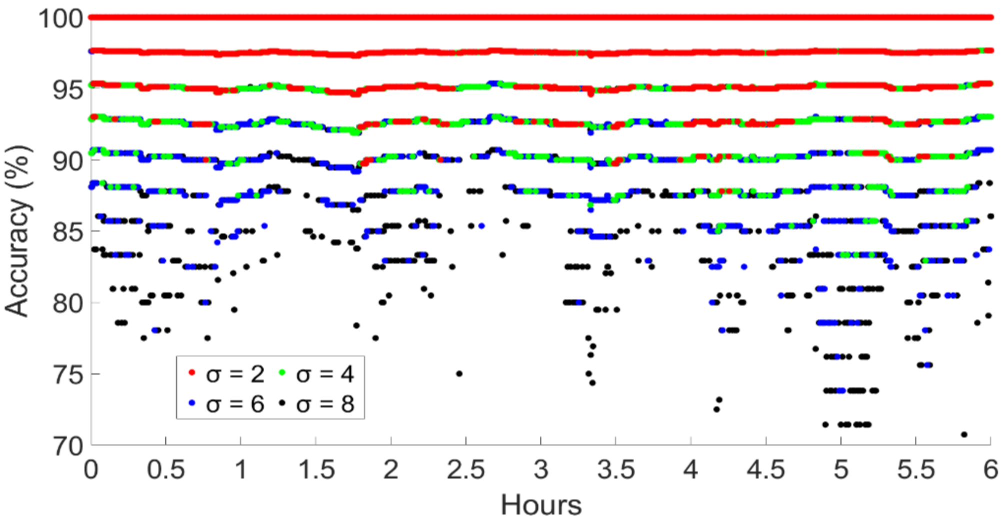

Figure 3-10 Relationship between the NLOS detection accuracy and simulated buildings heights error. The simulated building heights errors are subject to Gaussian distribution $\boldsymbol{N}(\mu, \delta^2)$. The color points denote the accuracy in different epochs. The solid lines represent fitted curves based on the result (in 6 hours) under different error noise models.

TABLE 3-6
PERFORMANCE SENSITIVITY OF PROPOSED METHOD AGAINST BUILDING HEIGHTS ERRORS

Yaw Bias	**Mean NLOS Detection Accuracy**	**Std**
$\boldsymbol{N(0,\ (2)^2)}$	99.07%	1.72%
$\boldsymbol{N(0,\ (4)^2)}$	97.81%	2.74%
$\boldsymbol{N(0,\ (6)^2)}$	96.69%	3.56%
$\boldsymbol{N(0,\ (8)^2)}$	95.47%	4.46%

We can conclude from the result that: 1) the building height can error can have a slight negative impact against the LOS/NLOS classification, thus can deteriorate the performance of the proposed method. 2) the proper way to identify the height of buildings can increase the robustness of the proposed method.

5. **Performance sensitivity of the proposed method against yaw (heading) angle errors**:

In this method, the yaw angle is derived from the highly accurate RTK GNSS/INS integrated navigation system. To analyze the performance sensitivity of the proposed method against yaw angle errors, we propose to manually add error noise to the yaw angle. The added error noise is subject to Gaussian distribution ($\boldsymbol{N}(\mu, \delta^2)$). After posing different noise magnitude ($\delta = 2, 4, 6, 8$) to the yaw angle, the NLOS detection accuracy is decreased accordingly. Be noted that the NLOS detection accuracy calculation is the same as that in subsection E-6.

If the yaw angle with noise error model $\boldsymbol{N(0, (1°)^2)}$ is applied, the performance of the proposed method remains the same. However, the mean positioning error increased from 26.70 to 29.56 meters after increase the δ to $\mathbf{2°}$. The NLOS detection accuracy is also reduced to 98.2%. If the δ is set at $\mathbf{4°}$ and $\mathbf{6°}$, the NLOS detection accuracies are decreased to 95.43% and 93.01%, respectively. Meanwhile, the mean positioning errors and standard deviations are slightly increased. We can conclude that: the yaw angle error can hurt the performance of the proposed method.

TABLE 3-7
PERFORMANCE SENSITIVITY OF PROPOSED METHOD AGAINST YAW ERRORS IN URBAN CANYON 2 DATASET

Yaw Bias	**NLOS Detection Accuracy**	**Mean positioning error (m)**	**Std**
$\boldsymbol{N(0, (1°)^2)}$	100%	26.70	24.32
$\boldsymbol{N(0, (2°)^2)}$	98.2%	29.56	24.75
$\boldsymbol{N(0, (4°)^2)}$	95.43%	30.41	24.93
$\boldsymbol{N(0, (6°)^2)}$	93.01%	31.39	25.51
$\boldsymbol{N(0, (8°)^2)}$	92%	31.10	27.39

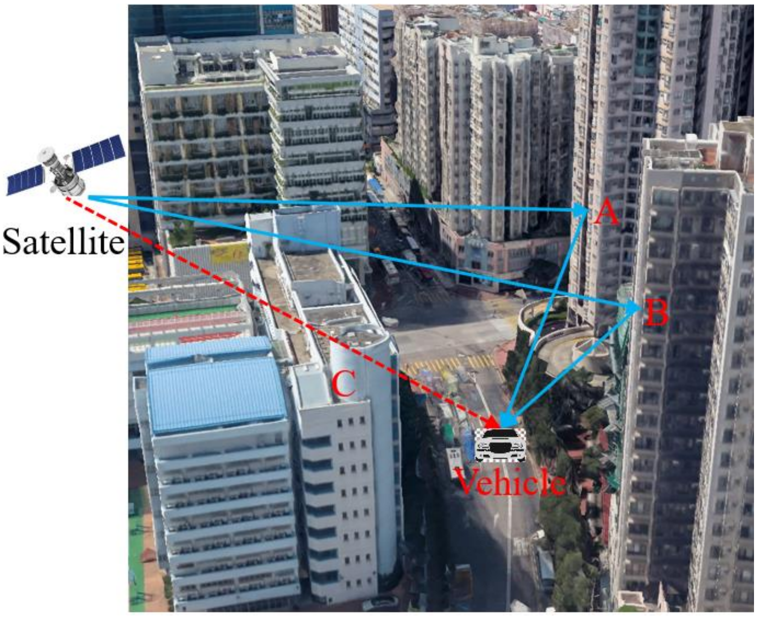

Figure 3-11. Illustration of multiple NLOS signal transmission routes. The direct route from the GNSS satellite to the vehicle is blocked by building C. However, the signal

can be reflected by building A or building B. As a result, the vehicle can receive the GNSS signal as well.

3.5.6 Conclusions

In this method, we propose an NLOS correction and an improved GNSS positioning method aided by 3D LiDAR. Innovatively, the top boundary of the building is detected using the 3D LiDAR-based point cloud segmentation method and NLOS satellites are detected based on the detected TEBs. The NLOS is corrected using an NLOS error model, instead of direct exclusion. The GNSS positioning is conducted based on corrected and healthy LOS satellites. The evaluated results show that the proposed method can obtain improved GNSS positioning accuracy compared with the standalone WLS.

The paper proposes to cope with the effects of static buildings on GNSS positioning using 3D LiDAR. In the future work, we propose to integrate a sky-pointing camera with LiDAR to correct the NLOS receptions, therefore, to improve the GNSS positioning. As the camera can capture the sky view in a real-time manner. In this case, the camera will play the role to describe Skyplot with obstacles and LiDAR is used to provide the distances between the vehicle and the obstacles. Moreover, the yaw angle in this paper can be provided by the LiDAR-based positioning [68].

4. OBJECT DETECTION AIDED GNSS AND ITS INTEGRATION WITH LIDAR POSITIONING

4.1 Introduction

Autonomous vehicles [1, 42] receive increasing attention due to its immense potential market. To fully activate autonomous vehicles, the globally referenced and meter-level positioning is required in all scenarios. LiDAR is a commonly used sensor for autonomous driving which is not only be used for object detection [44, 45] and also is employed to provide continuous positioning [69]. In LiDAR-based positioning, the SLAM [70, 71] algorithm is usually employed to calculate the transformation between the consecutive point clouds provided by LiDAR. However, the localization from the SLAM can introduce accumulated error over time, and only relative positioning is obtained which cannot satisfy the requirement of the autonomous vehicles. With the rise of multi-GNSS, the availability of satellites has been significantly enhanced, which makes it possible to receive enough satellites for GNSS positioning even in an urban canyon. GNSS is currently a significant source providing continuous global positioning. It is usually integrated with the LiDAR-based localization to take advantage of both positioning sources [22, 53, 56, 57, 72]. Based on the principle of sensor fusion, the sensor integration methods can be divided into two groups, the filtering-based, and the smoothing-based integrations. The symbolic filtering based sensor integration method is the Bayes filter, including Kalman filter [73, 74], information filter [75-77] and particle filter [78-80]. The Bayes filter-based sensor integration estimates the current state based on current observation and the previous state estimation, failing to make use of all the states before the previous states. This is because of the assumption of the first order of the Markov model [81] which is one of the key assumptions of the Bayes filter. Conversely, the smoothing approaches [82-85] estimate the pose and map by considering the full sets of measurements from the first epoch to the current epoch. The most well-known smoothing method is the graph-based SLAM [86]. These GNSS/LiDAR integration solutions can obtain decent positioning performance in sparse areas as shown in Figure 4-1. However, performance can be severely challenged in the super-urbanized area due to poor performance and large uncertainty in GNSS positioning. GNSS can achieve 5~10 meters regarding positioning accuracy in an open area or sparse scenarios base on the conventional single point positioning (SPP).

However, the positioning error can significantly increase to ~50 meters in super-urbanized areas [3], due to the reflection and blockage from the surrounding buildings. The reflection can cause extra travel delay in the pseudorange domain, thus causing the well-known multipath effects and NLOS receptions. Moreover, the uncertainty of the GNSS positioning is also greatly increased due to the severe NLOS.

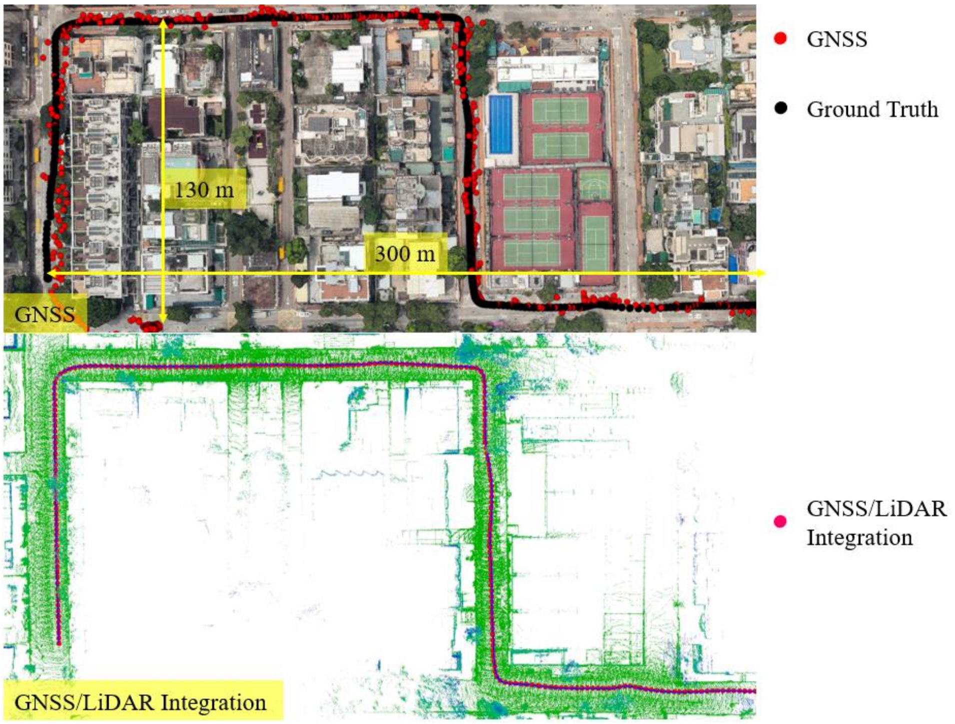

Figure 4-1. The top panel shows the performance of GNSS standalone positioning. The red circle indicates the GNSS positioning result and the black circle is for the ground truth. The bottom panel shows the GNSS/LiDAR integration, where the green points represent the point cloud map and the pink circles represent the trajectory generated by GNSS/LiDAR integration.

According to a recent survey [7], the NLOS is the dominant component to blame for positioning errors in dense urban areas. Numerous studies [17, 28, 30-32] are conducted to identify the NLOS receptions. Because NLOS is caused by the reflection from buildings, the 3D city models are employed to identify the NLOS receptions [4, 9, 18, 87-89]. With the aided of the 3D city models, the possible blockage from the buildings can be detected, and the corresponding NLOS is obtained. Then the NLOS measurements are excluded from the GNSS positioning subsequently. However, this method relies heavily on the availability of the 3D city models which is the main problem for its implementation. The range-based 3D map aided GNSS (3DMA GNSS) [17, 30-32] is one of the most mature techniques to mitigate the positioning errors from

NLOS receptions. It innovatively employs the ray-tracing simulation to simulate the possible transmission routes of the GNSS signals. Thus, the travel delay can be calculated based on the simulated signal transmission route. The NLOS measurement is also corrected and used in the further GNSS positioning calculation. However, this method introduces a heavy computational load due to the ray-tracing simulation. Moreover, 3D city models are also needed, and those map-aided GNSS positioning methods rely heavily on the initial guess of the receiver.

The other problem in GNSS/LiDAR integration is the large uncertainty of GNSS positioning in urbanized areas. The uncertainty is referred to as the noise covariance which is essential in the GNSS/LiDAR integration. Satisfactory performance can be obtained using the GNSS/LiDAR integration scheme on the condition that each sensor noises are well modeled. However, the researches in [53, 56, 57, 72] model the GNSS positioning uncertainty as Gaussian distribution. This rough modeling of GNSS positioning uncertainty can work in the places where GNSS positioning is robust and accurate with few NLOS receptions. However, the GNSS positioning error does not subject to Gaussian distribution any more in the urbanized area [3]. The conventional constant and Gaussian distribution-based covariance cannot model the actual performance of the GNSS positioning. As a result, the GNSS/LiDAR integration result can introduce additional positioning errors. A GNSS covariance estimation solution based on satellite numbers and signal to noise ratio (SNR) which can obtain improvements [90] comparing with the conventional constant covariance solution. However, this scheme cannot effectively model the positioning error from NLOS. Taking advantage of the 3D LiDAR sensor, the 3D point cloud map is employed to identify the NLOS measurement [36]. This method can effectively detect the NLOS signals which are similar to the methods in [4, 91] using the 3D city models. Then, the NLOS receptions are all excluded from the GNSS positioning. The point cloud map plays a similar role in the 3D city maps. However, this implementation of this method is subjected to the availability of the fully 3D point cloud map of buildings, which is difficult to obtain. Moreover, the GNSS positioning uncertainty is simply estimated based on the new horizontal dilution of precision (HDOP), and the actual NLOS errors are not effectively modeled.

This Section 4 innovatively employs the 3D LiDAR to facilitate the GNSS positioning and corresponding covariance estimation using the real-time point clouds-based object detection. Then, the improved GNSS positioning is integrated with LiDAR odometry under a graph-based SLAM framework. The flowchart of the GNSS/LiDAR integration solution is shown in Figure 4-2.

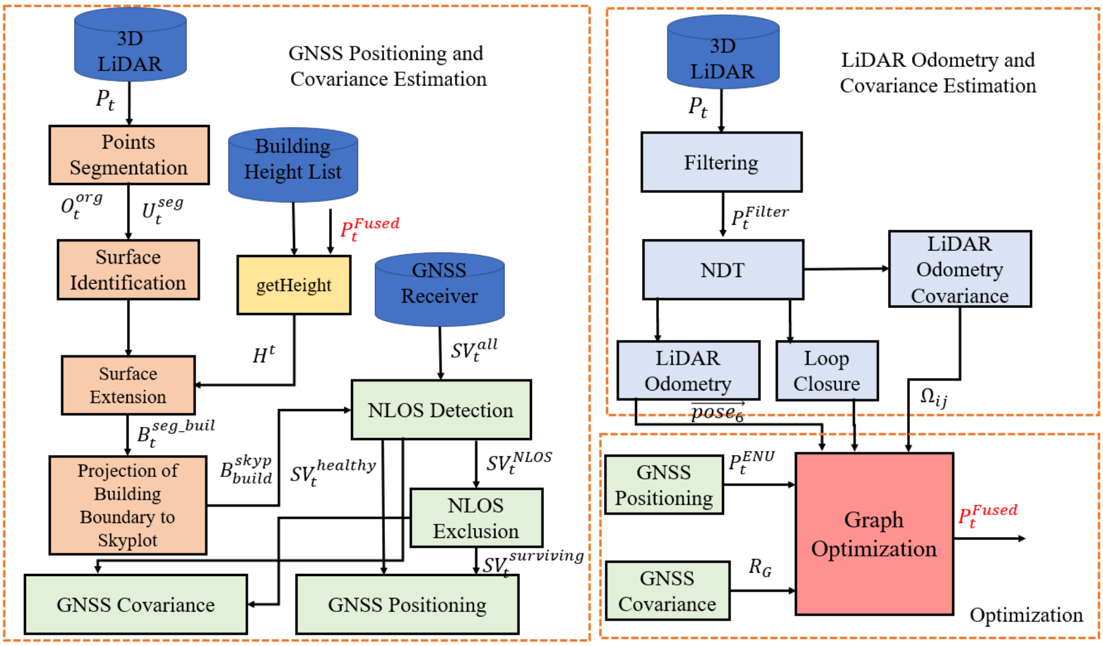

Figure 4-2 The flowchart of the proposed GNSS/LiDAR integration method. Three parts are included: (a). GNSS positioning and its covariance estimation, (b). LiDAR odometry and its covariance estimation and (c). The graph-based optimization.

Firstly, the building boundary is detected based on the algorithm proposed in the previous work [92] of our research team. The point clouds are fixed to the GNSS frame based on the orientation obtained from LiDAR odometry (shown in Figure 4-2). The satellites and the building boundary are both projected to a GNSS Skyplot [48]. Secondly, the NLOS detection is conducted based on a proposed NLOS detection algorithm. GNSS measurements suffered from both NLOS, and low elevation angle is excluded based on a proposed FDE algorithm. Then, the GNSS positioning is conducted based on the survived GNSS measurements. Thirdly, the GNSS positioning covariance is calculated by considering the potential positioning errors caused by NLOS receptions and line-of-sight (LOS) receptions. Finally, the improved GNSS positioning result and corresponding covariance are integrated with the LiDAR odometry using a graph-based SLAM framework

4.2 LIDAR Odometry and Its Covariance Estimation

4.2.1 LiDAR Odometry

The principle of LiDAR odometry [93] is to track the transformation between two successive frames of point clouds by matching the two frames (called as a reference and an input point cloud in this paper). The matching process is also called point cloud registration. The objective of point cloud registration is to obtain the optimal transformation matrix to match or align the reference and the input point clouds. The most well-known method of point cloud registration is the iterative closest point (ICP) [94]. The ICP is a straightforward method to calculate the transformation matrix between two consecutive scans by iteratively searching pairs of nearby points in the two scans and minimizing the sum of all point-to-point distances. The objective function can be expressed as follows [94]:

$$C(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{T}}) = \arg\min \textstyle\sum_{i=1}^{N} ||(\boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{T}) - \boldsymbol{q}_i||^2 \tag{4-1}$$

where the N indicates the number of points in one scan $\boldsymbol{p}$, $\boldsymbol{R}$ and $\boldsymbol{T}$ indicate the rotation and translation matrix, respectively, to transform the input point cloud ($\boldsymbol{p}$) into the reference point cloud ($\boldsymbol{q}$). Objective function $C(\widehat{\boldsymbol{R}},\widehat{\boldsymbol{T}})$ indicates the error of the transformation. The main drawback of this method is that ICP can easily get into the local minimum problem. The normal distribution transform [95] (NDT) is a state-of-art method to align two consecutive scans with modeling of points based on Gaussian distribution. The NDT innovatively divides the point cloud's space into cells. Each cell is continuously modeled by a Gaussian distribution. In this case, the discrete point clouds are transformed into successive continuous functions. In this paper, the NDT is employed as the point cloud registration method for the LiDAR odometry. Assuming that the transformation between two consecutive frames of point clouds can be expressed as $\overrightarrow{\boldsymbol{pose}_6} = [t_x \quad t_y \quad t_z \quad \phi_x \quad \phi_y \quad \phi_z]^T$. The t_i indicates the translation in the *x*, *y*, and *z*-axis, respectively. The ϕ_x represents the orientation angle of the roll, pitch, and yaw, respectively. Steps of calculating the relative pose between the reference and the input point clouds are as follows:

1) Fetch all the points $\boldsymbol{x}_{i=1\dots n}$ contained in a 3D cell [96].

   Calculate the geometry mean $\boldsymbol{q} = \frac{1}{n}\sum_i \boldsymbol{x}_i$.

   Calculate the covariance matrix

$$\boldsymbol{\Sigma} = \frac{1}{n}\textstyle\sum_i(\boldsymbol{x}_i - \boldsymbol{q})(\boldsymbol{x}_i - \boldsymbol{q})^T \tag{4-2}$$

2) The matching score is modeled as:

$$f(\boldsymbol{p}) = -\text{score}(\boldsymbol{p}) = \textstyle\sum_i \exp\left(-\frac{(\boldsymbol{x}_i{}' - \boldsymbol{q}_i)^T\boldsymbol{\Sigma}_i{}^{-1}(\boldsymbol{x}_i{}' - \boldsymbol{q}_i)}{2}\right) \tag{4-3}$$

where $\boldsymbol{x}_i$ indicates the points in the current frame of scan $\boldsymbol{p}$. $\boldsymbol{x}_i'$ denotes the point in the previous scan mapped from the current frame using the $\overrightarrow{pose_6}$. $\boldsymbol{q}_i$ and $\boldsymbol{\Sigma}_i$ indicate the mean and the covariance of the corresponding normal distribution to point $\boldsymbol{x}_i'$ in the NDT of the previous scan.

3) Update the pose using the Quasi-Newton method based on the objective function to minimize the score, $\boldsymbol{f}(\boldsymbol{p})$.

With all the points in one frame of point clouds being modeled as cells, the objective of the optimization for NDT is to match current cells into the previous cells with the highest probability. The optimization function $f(\boldsymbol{p})$ can be found in [95]. In each cell containing several points, the corresponding covariance matrix can be calculated and represented by $\boldsymbol{\Sigma}$. The shape (circle, plane or linear) of the cell is indicated by the relations between the three eigenvalues of the covariance matrix [95]. In this case, comparing with the conventional ICP algorithm, the NDT innovatively optimize the transformation by considering the features of points. The loop closure detection is conducted based on these shape features [97].

4.2.2 Covariance Estimation of LiDAR Odometry

The LiDAR odometry can provide continuous relative pose estimation, $\overrightarrow{\boldsymbol{pose}_6}$. The associated covariance of this pose estimate is essential for the later integration with the GNSS positioning. During the NDT process, the covariance of pose estimation is related to the uncertainty of the matching between the reference and the input point clouds. In the graph-based optimization which will be introduced in Section IV, the covariance is indicated as the inverse of the information matrix $\boldsymbol{\Omega}_{ij}$. In each matching process between a point from the reference point cloud and a point from the input point cloud, we model the degree of matching as:

$$dm_{ij} = \frac{1}{n}\sum_{k=1}^{n}\sqrt{\Delta x_k^2 + \Delta y_k^2 + \Delta z_k^2} \qquad (4\text{-}4)$$

where the dm_{ij} represent the degree of matching between the reference and the input point clouds. *n* represents the number of points in the input point cloud. Δx_k indicates the positional difference in the *x*-axis between input and reference points after the convergence of NDT is obtained. Δy_k and Δz_k indicate the positional differences in *y* and *z*-axis, respectively. Thus, the information matrix $\boldsymbol{\Omega}_{ij}$ of the degree of matching between reference and input can be expressed as:

$$\boldsymbol{\Omega}_{ij} = \begin{bmatrix} \boldsymbol{\Omega}_{ij}^{p} & 0 \\ 0 & \boldsymbol{\Omega}_{ij}^{r} \end{bmatrix} \tag{4-5}$$

$$\boldsymbol{\Omega}_{ij}^{p} = \boldsymbol{I}/(\mathrm{C}_{p}^{2} dm_{ij}) \tag{4-6}$$

$$\boldsymbol{\Omega}_{ij}^{r} = \boldsymbol{I}/(\mathrm{C}_{r}^{2} dm_{ij}) \tag{4-7}$$

where $\boldsymbol{I}$ indicate identity matrix, C_{p}^{2} is a coefficient that is heuristically determined. In this case, the covariance for LiDAR odometry is correlated with the degree of matching. Usually, the LiDAR matching can obtain relatively pose estimates on the lateral direction crossing the building. However, the longitudinal pose estimate is not as accurate as of the lateral one because the building surface tends to be plain and featureless. Thus, the covariance should be adaptively changed according to the degree of matching in different scenarios.

4.3 Object Detection Aided GNSS Positioning And Its Covariance Estimation

In this section, the detection of the building boundary is presented firstly. The NLOS fault detection and exclusion (FDE) method is presented subsequently. Secondly, GNSS positioning is implemented based on the NLOS FDE. Finally, the innovative covariance estimation of GNSS positioning is introduced.

4.3.1 Clustering for Double-decker Bus Detection

To identify which satellite is blocked by the surrounding buildings, the pose of the building boundaries relative to the GNSS receiver is needed. As the 3D LiDAR can provide sufficient points representing the environments, our previous work in [92] presents the detection of double-decker bus and dimensions extension algorithm based on LiDAR-based object detection. Building boundary detection is based on a similar approach. The process of building boundary detection is listed as follows:

1) Segment the point clouds into clusters to represent different objects.
2) Identify the building surface and extend the surface dimensions to the actual dimensions using Algorithm 4-1.
3) Obtain the bounding box indicating the building surface, and the corresponding top boundary.
4) Calculate the pose of the building relative to the GNSS receiver.

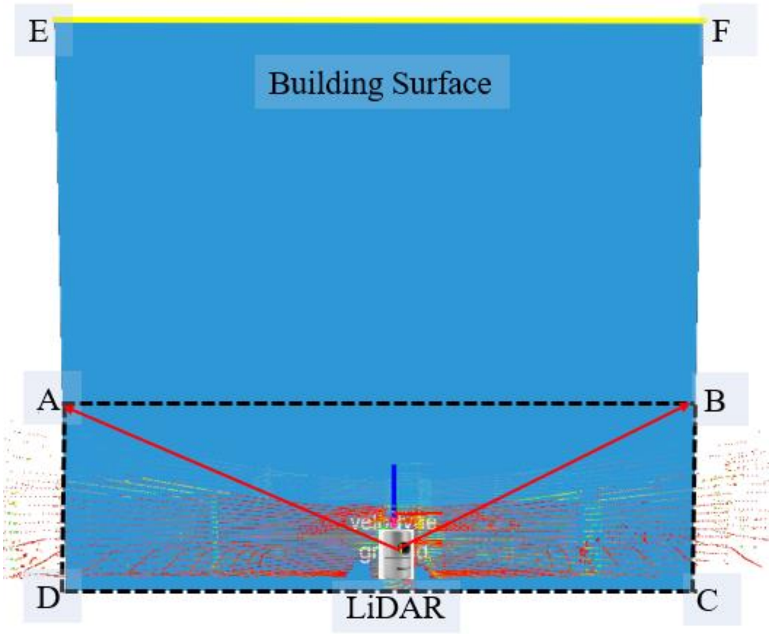

Figure 4-3 Illustration of point sets segmentation and building surface identification. Blue box ABCD represents the initially detected building surface. Blue box CDEF represents the extended building surface.

The inputs of Algorithm 4−1 are $\boldsymbol{U}_t^{seg}$ and $\boldsymbol{O}_t^{org}$ obtained from the segmentation based on the work in [92], point number threshold num_{thres}, length threshold len_{thres} and height threshold hei_{thres}, building height list $\boldsymbol{H}_{build}$, receiver position $\boldsymbol{P}_t^{fused}$, yaw angle $Yaw_r$from GNSS/LiDAR integration. The output is the bounding box set $\boldsymbol{B}_t^{seg\_buil}$ specifically represent the building surface. Each bounding box is indicated by $\boldsymbol{U}_i = [x_i^c, y_i^c, z_i^c, roll_i^c, pitch_i^c, yaw_i^c, d_i^{len}, d_i^{wid}, d_i^{hei}]$. The function *Num* mentioned in Algorithm 1 is used to calculate the points number of each cluster $\boldsymbol{O}_i$. The function $getHeight$ is used to search the height information from a saved building height list containing the rough height information. To determine the actual height of the identified building surface, $\boldsymbol{P}_t^{fused}$, $\boldsymbol{U}_i$ and Yaw_r are also needed. $\boldsymbol{P}_t^{fused}$ indicates the GNSS position given by previous epoch positioning result from GNSS/LiDAR integration. The relative position between the GNSS receiver and detected building can be obtained from $\boldsymbol{U}_i$. Moreover, the yaw angle can be acquired from GNSS/LiDAR integration. For each bounding box $\boldsymbol{B}_i$, the distance α_i from the receiver to the detected building surface can be calculated as follows:

$$\alpha_i = \sqrt{(x_i^{c2} + y_i^{c2} + z_i^{c2})} \tag{4–8}$$

Algorithm 4–1: Building surface identification from Bounding Box sets and height extension

Input: Bounding Box sets $\boldsymbol{U}_t^{seg} = \{\boldsymbol{U}_1, \boldsymbol{U}_2, \dots, \boldsymbol{U}_i, \dots \boldsymbol{U}_n, t\}$, Organized point clusters $\boldsymbol{O}_t^{org} = \{\boldsymbol{O}_1, \boldsymbol{O}_2, \dots, \boldsymbol{O}_i, \dots \boldsymbol{O}_n, t\}$, point number threshold num_{thres}, length threshold len_{thres} and height threshold hei_{thres}, building height list $\boldsymbol{H}_{build}$, receiver position $\boldsymbol{P}_t^{fused}$, yaw angle Yaw_r

Output: Bounding Box set represent building surfaces $\boldsymbol{B}_t^{seg\_buil} = \{\boldsymbol{B}_1, \boldsymbol{B}_2, \dots, \boldsymbol{B}_i, \dots \boldsymbol{B}_n, t\}$

```
set up an empty clusters list $\boldsymbol{B}_t^{seg\_buil}$ to save bounding box
for all bounding box $\boldsymbol{U}_i$ in $\boldsymbol{U}_t^{seg}$ do
  if Num$(O_i) > num_{thres}$
    $\boldsymbol{U}_i \leftarrow [x_i^c, y_i^c, z_i^c, roll_i^c, pitch_i^c, yaw_i^c, d_i^{len}, d_i^{wid}, d_i^{hei}]$
    if $d_i^{len} > len_{thres}$ and $d_i^{hei} > hei_{thres}$
      $d_i^{hei} \leftarrow getHeight(\boldsymbol{H}_{build}, \boldsymbol{P}_t^{fused}, \boldsymbol{U}_i, Yaw_r)$
      $\boldsymbol{B}_i \leftarrow \boldsymbol{U}_i$
    end if
  end if
end for $\boldsymbol{U}_t^{seg}$
```

Thus, the bounding box with extended height representing the building surface can be identified using Algorithm 4-1. The bounding box is extended from rectangles ABCD to CDEF as can be seen in Figure 4-3. The bounding boxes ABCD and CDEF indicate the initially detected dimensions and the extended dimensions of the building, respectively. Then, the boundary parameters for the bounding box $\boldsymbol{B}_i$ corresponding to building surface is denoted by the line segment $\overline{EF}$ denoted as $\boldsymbol{B}_{build}^{3d}$, the matrix of bus boundary. To represent the building, two points, E and F, are required. The $\boldsymbol{B}_{build}^{3d}$, which is relative to the LiDAR coordinate system, is structured as follows:

$$\boldsymbol{B}_{build}^{3d} = \begin{bmatrix} x_{3dE} & y_{3dE} & z_{3dE} \\ x_{3dF} & y_{3dF} & z_{3dF} \end{bmatrix} \tag{4-9}$$

In this case, the top boundary of the building is detected which is used for NLOS detection in the following sub-section. The distance between the receiver and the building surface is calculated as α_i which will be used for the covariance estimation of GNSS positioning in the following sections.

4.3.2 NLOS Detection and Exclusion

The boundary of the building is detected as $\boldsymbol{B}_{build}^{3d}$. The satellites and the building boundary can be projected into a GNSS Skyplot which is shown in Figure 4-4. The circles indicate the satellites and the associated number represents the satellite index. The yellow line indicates the building boundary projected into the Skyplot. The NLOS is indicated with a red circle in Figure 4-4. Assume that the initial satellites set are $\boldsymbol{SV}_t^{all} = \{\boldsymbol{SV}_1, \boldsymbol{SV}_2, \ldots, \boldsymbol{SV}_i, \ldots \boldsymbol{SV}_s\}$, where $\boldsymbol{SV}_i = \{az_i, el_i, SNR_i, \rho_i\}$. az_i and el_i denote the azimuth and elevation angles of a satellite, respectively, SNR_i indicates satellite SNR and ρ_i denotes the pseudorange measurement. The satellite visibility classification based on satellite information and boundaries is introduced in the previous work [67] of our research team. According to Fig. 5, we can have two satellite sets. One is the satellites set $\boldsymbol{SV}_t^{nlos} = \{\boldsymbol{SV}_1, \boldsymbol{SV}_2, \ldots, \boldsymbol{SV}_i, \ldots \boldsymbol{SV}_n\}$ containing only the NLOS ones. The other one is the LOS satellite set $\boldsymbol{SV}_t^{los} = \{\boldsymbol{SV}_1, \boldsymbol{SV}_2, \ldots, \boldsymbol{SV}_i, \ldots \boldsymbol{SV}_l\}$ and $s = n + l$ is satisfied.

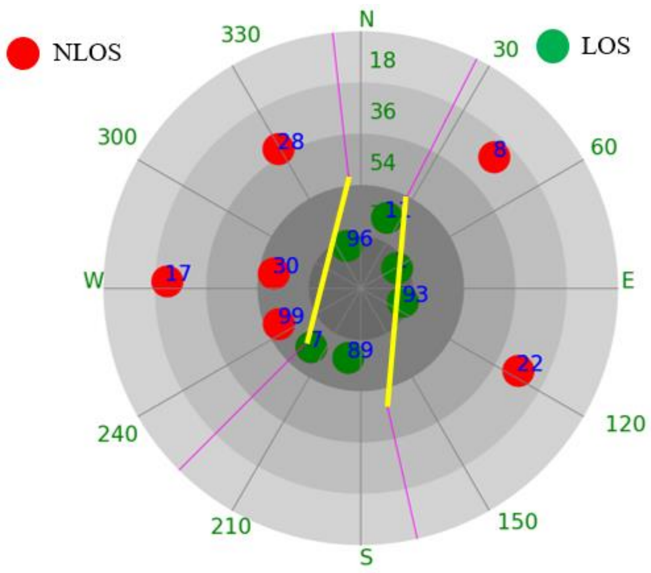

Figure 4-4 GNSS Skyplot indicates the satellites distribution and building boundary

We can see from the Skyplot in Figure 4-4, the majority of the satellites are blocked (6 blocked out of 12 satellites). Almost only satellites with elevation angle more than 72 degrees are not blocked. The exclusion of all the NLOS receptions can result in a significant increase in the HDOP which will magnify the pseudorange errors in GNSS positioning. In other words, the full exclusion of NLOS receptions will conversely deteriorate the GNSS positioning result. In the previous work, we analyze NLOS errors in [3]. The pseudorange error is smaller while the higher of elevation angle and smaller the distance from GNSS receiver to the reflector (refers to α_i). In other words, the NLOS with lower elevation angle can introduce a larger GNSS positioning error. The relations between the satellite elevation, α_i and pseudorange error are presented in [3].

Inspired by this result [98], this paper proposes to exclude the measurement based on the satellite elevation angle and the HDOP of satellite distribution. The proposed satellite exclusion method is shown in Algorithm 4-2. The inputs of the algorithm include the NLOS satellites information sets $\boldsymbol{SV}_t^{nlos} = \{\boldsymbol{SV}_1, \boldsymbol{SV}_2, \dots, \boldsymbol{SV}_i, \dots \boldsymbol{SV}_n\}$ and $\boldsymbol{SV}_t^{los}$. Only the satellites blocked by buildings are contained in the satellite set $\boldsymbol{SV}_t^{nlos}$ (for example, the satellite 8, 17, 28, 22, 30, 39 shown in Figure 4-4). The thresholds for elevation angle and HDOP are also the inputs of Algorithm 4-2. The output of Algorithm 4-2 is the satellites set survived from this NLOS exclusion process, indicated as $\boldsymbol{SV}_t^{los\_nlos} = \{\boldsymbol{SV}_1^s, \boldsymbol{SV}_2^s, \dots, \boldsymbol{SV}_i^s, \dots \boldsymbol{SV}_m^s\}$.

After the proposed NLOS exclusion, part of the NLOS measurements is excluded from low elevation angles. The survived NLOS and LOS measurements are saved into the $\boldsymbol{SV}_t^{los\_nlos}$. This satellite set is used for GNSS positioning using the WLS method in the following sub-section.

Algorithm 4-2: Proposed NLOS Exclusion

Input: Satellites information set $\boldsymbol{SV}_t^{nlos} = \{\boldsymbol{SV}_1, \boldsymbol{SV}_2, \dots, \boldsymbol{SV}_i, \dots \boldsymbol{SV}_n\}$, $\boldsymbol{SV}_t^{los} = \{\boldsymbol{SV}_1, \boldsymbol{SV}_2, \dots, \boldsymbol{SV}_i, \dots \boldsymbol{SV}_l\}$, elevation angle threshold $el_{threshold}$, the threshold of HDOP H_{thres}

Output: corrected satellites set after NLOS exclusion: $\boldsymbol{SV}_t^{los\_nlos} = \{\boldsymbol{SV}_1^s, \boldsymbol{SV}_2^s, \dots, \boldsymbol{SV}_i^s, \dots \boldsymbol{SV}_m^s\}$

```
Step 1: sort the satellites set in $\boldsymbol{SV}_t^{nlos}$ based on elevation angle from
        small to large
Step 2: exclude satellite $\boldsymbol{SV}_i$ from $\boldsymbol{SV}_t^{nlos}$ **if**:
    •  its elevation angle is smaller than $el_{threshold}$
    •  The HDOP of the remaining satellites (including the
       remaining satellites in $\boldsymbol{SV}_t^{nlos}$ and satellites in $\boldsymbol{SV}_t^{los}$) is
       smaller than the HDOP threshold $H_{thres}$.
Step 3: Repeat step 2 until all the conditions in step 2 cannot be fully
        satisfied.
Step 4: save the remaining satellites in $\boldsymbol{SV}_t^{nlos}$ and $\boldsymbol{SV}_t^{los}$to $\boldsymbol{SV}_t^{los\_nlos} =$
        $\{\boldsymbol{SV}_t^{nlos},\ \boldsymbol{SV}_t^{los}\}$.
```

4.3.3 GNSS Positioning Covariance Estimation

The GNSS positioning result is represented as $\boldsymbol{P}_t^{ENU} = [x_E \quad y_N \quad z_u]$ in the ENU coordinate system [64]. Conventionally, the GNSS uncertainty is usually modeled by considering the SNR, satellite numbers and HDOP if the NLOS satellites are not identified [90]. This rough modeling can only work in open-sky environments with little NLOS receptions. The team at the University of Illinois [91] proposes to model the GNSS positioning uncertainty solely based on the SNR [91] after identifying and excluding the NLOS receptions. However, fully NLOS exclusion is not acceptable in the super-urbanized area as it can result in a significant increase in HDOP. In this paper, we propose to model the covariance matrix of GNSS positioning that consisted of two parts, the NLOS and LOS, as following:

$$\boldsymbol{R}_G = \boldsymbol{R}_C + \boldsymbol{R}_E \tag{4-10}$$

The $\boldsymbol{R}_C$ is the covariance matrix indicates the uncertainty of GNSS positioning with the assumption that all the satellites used for positioning in $\boldsymbol{SV}_t^{los\_nlos}$ are line-of-sight. The $\boldsymbol{R}_C$ is calculated as follows:

$$\boldsymbol{R}_C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} HDOP_{xy}\sigma_{UERE}^2 \tag{4-11}$$

σ_{UERE} indicates the user-equivalent range error (UERE) and is experimentally determined in this paper. The $HDOP_{xy}$ is the HDOP of the GNSS positioning.

The $\boldsymbol{R}_E$ is the covariance matrix indicates the extra uncertainty of GNSS positioning caused by the NLOS satellite. The $\boldsymbol{R}_E$ is calculated as follows:

$$\boldsymbol{R}_E = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} HDOP_{xy}\sigma_{NLOS}^2 \tag{4-12}$$

σ_{NLOS} indicates the extra uncertainty caused by the NLOS receptions. According to [3], the pseudorange error for each NLOS measurement can roughly be modeled as follows:

$$\gamma = \alpha(sec\theta_{elc}(1 + cos2\theta_{elc}) + sec\theta_{azm}(1 + cos2\theta_{azm})) \tag{4-13}$$

where α represents the distance between the GNSS receiver and the reflector and is obtained from the surface detection presented in Section III-A. The θ_{elc} and θ_{azm} represents the elevation and azimuth angles, respectively. Thus, we can obtain the total uncertainty of pseudorange σ_{NLOS} for all the satellites (totally *k* satellites) as following:

$$\sigma_{NLOS} = \textstyle\sum_{i=1}^{k} \gamma_i \tag{4-14}$$

In this case, the covariance of GNSS positioning is calculated by considering both the LOS and the NLOS measurements. The component needed to be estimated in the matrix $\boldsymbol{R}_G$ is the following:

$$\overline{\boldsymbol{R}} = HDOP_{xy}\sigma_{NLOS}^2 + \ HDOP_{xy}\sigma_{UERE}^2 \tag{4-15}$$

Moreover, only the covariance in the horizontal direction is obtained. In the super-urbanized area, the vertical dilution of precision (VDOP) is significantly larger than the HDOP. The positioning error in the vertical direction can be very bad due to the distorted vertical distribution of the satellites. Thus, only the horizontal GNSS positioning and the corresponding covariance are used in the proposed GNSS/LiDAR integration.

4.4 Graph-Based GNSS/LiDAR Integration

This section presents the graph-based GNSS/LiDAR integration. Pose graph optimization is to construct all the measurements into a graph as constraints and calculate the best set of poses by solving a non-linear optimization problem. In this paper, the constraints are provided by both the object detection aided GNSS positioning and the LiDAR odometry. Two steps are needed to implement the graph-based GNSS/LiDAR integration optimization, graph generation, and graph optimization.

4.4.1 Graph Generation

The graph is constituted by edges and vertexes [86]. Edges are provided by the observation measurements including the GNSS and the LiDAR as shown in Fig. 6. The $\boldsymbol{x}_i$ represents the 6-dimension (6D) pose estimation that included the position and orientation. $\boldsymbol{e}_{ij}$ indicates the error function evaluating the difference between the estimated state and the observation from sensors. $\boldsymbol{z}_{ij}$ represents the observation. In graph optimization, the $\boldsymbol{x}_i$ is the state. The observations include three parts, the measurements from the GNSS, loop closure, and the LiDAR positioning presented in Sections II and III, respectively. The blue circles and red lines represent the nodes and the edges respectively, which are provided by the globally referenced GNSS positioning. The red circles and blue lines indicate the nodes and the edges respectively, provided by LiDAR odometry. The black line indicates the edge provided by loop closure. The error function for GNSS observation is expressed as follows:

$$\boldsymbol{e}_i^{GNSS} = ||h_i(\boldsymbol{x}_i) - \boldsymbol{z}_i^{GNSS}||_{\boldsymbol{\Omega}}^2 \tag{4-16}$$

where the $h_i(*)$ is the measurement function, relating between the GNSS measurement $\boldsymbol{z}_i^{GNSS}$ to the state $\boldsymbol{x}_i$. $\boldsymbol{\Omega}$ is the covariance matrix of the corresponding observation measurement.

LiDAR odometry can provide continuous 6D pose estimates and corresponding covariance. The error function for LiDAR odometry is expressed as:

$$\boldsymbol{e}_i^{LiDAR} = ||\boldsymbol{x}_i - \boldsymbol{x}_{i-1} - \boldsymbol{z}_i^{LiDAR}||_{\boldsymbol{\Omega}}^2 \tag{4-17}$$

where the $\boldsymbol{z}_i^{LiDAR}$ is the measurement from LiDAR odometry. Loop closure can be detected when the vehicle passes a similar or neighboring area again. The error function for loop closure is expressed as:

$$\boldsymbol{e}_{i,j}^{loop} = ||\boldsymbol{x}_i - \boldsymbol{x}_j - \boldsymbol{z}_i^{loop}||_{\boldsymbol{\Omega}}^2 \tag{4-18}$$

where the $\boldsymbol{z}_i^{loop}$ is the measurement from loop closure.

Only 2D horizontal positioning and corresponding covariance are provided by GNSS positioning in this paper. GNSS positioning error can go up to ~50 meters in a super-urbanized area. As the covariance of GNSS positioning is reasonably estimated by considering the NLOS and LOS receptions, we propose to add the GNSS results into the graph (shown in Figure 4-5) only when the $\sqrt{\overline{\boldsymbol{R}}}$ of GNSS positioning is smaller than a threshold $\boldsymbol{R}_{threshold}$. In this case, only the GNSS measurement with small covariance is applied to the graph optimization for providing the globally referenced update. This exclusion can prevent the severely biased GNSS positioning result from being applied to the graph. The detail of graph generation is shown in Algorithm 4-3.

Algorithm 4-3: Proposed Graph Generation

Input: GNSS results $\boldsymbol{z}_i^{GNSS}$ and the corresponding covariance $\overline{\boldsymbol{R}}_i$. LiDAR odometry observation $\boldsymbol{z}_i^{LiDAR}$, loop closure $\boldsymbol{z}_i^{loop}$

Output: Graph of nodes and vertexes

Step 1: Initialize the estimated state using the GNSS results.

Step 2: Add the observation $\boldsymbol{z}_i^{LiDAR}$ from LiDAR odometry into the graph if any of the following conditions are satisfied:

- The translation between the current LiDAR odometry and the previous node in the graph overweigh $Tran_{threshold}$.
- The rotation between the current LiDAR odometry and the previous node in the graph overweigh $Rot_{threshold}$.

Step 3: Add the observation of GNSS results in the graph if:

- The value $\sqrt{\overline{\boldsymbol{R}}_i}$ is smaller than $\boldsymbol{R}_{threshold}$.

Step 4: Add the observation $\boldsymbol{z}_i^{loop}$ from loop closure into the graph if loop closure is detected.

Step 5: Repeat Steps 2 and 3 until the end.

Comparing with the conventional graph-based GNSS/LiDAR integration, this paper innovatively adds the improved GNSS results and corresponding covariance into the graph optimization. The effectiveness of this novelty is subjected to the performance of the uncertainty estimation of GNSS positioning which is introduced in Section 4.4.3.

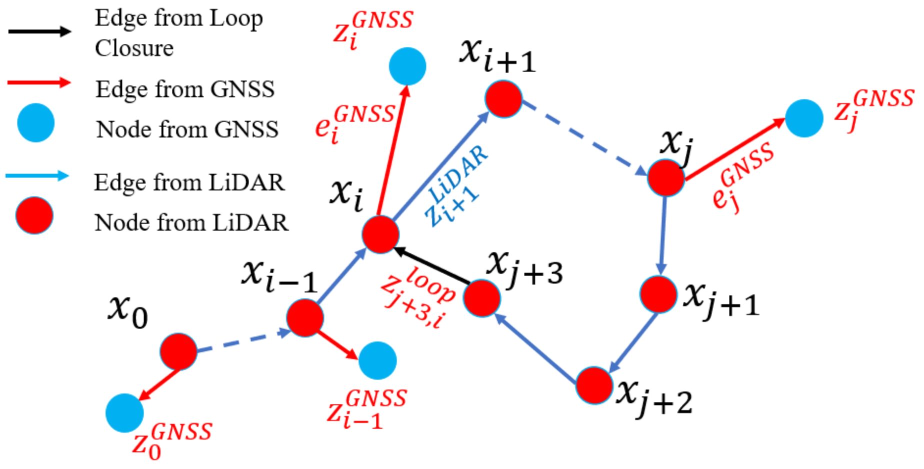

Figure 4-5 Demonstration of graph generation based on GNSS and LiDAR positioning.

4.4.2 Graph Optimization

The graph optimization [99] is straightforward that took all the constraints into a non-linear optimization problem. The optimization form is shown as following:

$$F(\mathrm{x}) = \sum_{i,j} ||h_i(\boldsymbol{x}_i) - \boldsymbol{z}_i^{GNSS}||_{\boldsymbol{\Omega}}^2 + ||\boldsymbol{x}_i - \boldsymbol{x}_{i-1} - \boldsymbol{z}_i^{LiDAR}||_{\boldsymbol{\Omega}}^2 + ||\boldsymbol{x}_i - \boldsymbol{x}_j - \boldsymbol{z}_i^{loop}||_{\boldsymbol{\Omega}}^2 \quad \text{(4-19)}$$

where $F(x)$ is the optimization function which is the sum errors of all the edges. $\boldsymbol{\Omega}_{ij}$ is the information matrix indicating the importance of each constraint in the global graph optimization. The information matrix is the inverse of the covariance matrix. The final solution to this optimization is the $\boldsymbol{x}^*$ (6D pose estimate) satisfying the following function:

$$\boldsymbol{x}^* = argminF(\boldsymbol{x}) \quad \text{(4-20)}$$

Thus, the optimization lies in solving the equation above to obtain the optimal $\boldsymbol{x}^*$. We can see from the optimization form $F(\boldsymbol{x})$, the covariance of the GNSS and LiDAR odometry positioning results are reflected in $\boldsymbol{\Omega}$. If the covariance of each positioning result is not properly estimated, the global optimization will be deflected resulting in the erroneous final pose sets.

4.5 Experimental Evaluation and Conclusions

To evaluate the performance of the proposed GNSS/LiDAR integration method in this paper, two experiments are presented in this section. The performance of LiDAR stand-alone positioning in diverse urban scenarios is extensively evaluated in previous work [68] of our research team. GNSS positioning results are presented at first. Then, the GNSS/LiDAR integration experiment results are analyzed.

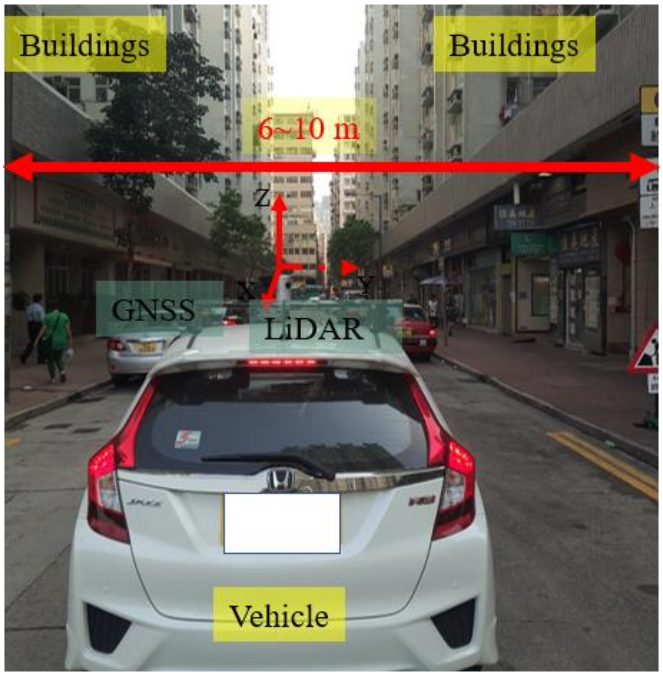

Figure 4-6 The sensors setup of the vehicle: GNSS and LiDAR sensors are installed on the top of the vehicle.

4.5.1 Experiment Setup

Two experiments are conducted in Hong Kong. The first experiment is implemented in a narrow street with buildings on both sides which can be seen in Fig. 7. Both sides of the road are filled with buildings and the distance between the buildings is just 7~10 meters (see in Fig. 7). The experiment is conducted with an open-loop route. The other experiment is conducted with a closed-loop route, and loop closure is available for the GNSS/LiDAR integration. The objective of this closed drive is to validate that the proposed GNSS/LiDAR integration solution. is repeatable in diverse scenarios and the overall performance can be well enhanced with the aid of loop closure.

In both of the experiments, the u-blox M8T receiver is used to collect raw GPS and BeiDou measurements. 3D LiDAR sensor, Velodyne 32, is employed to provide the real-time point clouds scanned from the surroundings. Both u-blox receiver and 3D LiDAR are installed on the top of a vehicle during the experiment which can be seen in Fig. 7. The data were collected within approximately 5 minutes' drive at a frequency of 1 Hz for GNSS and 10 Hz for 3D LiDAR using Robot Operation System (ROS) [66] time to synchronize all the sensor information. The sensor setup and the corresponding coordinate system are shown in Fig. 7 with the *x*-axis (LiDAR coordinate system) pointing back of the vehicle. The GNSS positioning is represented in the ENU reference system. The initial position of the experiment is employed as the initial position calculation of ENU coordinates. Moreover, LiDAR coordinate is shown in Figure 7 and are calibrated [68] with GNSS in ENU [100] coordinate at the beginning of the experiment.

Besides, the NovAtel SPAN-CPT, GNSS/INS (fiber optic gyroscopes) integrated navigation system is used to provide the ground truth trajectory with decimeter level accuracy.

TABLE 4-1
PERFORMANCE OF THE THREE GNSS POSITIONING METHODS (2D POSITIONING)

All data	**Conventional: WLS**	**WLS-NE-A (Excluding all NLOS)**	**WLS-NE-P (Partially Excluding NLOS)**
Mean Error	29.81 m	30.25 m	**27.09 m**
std	21.09 m	22.28 m	19.6 m
Availability	100%	97.45%	100%

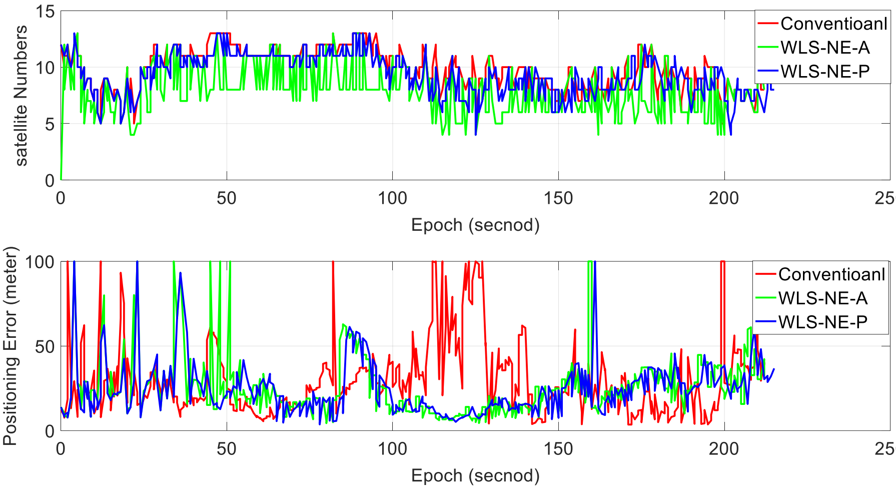

Figure 4-7. Experiment 1: Experimental results of WLS and WLS-NE, which depicted in red and blue dots, respectively. The top panel indicates the number of satellites used. The bottom panels indicate 3D positioning errors.

4.5.2 Experimental Evaluation using Data Collected by Open Loop Route

(1) GNSS Positioning Evaluation

GNSS positioning is evaluated by comparing WLS-based GNSS positioning with the GNSS positioning aided by NLOS exclusion. The results of the GNSS positioning (2 dimensions) using different methods are listed in TABLE 4-1. If all the NLOS receptions are excluded from GNSS positioning, the result is shown in the third column. The result obtained by the proposed NLOS exclusion method in Algorithm 4-2 (WLS-NE) is shown in the fourth column. Due to the blockage from the tall buildings, the majority of the measurements are NLOS

The conventional WLS method can obtain 29.81 meters of mean error. The error magnitude is much larger than the positioning error in [9] where its experiment is conducted in less urbanized areas. The standard deviation is 21.09 and the availability is 100% during the test. With the exclusion of all the NLOS measurements, the GNSS positioning is even worse. The mean of its positioning error goes up to 30.25 meters and the standard deviation also slightly increases. Moreover, the availability of this solution decreases to 97.45 %. This result shows that the exclusion of all NLOS measurements may not improve the overall performance in highly urbanized areas. This is due to the distortion of the satellite's geometric distribution, namely, larger HDOP occurs.

With the proposed method shown in Algorithm 4-2, the mean positioning error is slightly improved from 29.81 to 27.09 meters. Moreover, the availability of the GNSS solution is guaranteed (100%). The improvement is not too large because of the excessive NLOS receptions in the tested scenario.

The satellite numbers and the GNSS positioning results are shown in Figure 4-7. The green curve represents the number of the satellite when all the NLOS receptions are excluded. The blue curve indicates the satellite number based on the proposed NLOS exclusion algorithm (Algorithm 4-2). The satellite number can be decreased to less than 5 if all the NLOS receptions are excluded which can be seen in the top panel of Figure 4-7. Due to the frequent NLOS exclusion based on Algorithm 4-2, the satellite number is slightly decreased comparing to the red curve. Only part of the identified NLOS is excluded using Algorithm 4-2 can guarantee enough satellites for GNSS positioning calculation. As shown in the bottom panel of Figure 4-7, the proposed NLOS exclusion can introduce improvements sometimes instead of all the time. This is because there are too many NLOS receptions and exclusion can also enlarge the HDOP in some ways.

The result of covariance estimation based on the proposed method is shown in Figure 4-8. The black dots represent the GNSS positioning error using the WLS-NE method. This is the value that the estimated covariance expected to approach. The red dots represent the conventional GNSS covariance estimation based on the method in [36] ($\boldsymbol{R_C}$). This method cannot model the NLOS error caused by signal reflection. We can see from the Figure 4-8, this covariance estimation slightly fluctuated due to the change in HDOP. However, this covariance is far from the black dots. The blue dots represent the proposed GNSS positioning covariance estimation result based on $\sqrt{\overline{\boldsymbol{R}}}$. This covariance is closer to the black dots (refer to the ground truth of covariance) comparing with that of the conventional covariance. As shown in Figure 4-8, the proposed covariance can effectively model the GNSS positioning error in some epochs, especially when then GNSS positioning error is smaller. However, some epochs encounter a large difference between the estimated covariance and actual positioning error. This is because the proposed method can only identify the NLOS near the vehicle which is within the detection range of the 3D LiDAR (commonly 120 meters). The NLOS that its reflector too far away from the vehicle is not be modeled by the proposed covariance estimation method. Moreover, the multipath effects are also not modeled in this covariance estimation method.

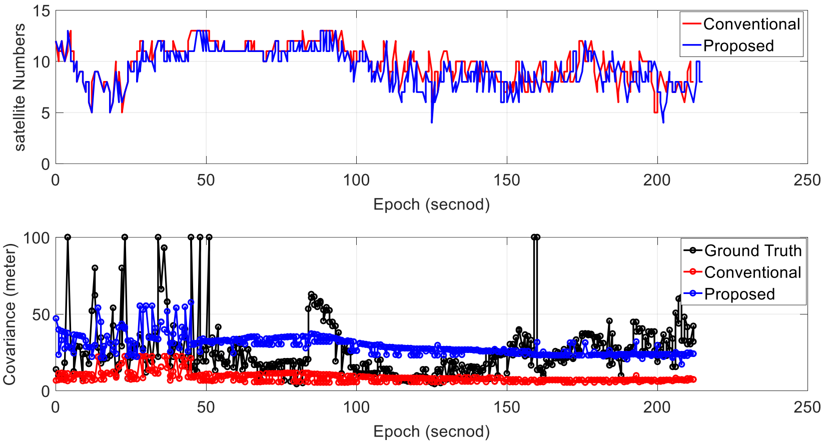

Figure 4-8 Experiment 1: Top panel indicates the numbers of satellites used in the conventional and proposed GNSS positioning methods. The bottom panel indicates the corresponding covariance estimated. The conventional and proposed covariance estimation is indicated in red and blue dots, respectively. GNSS positioning error using the WLS-NE is represented in black dots (ground truth for covariance).

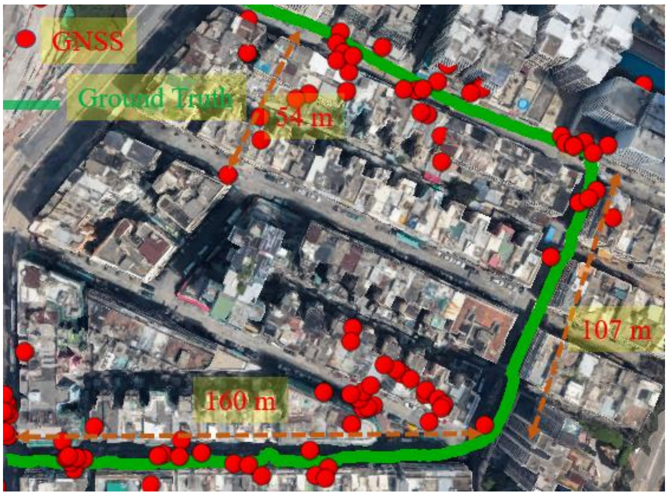

Figure 4-9. Experiment 1: Trajectory of the autonomous vehicle is indicated by the green curve. The red circles indicate the GNSS positioning result.

(2) GNSS/LiDAR Integration Evaluation

The trajectory of the tested vehicle is shown in Figure 4-9. The red circles represent the GNSS positioning results using the proposed WLS-NE method. The green curve indicates the ground truth of the tested trajectory. We can see from the figure that the majority of the epochs possess large positioning errors. In the GNSS/LiDAR integration, GNSS is the only source that can provide absolute positioning information. The graph generation in the SLAM is shown in Figure 4-10. Figure 4-10 indicates the organized point cloud, nodes, and edges for further graph-based optimization.

As we can see from Figure 4-7, the WLS-NE based GNSS positioning solution can still even reach 54 meters. In this section, three GNSS/LiDAR integration methods are compared.

- Method (a): GNSS/LiDAR integration with conventional GNSS covariance estimation [36].
- Method (b): GNSS/LiDAR integration with proposed GNSS covariance estimation.
- Method (c) GNSS/LiDAR integration with proposed GNSS covariance estimation. However, GNSS positioning is integrated into graph optimization only when $\sqrt{\boldsymbol{R}}$ is smaller than the threshold$R_{threshold}$.

The GNSS/LiDAR integration results are given in Table 4-2 using the three methods. The mean error of the conventional GNSS/LiDAR integration is 24.07 meters and is improved comparing with the performance of the GNSS standalone (27.09 meters). With the aid of proposed GNSS positioning covariance (Method (b)), the error of GNSS/LiDAR integration is slightly decreased to 22.67 meters. The standard deviation is also slightly decreased. In the integration method (b), all the GNSS positioning results and corresponding covariance are applied in the GNSS/LIDAR integration. As the majority of the GNSS positioning is erroneous, it is reasonable to use GNSS results when it is accurate. The accurate results can be identified when its estimated covariance is less than $R_{threshold}$. Dramatic improvement is obtained after the constraint of covariance is applied. The mean error and standard deviation are decreased to 12.67 and 6.57 meters, respectively. Moreover, the availabilities of all three methods are 100%. This improvement shows that the proposed covariance estimation can improve the performance of the GNSS/LiDAR integration. The GNSS/LiDAR integration results are shown in Figure 4-11. Comparing to the Bayes filter-based [73, 74] sensor fusion method, the graph-based GNSS/LiDAR integration takes all the constraints into the optimization framework. Thus, the poses of the whole organized point clouds, nodes, edges changed over time. We can see from Figure 4-11 and conclude that:

(1) The proposed method (c) obtained the most accurate trajectory over the three methods.

(2) The positioning error decreased near the end of the drive, meaning that the GNSS/LiDAR integration can mitigate the drift of LiDAR odometry.

The error of GNSS/LiDAR integration is shown in Figure 4-12. We can see from the figure that the method (c) outperforms the other two methods over the majority of the epochs.

The previous research [53, 90] tends to integrate the GNSS and LiDAR in the scenario where the GNSS positioning error is less than 5~8 meters using GNSS WLS. In this tested scenario, the GNSS results with an enormous error are applied in the GNSS/LiDAR integration which is very common in super-urbanized cities such as Hong Kong

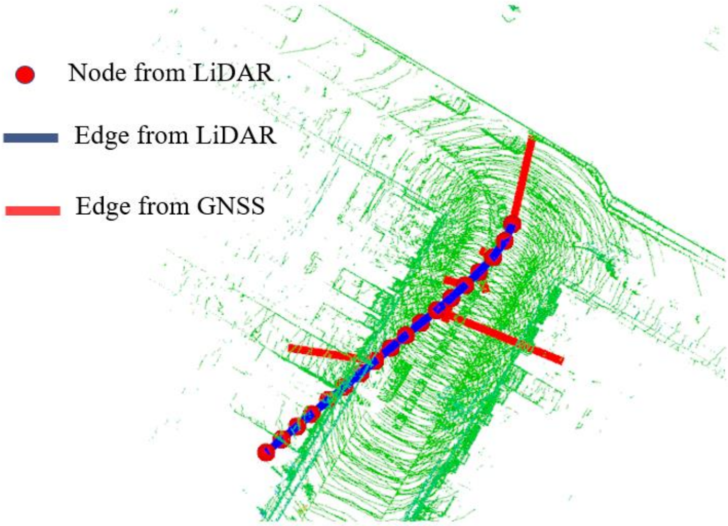

Figure 4-10. Graph generation in the real graph-slam process.

TABLE 4-2
EXPERIMENT 1: PERFORMANCE OF THE THREE GNSS/LIDAR INTEGRATION METHOD

All data	Method (a)	Method (b)	Method (c)
Mean Error	24.07 m	22.67 m	12.67 m
STD	14.69 m	14.48 m	6.57 m
Availability	100%	100%	100%

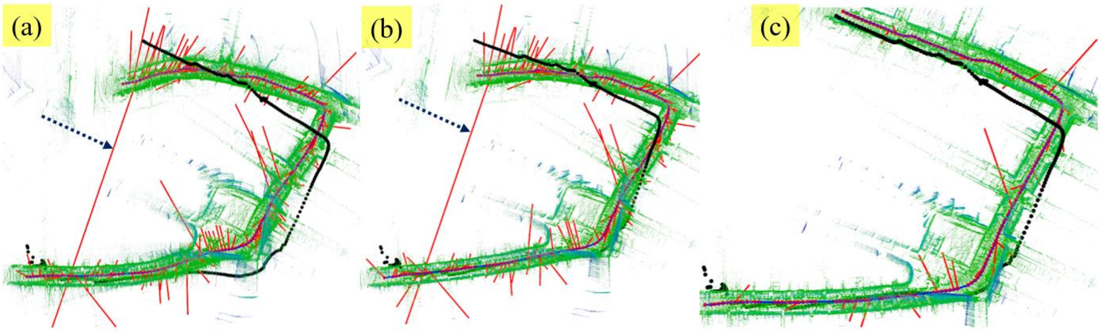

Figure 4-11. Experiment 1: Results of the GNSS/LiDAR integration based on three integration methods. The blue curve is constituted by the optimized nodes (refer to the red node in Figure 4-5. The black curve indicates the ground truth of the trajectory provided by NovAtel SPAN-CPT. The red line represents the edge from GNSS positioning: refer to the red line in Figure 4-5). The green points represent the organized point clouds.

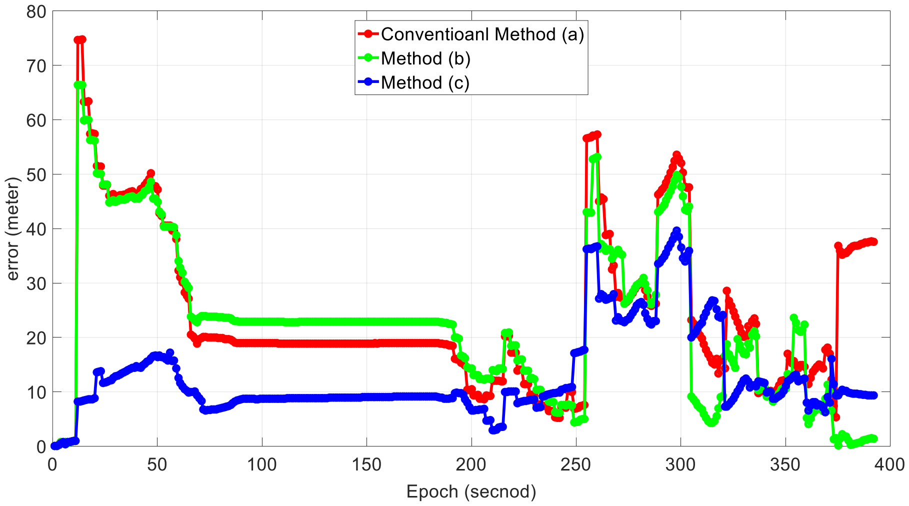

Figure 4-12. Experiment 1: Positioning error of the GNSS/LiDAR integration results based on the three methods. Red, green and blue curves indicate the GNSS/LiDAR integration methods (a), (b) and (c), respectively.

4.5.3 Experimental Evaluation using Data Collected by Closed Loop Route

(1) GNSS Positioning Evaluation

This experiment is conducted in a super-urbanized area with fewer satellites visible comparing with that of the first experiment. Moreover, this experiment route is closed-loop. The loop closure [97] detection is employed in the GNSS/LiDAR integration process in this experiment. The experiment scene is shown in Figure 4-13. The height of the building is about 30 meters. The distance between the buildings is just about 8 meters. We can see from the figure that the majority of the GNSS positioning results lie in the buildings due to the multipath effects and the excessive NLOS receptions. Again, the mean error is slightly reduced from 46.62 (conventional WLS) meters to 43.12 meters (proposed WLS-NE).

The covariance estimation result of the proposed method is shown in Figure 4-14. Compared with the covariance estimation in the first experiment shown in Figure 4-8, the covariance is better estimated in this experiment. The main reason is that the majority of the satellites are NLOS due to the tall building which means fewer multipath effects. As discussed earlier, the multipath is not modeled in the proposed covariance estimation method. In other words, the proposed GNSS positioning covariance estimation method can obtain better performance in narrower streets.

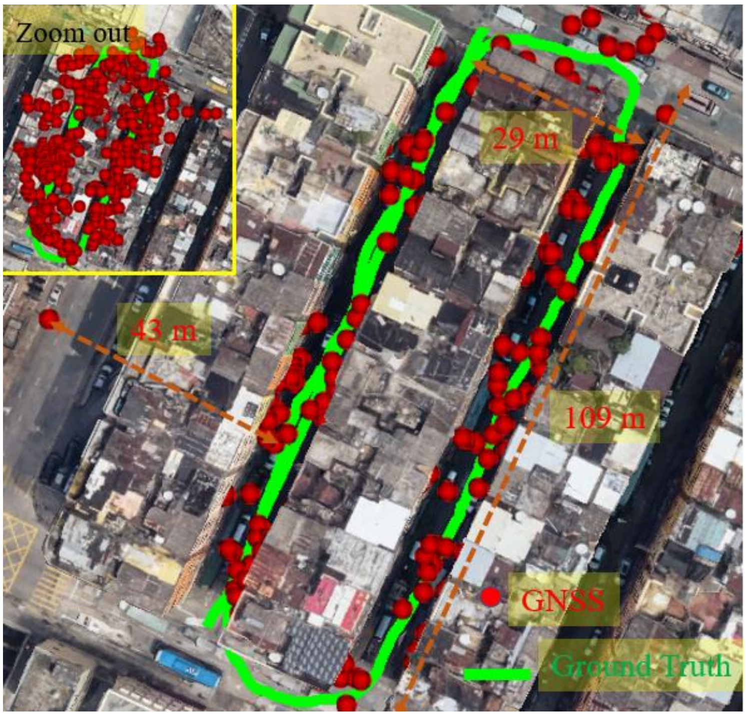

Figure 4-13. Experiment 2: Trajectory of the vehicle is indicated by the green curve. The red circles indicate the object detection aided GNSS positioning results.

(2) GNSS/LiDAR Integration Evaluation

GNSS/LiDAR integration performance is shown in TABLE III. The loop closure detection is applied in this experiment as the driving route is a closed-loop shown in Figure 4-13.

The conventional solution obtains a mean positioning error of 25.68 meters with a standard deviation of 28.09 meters. With the assistant of the proposed covariance estimation, the mean positioning error drastically decreased to 8.14 meters. The mean positioning error is reduced to 7.49 meters with the covariance magnitude constraint being applied. Moreover, the standard deviation is also decreased to 5.43 meters.

The final optimized nodes and organized point clouds are shown in Figure 4-15. The positioning error during the test is shown in Figure 4-16. We can see from Figure 4-15, the edges of GNSS are dramatically decreased with the constraint of covariance.

Interestingly, the positioning error of all methods decreased after epochs 120. The reason is the detection of the loop closure, which is a strong constraint for further graph-based optimization. Regarding the performance of conventional GNSS/LiDAR integration with no loop closure in the first experiment, the positioning error can still reach about 40 meters at the end of the test.

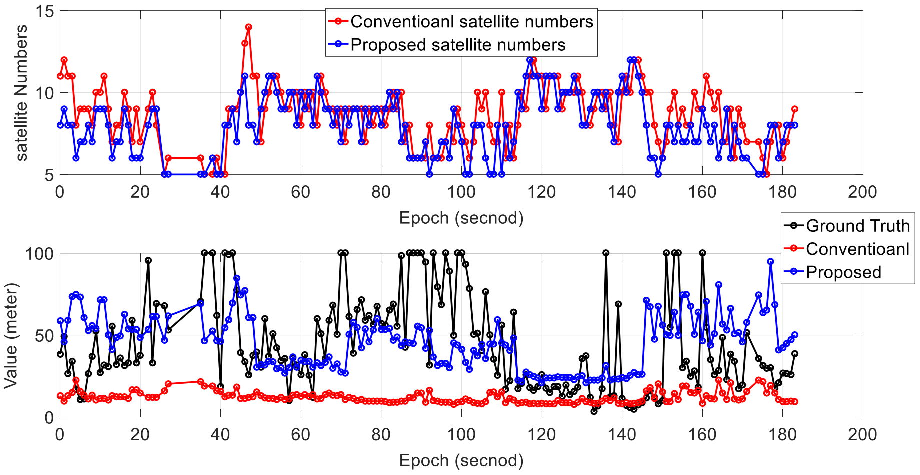

Figure 4-14. Experiment 2: Top panel indicates the numbers of satellites used in the conventional and proposed GNSS positioning methods. The bottom panel indicates the corresponding covariance estimated. The conventional and proposed covariance estimation is indicated in red and blue dots, respectively. GNSS positioning error using the WLS-NE is represented in black dots (ground truth for covariance).

TABLE 4-3
EXPERIMENT 2: PERFORMANCE OF THE THREE GNSS/LIDAR INTEGRATION METHODS

All data	Method (a) Conventional GNSS/LiDAR Integration	Method (b) Proposed GNSS/LiDAR Integration	Method (c) Proposed GNSS/LiDAR Integration
Mean Error	25.68 m	8.14 m	7.49 m
STD	28.09 m	6.73 m	5.43 m
Availability	100%	100%	100%

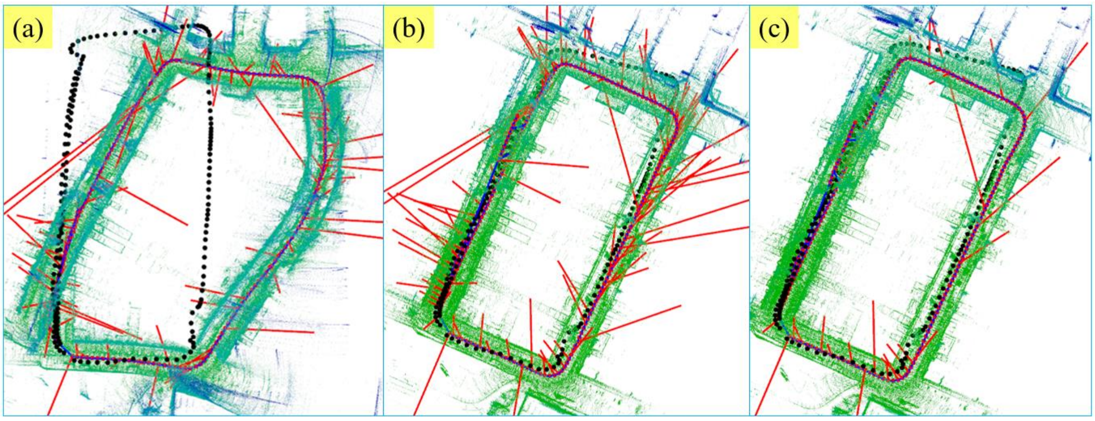

Figure 4-15. Experiment 2: Results of the GNSS/LiDAR integration based on three integration methods. The blue curve is constituted by the optimized nodes (refer to the red node in Fig. 6. The black curve indicates the ground truth of the trajectory provided by NovAtel SPAN-CPT. The red line represents the edge from GNSS positioning: refer to the red line in Fig. 6). The green points represent the organized point clouds.

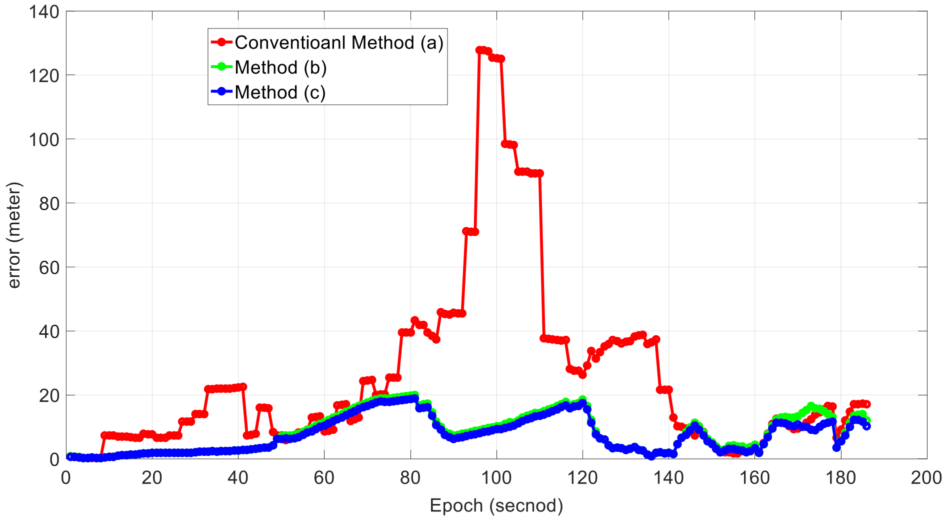

Figure 4-16. Experiment 2: Positioning error of the GNSS/LiDAR integration results based on the three methods. Red, green and blue curves indicate the GNSS/LiDAR integration methods (a), (b) and (c), respectively.

The proposed GNSS/LiDAR integration method obtained improved performance with the aid of the NLOS exclusion (empowered by LiDAR-based object detection) and the proposed covariance estimation. The proposed NLOS exclusion can obtain improvements when more satellites are available. Though the positioning performance of GNSS is very unsatisfactory during both of the experiments. GNSS is still indispensable for providing the globally referenced positioning.

The proposed covariance estimation can capture the majority of GNSS positioning errors. However, the GNSS positioning error caused by the multipath effect cannot be modeled using the proposed covariance model. In the first experiment, the mean GNSS positioning error is less than 30 meters which is better than the second experiment. This is because the buildings in the second experiment are even taller which introduces more NLOS receptions consequently. As presented in Algorithm 2, only the NLOS is modeled in the covariance. Thus, the second experiment obtains better performance regarding the GNSS covariance estimation which can be seen by comparing the Figs. 9 and 15. As the multipath effects are random and difficult to model. Thus, effectively modeling of multipath is a promising work to yield.

The proposed constraint of covariance applied to the GNSS/LiDAR integration can improve performance. This novel constraint guarantees that only the accurate GNSS positioning will be applied to the integration. In other words, the effectively estimated covariance can identify the erroneous GNSS results.

Overall, the proposed GNSS covariance estimation can improve the GNSS/LiDAR integration performance. The globally referenced positioning is obtained. This result proves that the covariance estimation is significant for the GNSS/LiDAR integration. However, the integrated positioning result is still large with the best performance of even 7.49 meters of mean error in the second experiment.

To realize autonomous vehicles, this kind of scenario is still a challenge for GNSS positioning. Even the real-time kinematic (RTK) GNSS can suffer from severe NLOS and multipath effects. Direct NLOS exclusion will result in the big distortion of the satellite distribution, namely the HDOP. Thus, effectively modeling the covariance of GNSS positioning is a potential solution to improve the robustness of the GNSS/LiDAR integration in super-urbanized areas.

4.5.4 Conclusions

With the fast development of the autonomous vehicles, GNSS and LiDAR became the indispensable sensors to provide sensing and localization functions. The environment feature can be used to improve GNSS positioning performance in urbanized areas with excessive tall buildings. To the best of the author’s knowledge, this paper is the first attempt to employ the LiDAR-based object detection to improve the GNSS.

This paper innovatively employs the LiDAR perception to detect building surface to facilitate the covariance modeling of GNSS positioning for the GNSS/LiDAR integration. This study firstly employs the LiDAR to provide the LiDAR odometry based on the state-of-art NDT and the corresponding covariance is estimated. Then, the building surfaced is detected and identified using the object detection followed by the NLOS detection and novel NLOS exclusion. Thirdly, the GNSS positioning is implemented using the surviving range measurements. The GNSS positioning covariance is proposed based on an NLOS model. Fourthly, the LiDAR odometry and the GNSS positioning is integrated by a graph-based SLAM framework. Finally, the experiment is conducted to validate the propose GNSS/LiDAR integration framework. The results show that the proposed method of GNSS positioning covariance estimation can model the majority of the positioning error caused by NLOS reception. The performance of the proposed GNSS/LiDAR integration with adaptive covariance outperforms the conventional GNSS/LiDAR integration with the constant covariance.

Furthermore, the remaining GNSS positioning error caused by the multipath effects will be studied and modeled to improve the performance of GNSS positioning covariance estimation. Moreover, the real-time kinematic (RTK) GNSS will be applied to integrate with LiDAR to verify how much the proposed method can help the RTK GNSS/LiDAR integration.

5. SLIDING WINDOW MAP AIDED GNSS AND ITS INTEGRATION WITH INS

5.1 Introduction

Positioning in urban environments is becoming essential due to the increasing demand for autonomous driving vehicles (ADV) [101]. To achieve Level 4 [52] autonomous driving capability in all scenarios, centimeter-level absolute positioning is required. The global navigation satellite system (GNSS) [100] is currently one of the principal means of providing globally-referenced positioning for autonomous driving vehicle localization. With the increased availability of multiple satellite constellations, GNSS can provide satisfactory performance in open-sky areas, with a positioning error of around five meters [3]. However, positioning error can be up to 50 meters in highly-urbanized cities such as Hong Kong [3], due to signal reflection caused by static buildings and dynamic objects [67] such as double-decker buses. If the direct light-of-sight (LOS) is blocked, and reflected signals from the same satellite are received, the notorious non-light-of-sight (NLOS) receptions occur. According to a recent review paper [7], NLOS is currently the major difficulty in the use of GNSS in intelligent transportation systems. Because of NLOS receptions, the performance of GNSS positioning is highly influenced by real-time surrounding environmental features, such as buildings and dynamic objects. Effectively sensing and understanding surrounding environments is the key to improving GNSS positioning in urban areas, as GNSS positioning relies heavily on sky view visibility. Research into mitigating the effects of NLOS receptions in urban canyons can be divided into four areas: (1) 3D mapping-aided GNSS; (2) robust model-aided GNSS positioning; (3) camera-aided GNSS positioning; and (4) 3D LiDAR-aided GNSS positioning.

3D mapping-aided (3DMA) GNSS: Utilizing 3D building models to detect NLOS is straightforward. NLOS satellites can be detected with the aid of 3D models of buildings, and can then be excluded from use in GNSS positioning [8, 9]. However, NLOS exclusion will distort the perceived geometric distribution of the satellites, degrading accuracy [10], and even resulting in too few satellites for further GNSS calculation [10]. Moreover, these methods require the availability of 3D building models of the environment, and the performance of NLOS detection relies on the accuracy of an initial guess of the GNSS receiver’s position. Besides, NLOS reception caused by dynamic objects cannot be detected as well. A well-known method, GNSS shadow matching, was developed to match measured satellite visibility with the predicted satellite visibility of hypothesized positions [11-13]. However, the performance of shadow matching is dependent upon the quality of satellite visibility classification and the initial guess as to the position of the GNSS receiver. A likelihood-based 3DMA GNSS method, which modeled the measurement uncertainty, and used this value to mitigate NLOS effects, has been proposed to provide accurate positioning in the along-street direction [14]. Due to the complementarity of the shadow matching and likelihood-based 3DMA GNSS, approaches to the integration of these approaches have recently been studied [15]. Another range-based 3DMA GNSS method is to correct the NLOS affected measurements for GNSS positioning [16-19]. These methods were proposed to simulate signal transmission routes, using a ray-tracing method. However, these ray-tracing-based 3DMA GNSS methods have the drawbacks of requiring stringency in 1) the accuracy of 3D mapping databases; 2) the initial guess of receiver positions, and 3) the computational power of the processors due to the ray-tracing process. Recently, a skymask-based NLOS correction method has been proposed [20], which effectively reduces the computational load incurred by ray-tracing-based methods [16-19]. However, these 3D mapping-aided GNSS have the drawbacks of 1) reliance on the availability of 3D building models; 2) inability to mitigate NLOS receptions caused by surrounding dynamic objects.

Robust model-aided GNSS positioning: Instead of using additional information from 3D building models, a team from the Chemnitz University of Technology employed a robust model [102-104] to mitigate the effects of NLOS signals in GNSS single point positioning (SPP). In their earliest work, as reported in [105], a state-of-the-art factor graph [106], which makes use of all of the available historical GNSS measurements to estimate the state set of the GNSS receiver, was applied to GNSS positioning. The improved performance was obtained compared with the conventional weighted least squares (WLS) approach. This improvement was primarily because the factor graph also explores the correlation between consecutive epochs of GNSS measurements. Interestingly, the work described in [107] included a switchable constraint in the factor graph to model the probability of one satellite being an unhealthy measurement, either multipath or NLOS receptions. According to [107], the switchable constraint can effectively mitigate the effects caused by NLOS and multipath, so-called outliers in [107]. However, a major drawback is that the switchable constraint relies heavily on the initial guess of the prior factor [107], which requires a large number of hyperparameters, making it difficult to tune. Inspired by a covariance estimation technique used in the field of robotics [108], dynamic covariance estimation (DCE) [104] has been studied as a means by which to adaptively model the uncertainty of GNSS measurements. In this research, the covariance of GNSS measurements is treated as an unknown variable to be estimated in the factor graph optimization (FGO). In other words, the position of the GNSS receiver and the uncertainty in the measurement of GNSS are estimated simultaneously. Finally, the NLOS satellites are de-weighted from the FGO. Significantly improved results were obtained when the number of healthy measurements was more than the number of unhealthy measurements, such as NLOS receptions. However, it is difficult to satisfy this assumption in dense urban areas, and the DCE requires numerous parameterization as well. To get rid of the excessive parameterization, a Gaussian mixture model (GMM) [102, 103] has been proposed for modelling the uncertainty of GNSS pseudorange measurements. According to the evaluation reported in [103], the noise model of GNSS measurements is not subject to the Gaussian assumption due to the long tail of the distribution, which is caused by NLOS receptions [103]. Therefore, a GMM was employed to model the noise of the GNSS measurements, and an expectation-maximization (EM) algorithm was applied to estimate the parameters of the GNSS noise model. Then, the GMM was applied to the FGO. This is the first work to date to make use of GMM to model the uncertainty of GNSS measurements in factor graph optimization. Better and more robust performance can be obtained using this approach than can be achieved by methods using Gaussian noise, switchable constraints, or DCE. However, a major drawback of the GMM-based method is that the estimation of the parameter values of the GMM relies heavily on the accuracy of the initial guess of the position of the GNSS receiver, which is in fact similar to range-based 3DMA GNSS [16-19]. The sensitivity of the parameter estimation of GMM with respect to the initial position guess is still an open question. A team from West Virginia University (WVU) carried out similar research [109-111], applying robust models to GNSS precise point positioning (PPP) and obtained improved results.

Camera-aided GNSS positioning: Another research approach makes use of a camera to capture the environment surrounding the GNSS receiver. This approach is called camera-aided GNSS positioning [60, 112]. The camera was used to capture a sky view and establish satellite visibility. To detect the visibility of satellites, the researchers applied omnidirectional or fisheye cameras [38, 39, 60]. NLOS receptions could be detected in conjunction with the captured sky views, and improvements obtained after excluding all of the NLOS receptions detected from GNSS positioning. A major advantage of this approach is that camera-based NLOS detection relaxes the requirement of the initial guess of the position of the GNSS receiver, compared with the 3D building models-based method [8, 9]. NLOS receptions caused by dynamic objects can also be detected. Similar research [113, 114] has been conducted recently, in which improved GNSS positioning was integrated with visual simultaneous localization and mapping (VSLAM) [115]. Unfortunately, these methods still tend to exclude NLOS receptions from GNSS positioning, and so are not applicable in dense urban areas, such as Hong Kong, Tokyo, and New York. Instead of excluding all NLOS satellites from the GNSS positioning, our recent research, as reported in [116], showed the feasibility of remodeling the NLOS satellites after detecting NLOS receptions using a fisheye camera. However, the camera is sensitive to the conditions of illumination. Moreover, the classification of satellite visibility relies heavily on the quality of the sky view, and on non-sky view area segmentation [116]. The installation of sky-pointing cameras for autonomous driving vehicles is not always possible.

3D LiDAR aided (3DLA) GNSS positioning: Recently, 3D LiDAR sensors, the so-called "eyes" of ADV, the typical indispensable onboard sensor for autonomous driving vehicles, have been used to detect NLOS caused by dynamic objects [67]. The typical dynamic object, the Hong Kong double-decker bus, was detected based on real-time 3D point clouds to identify NLOS signals blocked by the bus, but reflected by surrounding buildings. A static experiment was first performed to demonstrate the phenomenon of NLOS receptions caused by a double-decker bus. The dynamic experiment was conducted to investigate performance improvements produced by detecting and excluding NLOS receptions caused by dynamic objects from use in further GNSS positioning. According to the experiments, more than five meters of positioning error can be introduced by the detected NLOS. In short, the NLOS receptions can be caused by dynamic objects which should also be solved. However, the method proposed in [67] relies heavily on the accuracy of object detection. Due to the limited field of view (FOV: $-30^{\circ}\sim+10^{\circ}$) of 3D LiDAR, only part of a double-decker bus can be scanned. However, to the best of our knowledge, this is the first work that employed object detection to help GNSS positioning. Instead of detecting only dynamic objects, we also explored the detection of surrounding static buildings using 3D LiDAR point clouds [10]. Due to the limited field of view of 3D LiDAR, only part of the buildings can be scanned. Therefore, information about building height is required [10]. Given this information, NLOS receptions caused by the buildings can be detected. Instead of excluding the detected NLOS receptions, we explored ways in which to correct NLOS pseudorange measurements with the help of LiDAR, which can measure the distance from the GNSS receiver to the surface of a building which may have reflected the GNSS signal. Then the corrected and the healthy GNSS measurements can both be used in further GNSS positioning. The improved performance was obtained after correcting the detected NLOS satellites. Unlike 3DMA GNSS, the method proposed in [10] reduces the importance of the initial guess of the position of the GNSS receiver. Unfortunately, the performance of this approach [10] relies on the accuracy of the detection of buildings and reflectors. Both building detection and reflector detection can fail when a building surface is irregular. The limited FOV of LiDAR remains a drawback in the detection of both dynamic objects and buildings. Overall, the work reported in [67] and [10] shows the feasibility of detecting GNSS NLOS using real-time onboard sensing: the real-time point clouds. To overcome the drawback of the limited FOV of 3D LiDAR, we explored the use of both fish-eye cameras and 3D LiDAR to detect and correct NLOS signals [117]. The fish-eye camera was applied to detect NLOS signals and the 3D LiDAR was employed to measure the distance between the GNSS receiver and a potential reflector causing NLOS signals. However, this approach shares the problems with the work described in [60, 112].

In short, state-of-the-art 3DMA GNSS relies on an initial guess of the position of the GNSS receiver, and the availability of 3D building models. The GMM-based method obtains the best performance of all the robust model-based solutions discussed. However, it relies heavily on an initial guess of the position of the receiver. The camera aided GNSS positioning has effectively relaxed the requirement for the initial guess. However, problems caused by the sensitivity of the camera to the illumination level are still an unsolved problem. 3D LiDAR aided GNSS positioning is an innovative solution for mitigating the effects of NLOS receptions and has several advantages: (1) both dynamic and static objects can be considered during NLOS detection; (2) NLOS detection does not rely on an initial guess of the position of the GNSS receiver; (3) the approach does not require the use of 3D building models, and 3D LiDAR is robust against illumination conditions. However, there are still three major drawbacks: (1) the limited FOV of 3D LiDAR causes limited environment sensing capability; (2) the performance of NLOS detection relies heavily on the accuracy of object detection, such as double-decker bus detection in [67] and building detection in [10]; and (3) the reflector detection method described in [67] [10] [117] can only work when a building surface is detected and is sufficiently regular.

In Section 5, we relaxed the three above-listed drawbacks of the previous 3D LiDAR aided (3DLA) GNSS positioning. First, a novel sliding window map (SWM) was generated based on real-time 3D point clouds from 3D LiDAR. Only the 3D point clouds inside a sliding window were employed to generate the SWM, as the point clouds far away from the GNSS receiver are not needed for NLOS detection. The environment description capability of SWM is significantly better than that of 3D real-time point clouds. Therefore, the FOV of LiDAR sensing is effectively enhanced (relax the drawback 1). Moreover, the magnitude of the drift of the SWM is bounded to a small value. As the generated SWM is in the body frame, which is located at the center of the 3D LiDAR, the orientation is directly adopted from an attitude and heading reference system (AHRS) to transform the SWM from the body frame to a local (ENU) frame [100]. Then, NLOS receptions are directly detected based on real-time SWM, which does not require object detection, using a fast searching method (relax the drawback 2). More importantly, the proposed NLOS detection method does not rely on the initial guess of the GNSS receiver. Instead of directly excluding detected NLOS satellites from use in further positioning estimation, this work proposes an approach to rectify the pseudorange measurement model by (1) correcting the pseudorange measurements if the reflecting point of the NLOS signals is detected based on a constrained searching method (relax the drawback 3) inside the dense SWM; and (2) re-modeling the uncertainty of NLOS pseudorange measurement using a novel weighting scheme if the reflector is not detected. Finally, both the corrected and healthy pseudorange measurements are tightly coupled with an inertial navigation system (INS) using factor graph optimization to see the potential of the proposed 3D LiDAR aided GNSS in sensor integration. The main contributions of this paper are as follows:

(1) This method is the continuous work in [10, 67, 117, 118], and the three listed drawbacks of the previous 3D LiDAR aided GNSS are relaxed in this paper.

(2) This approach effectively makes use of all the historical measurements to integrate with the onboard INS, using a state-of-the-art factor graph to obtain more robust positioning performance.

The remainder of Section 5 is organized as follows. An overview of the proposed method is given in Section 5.2. The generation of the sliding window map is elaborated in Section 5.3. In Section 5.4, the proposed NLOS detection, NLOS correction, and remodeling approaches are presented. Section 5.5 presents GNSS/INS integration using FGO based on the rectified and the healthy GNSS measurements. Two experiments were performed to evaluate the effectiveness of the proposed method in GNSS standalone positioning and GNSS/INS integration. Finally, conclusions are drawn, and further work is presented in Section 5.7.

5.2 Overview of the Proposed Method

An overview of the method proposed in this paper is shown in Figure 5-1. The system consists of two parts: the 3D LiDAR aided GNSS part, and the GNSS/INS integration part. The inputs of the system include (1) raw data ($\mathbf{A}_t$) from the accelerometer and orientation from AHRS ($\mathbf{R}_t$) at a given epoch *t*; (2) 3D real-time point clouds ($\mathbf{S}_t$) from a 3D LiDAR sensor; and (3) raw GNSS pseudorange measurements ($\mathbf{SV}_t^{all}$) from a GNSS receiver. The output of the system is the state of the GNSS receiver. First, the 3D point clouds within a sliding window are employed to generate the SWM ($\mathbf{M}_t$), which is described in Section 5-3. As the LiDAR point cloud is originally fixed at the body frame, the center of the 3D LiDAR, the orientation data from the AHRS are used to transform the point clouds to the local (ENU) frame [100]. NLOS detection is next performed based on the raw GNSS measurements, $\mathbf{SV}_t^{all}$, and the SWM, using the fast searching method described in Section 5-4. The output of the NLOS detection step is an estimation of satellite visibility ($\mathbf{Vis}_t$). Reflecting point detection is next conducted to find the potential point which reflects the NLOS signal. The NLOS pseudorange ($\mathbf{SV}_t^C$) is corrected based on an NLOS error model. If the reflecting point is not found inside the SWM, a novel remodeling scheme is applied to de-weight the NLOS measurements ($\mathbf{SV}_t^M$) in further GNSS positioning and its integration with INS. Finally, both the healthy ($\mathbf{SV}_t^H$) and rectified GNSS pseudorange ($\mathbf{SV}_t^C$, $\mathbf{SV}_t^M$) measurements are tightly integrated with the INS using a probabilistic factor graph optimization, as described in Section 5-5.

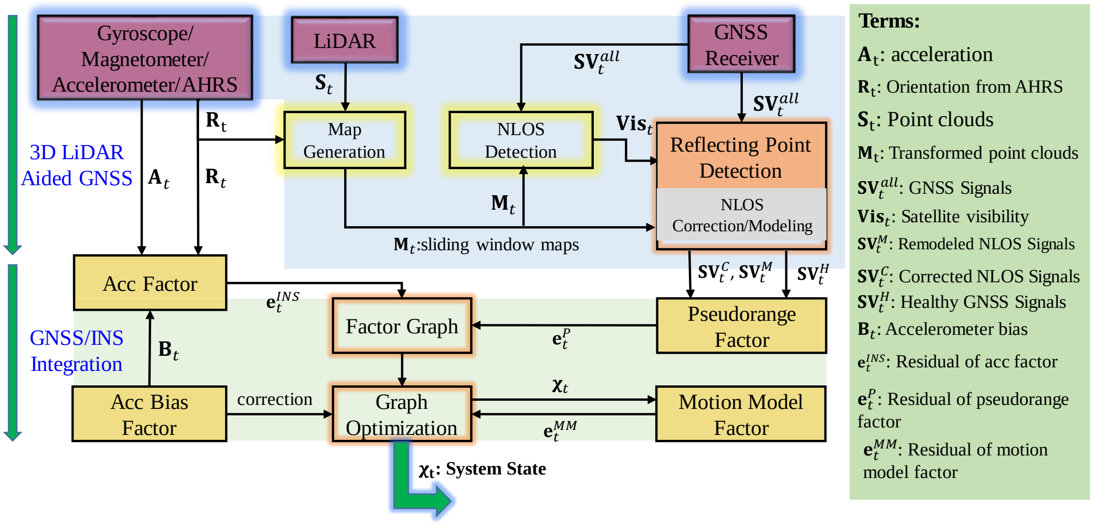
Gyroscope/ Magnetometer/ Accelerometer/AHRS
LiDAR
GNSS Receiver
$\mathbf{SV}_t^{all}$
$\mathbf{S}_t$
$\mathbf{R}_t$
Map Generation
NLOS Detection
$\mathbf{Vis}_t$
$\mathbf{SV}_t^{all}$
Reflecting Point Detection
NLOS Correction/Modeling
3D LiDAR Aided GNSS
$\mathbf{A}_t$
$\mathbf{R}_t$
$\mathbf{M}_t$
$\mathbf{M}_t$:sliding window maps
$\mathbf{SV}_t^C, \mathbf{SV}_t^M$
$\mathbf{SV}_t^H$
Acc Factor
$\mathbf{e}_t^{INS}$
Factor Graph
$\mathbf{e}_t^P$
Pseudorange Factor
GNSS/INS Integration
$\mathbf{B}_t$
Acc Bias Factor
correction
Graph Optimization
χ_t
$\mathbf{e}_t^{MM}$
Motion Model Factor
χ_t: System State
Terms:
$\mathbf{A}_t$: acceleration
$\mathbf{R}_t$: Orientation from AHRS
$\mathbf{S}_t$: Point clouds
$\mathbf{M}_t$: Transformed point clouds
$\mathbf{SV}_t^{all}$: GNSS Signals
$\mathbf{Vis}_t$: Satellite visibility
$\mathbf{SV}_t^M$: Remodeled NLOS Signals
$\mathbf{SV}_t^C$: Corrected NLOS Signals
$\mathbf{SV}_t^H$: Healthy GNSS Signals
$\mathbf{B}_t$: Accelerometer bias
$\mathbf{e}_t^{INS}$: Residual of acc factor
$\mathbf{e}_t^P$: Residual of pseudorange factor
$\mathbf{e}_t^{MM}$: Residual of motion model factor

Figure 5-1 Overview of the proposed method. The inputs are the raw measurements from INS, 3D LiDAR and GNSS receiver. The output is the position state of the GNSS receiver

5.3 Sliding Window Map Generation

This section describes in detail the methodology of generating an SWM for further NLOS detection. In our previous work, described in [10, 67, 117, 118], only real-time 3D point clouds were applied in object detection to further detect the NLOS satellites caused by buildings and dynamic objects. Due to the limited FOV of 3D LiDAR, building height information is required to extend the detected building to its exact height, as described in [10]. To solve this problem, we registered real-time 3D point clouds into a map that can effectively enhance the FOV of 3D LiDAR sensing. Figure 5-2 shows the difference between the real-time 3D LiDAR point clouds and the SWM. The white points in this Figure represent the real-time point clouds from the 3D LiDAR. The colored points denote the map points of the SWM. Note that the ground points were removed from the SWM for efficient NLOS detection.

It is clear from the Figure that only the low-lying parts of buildings or double-decker buses are scanned by the 3D LiDAR (we used Velodyne 32 [119] for the work described in this paper). The visibility of satellites with high elevation angles cannot be effectively classified simply based on real-time 3D point clouds. Real-time 3D point clouds are also sparse, due to the physical scanning angle distribution of 3D LiDAR. However, the SWM proposed in this paper can effectively ameliorate this problem. We can see from Figure 5-2 that the elevation mask angle can reach 76° with the help of

SWM, so the visibility of a satellite with an elevation angle of less than 76° can be classified in this case. The point clouds in SWM are significantly denser than raw real-time 3D point clouds, a factor that can contribute significantly to the accuracy of NLOS detection. A snapshot of the complete SWM map is shown at the top right of Figure 5-2. Both the buildings and the dynamic objects, such as double-decker buses, and even the trees are involved in the SWM, which is not included in the 3D building model [120].

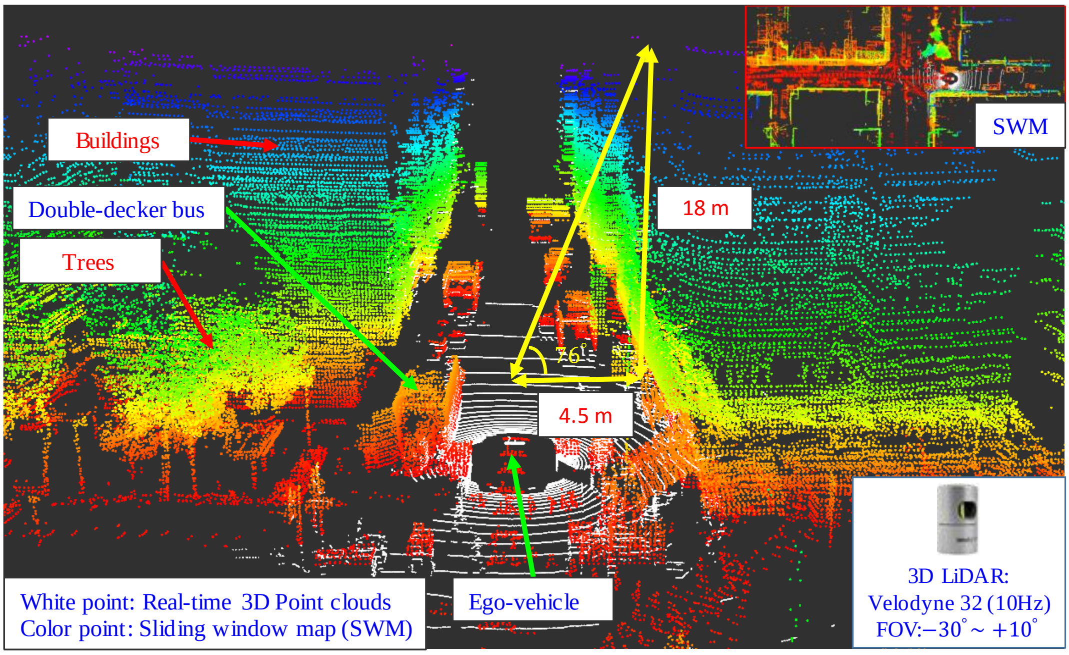

Figure 5-2. Demonstration of a generated sliding window map (SWM) and the real-time 3D point clouds. The white points represent the real-time 3D point clouds. The colored points come from the SWM, and the color is determined by the height.

To generate a point cloud map based on real-time 3D point clouds, simultaneous localization, and mapping (SLAM) [121] methods have been extensively studied over the past decades. Satisfactory accuracy can be obtained in a short period with low drift [122]. However, the error can accumulate over time, causing large errors after long-term traveling. Usually, only the objects inside a circle with a radius of 250 meters can cause GNSS NLOS receptions, and buildings far away can be ignored. We, therefore, employed only the last N_{sw} frames of the 3D point clouds to generate a sliding window map. In the conventional SLAM problem [122], the map is generated by accumulating the point clouds from the first epoch, 1, to the current epoch, t, as shown in the top panel of Figure 3. The pose of the point clouds is estimated by tracking the motion between consecutive frames of point clouds. However, error accumulates from the first epoch to the current epoch. As shown in Figure 5-3, x_t represents the pose of the keyframe [122] at epoch *t*. The proposed sliding window-based map generation is shown in the bottom panel of the figure, in which we consider only the keyframes inside a sliding window between epochs *i* and *t*. The drift error between epochs *i* and *t* for map generation is therefore bounded at a small value. The accuracy of an SWM relies heavily on the accuracy of motion estimation between consecutive frames of point clouds. In this work, we used the LiDAR odometry and mapping (LOAM) algorithm presented in [122] to generate the SWM. LOAM [122] is a state-of-the-art method that retains the top one accuracy in the KITTI dataset [123] for four years. Based on LOAM, as proposed in [122], three steps are included: feature extraction, point cloud registration, and local mapping. The rest of this section describes these three steps.

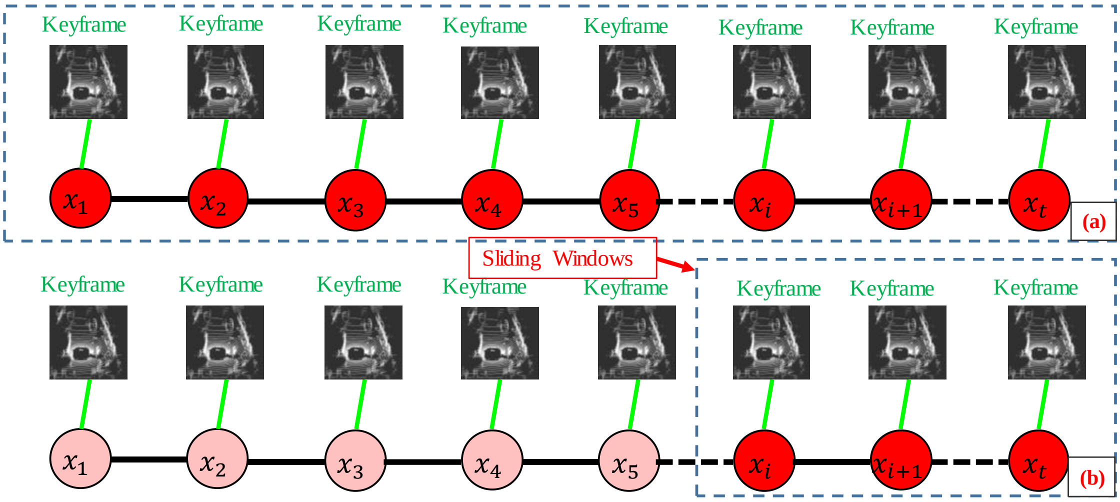

Figure 5-3. Comparison of (a) the conventional full SLAM and (b) the proposed sliding window map

Feature Extraction from 3D Point Clouds: As shown in Figure 5-1, the inputs into the map generation algorithm are raw 3D point clouds ($\mathbf{S}_t$) generated using 3DLiDAR. As described in Section 5.2, the orientation ($\mathbf{R}_t$) is used to transfer the $\mathbf{S}_t$ to a local frame as $\mathbf{P}_t$. The first step is feature extraction, which is carried out based on the work reported in [122]. The input into the feature extraction algorithm is the $\mathbf{P}_t\{P_{t,1}, P_{t,2}, \dots, P_{t,i}, P_{t,N}\}$ at a frequency of 10 Hz. *N* denotes the number of points inside the frame of a point cloud. The points are classified as planar points or edge points, depending upon the roughness of the points. The roughness of a point is determined as follows [122]:

$$c = \frac{1}{|\mathbf{S}|\cdot||P_{t,i}||} \left|\left| \sum_{j\in\mathbf{S},j\neq i}(P_{t,j} - P_{t,i}) \right|\right| \;, \tag{5-1}$$

where c represents the roughness of a point, $\boldsymbol{S}$ denotes a small local region near the point $P_{t,i}$, and $P_{t,j}$ indicates a point near $P_{t,i}$ inside the small local region. If the calculated roughness is larger than a pre-determined threshold [122], the point is classified as an edge point. Points with roughness lower than the threshold are classified as planar points. The output of the feature extraction process is the feature set $\mathbf{F}_t\{F_t^p, F_t^E\}$, in which F_t^p and F_t^E are feature sets containing all planar and edge points, respectively.

Point cloud registration: Point cloud registration is a process that estimates the relative motion between two consecutive frames of point clouds. The relative motion is calculated using point-to-edge and point-to-plane scan matching. The objective of this process is to find corresponding features for points in $\mathbf{F}_t\{F_t^p, F_t^E\}$ from the feature points set $\mathbf{F}_{t-1}\{F_{t-1}^p, F_{t-1}^E\}$. The detailed steps can be found in [122]. The point cloud registration process is as follows:

$$\mathbf{m}_{t-1,PR}^{t,PR} = \mathrm{PCR}(\mathbf{F}_t\{F_t^p, F_t^E\}, \mathbf{F}_{t-1}\{F_{k-1}^p, F_{t-1}^E\}), \tag{5-2}$$

where PCR denotes the point cloud registration function. The output of the point cloud registration process is the approximate relative motion, $\mathbf{m}_{t-1,PR}^{t,PR}$, at a frequency of 10 Hz.

Local mapping: To refine the relative motion estimation, the local mapping process based on [122] is applied to refine the motion estimation, $\mathbf{m}_{t-1,PR}^{t,PR}$. A detailed description can be found in [122]. The principle of the mapping process is that the extracted $\mathbf{F}_t\{F_t^p, F_t^E\}$ is mapped into the incrementally-built map to refine the motion estimate, $\mathbf{m}_{t-1,PR}^{t,PR}$. The output of local mapping is refined motion estimation, $\mathbf{m}_{t-1,LM}^{t,LM}$. All motion transformations between the keyframes in epochs i to k can be estimated, and the SWM obtained as $\mathbf{M}_t$. Note that $\mathbf{M}_t$ is in the local ENU frame, which is the same frame as the satellite elevation and azimuth angles. Then, $\mathbf{M}_t$ can be used in satellite visibility classification, as shown in the next section.

5.4 GNSS Pseudorange Measurements Rectification Based on SWM

From the measurements obtained from the GNSS receiver, each pseudorange measurement, ρ_n, can be described as follows [64]:

$$\rho_n = R_n + c(\delta t^r - \delta t_n^{sv}) + I_n + T_n + e_n \,, \tag{5–3}$$

where R_n is the geometric range between the satellite and the GNSS receiver, δt_n^{sv} denotes the satellite clock bias, δt^r is the receiver clock bias, $I_n$is the ionospheric delay distance, T_n is the tropospheric delay distance, and $e_n$represents the errors caused by factors such as multipath effects, NLOS receptions, receiver noise, and antenna delay. In a sparse area, e_n is small, and a small value can effectively bound the error caused by multipath and NLOS receptions. However, e_n can change dramatically in dense urban areas, and a fixed value cannot bound the error caused by GNSS signal reflections from the surroundings.

Recently, the work in [124] described a state-of-the-art general online sensor model validation and estimation framework. The framework consists of three parts: model validation, model calibration, and model repair. The authors propose that sensor measurements should be validated, calibrated or repaired before its integration with data from other sensors. The main reason behind this is that sensor measurements can be affected or polluted by environmental conditions, causing violations of the assumptions of the original sensor model. Many sensor measurements can violate the assumptions of the standard sensor model in challenging environments, such as urban canyons. For example, LiDAR-based positioning can be severely degraded in an urban canyon with numerous dynamic objects [125]. Therefore, a fixed sensor model cannot bound the potential error of LiDAR-based positioning. Therefore, the ability to effectively validate, calibrate and repair the sensor model as required is valuable for sensor fusion in such areas. Following the framework proposed by Jurado and Raquet [124], we applied the three phases to the GNSS pseudorange measurements (Figure5-4). First, model validation (Model 1 in Figure 5-4) was performed based on satellite visibility classification using SWM. Second, if one satellite is classified as NLOS, we proceed to the model calibration phase, which re-estimates the GNSS measurement by correcting the NLOS pseudorange measurements. However, if one satellite is classified as NLOS, but its reflecting point is not found inside the SWM, which means NLOS correction is not available, we proceed to the model repair phase (Model 3 in Figure 5-4) by de-weighting the NLOS measurements for use in further positioning. The remainder of this section describes these three phases.

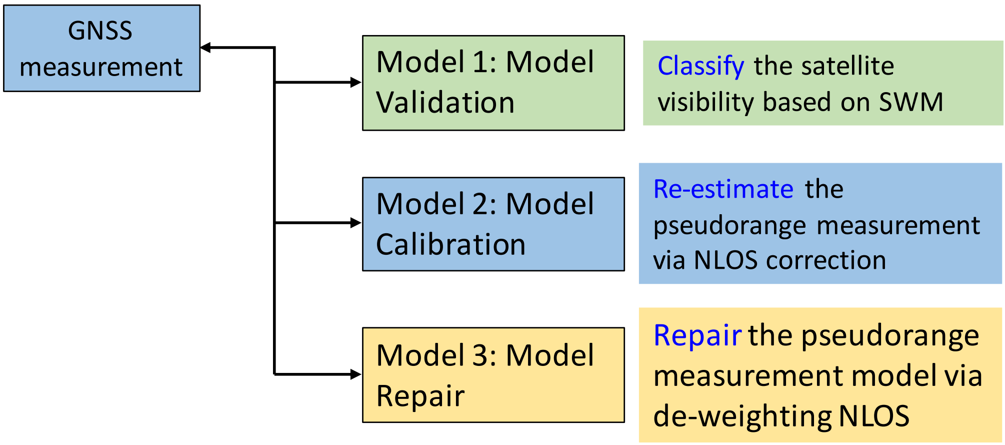
GNSS measurement
Model 1: Model Validation
Classify the satellite visibility based on SWM
Model 2: Model Calibration
Re-estimate the pseudorange measurement via NLOS correction
Model 3: Model Repair
Repair the pseudorange measurement model via de-weighting NLOS

Figure 5-4. Application of the framework proposed by Jurado and Raquet [124] to GNSS measurements before their use in the integration system. Three phases are included: model validation, model calibration, and model repair

5.4.1 Model Validation: NLOS Detection Based on SWM

In this section, we describe the details of NLOS detection based on the SWM generated as discussed in Section 5.3. Unlike the 3D building models, which consist of consistent surfaces from buildings [120], the SWM only provides unorganized discrete points. To effectively classify satellite visibility based on the SWM, we developed a fast searching method (Algorithm 5-1). The inputs of the algorithm include the SWM $\mathbf{M}_t$, elevation angle $\varepsilon_{t,i}$of satellite i, the azimuth angle $\alpha_{t,i}$ of satellite i at epoch t, the maximum searching distance, D_{thres}, and a constant incremental value, $\Delta\mathrm{d}_{\mathrm{pix}}$. The output is the satellite visibility, $\mathbf{Vis}_{t,i}$, of satellite i. In Step 1, a search point is initialized at $(x_{L,t}^{c}, y_{L,t}^{c}, z_{L,t}^{c})$, the center of the 3D LiDAR. A search direction connecting the GNSS receiver and the satellite is determined based on the elevation and azimuth angle of satellite i. The SWM is transformed into a *kdTree* structure [63], $\mathbf{M}_{t,tree}$, for finding neighboring points. The *kdTree* is a special structure for point cloud processing which can perform efficiently when searching neighboring points. In Step 2, given a fixed incremental value, $\Delta\mathrm{d}_{\mathrm{pix}}$, the search point is moved to the next point $(x_{L,t,k}^{s}, y_{L,t,k}^{s}, z_{L,t,k}^{s})$ calculated using (5-4)-(5-6), based on the search direction shown on the left-hand side of Figure 5. The number (N_k) of neighboring points near the search point is counted using the *kdTree* structure [63]. If N_k exceeds a certain threshold, there are some map points from buildings or dynamic objects near the search point $(x_{L,t,k}^{s}, y_{L,t,k}^{s}, z_{L,t,k}^{s})$, and we consider that the line-of-sight connecting the GNSS receiver and satellite is blocked. Therefore, satellite i is classified as an NLOS satellite. Otherwise, repeat Steps 2 and 3.

If $k\Delta\mathrm{d}_{\mathrm{pix}} > D_{thres}$, it means that the direction between the GNSS receiver and the satellite is line-of-sight. In this work, D_{thres} was set to 250 meters, so points within 250 meters were considered for NLOS detection. Only the direction connecting the GNSS receiver and the satellite needs to be considered, instead of traversing the whole SWM, an approach that contributes to the efficiency of NLOS detection. A satellite visibility classification result is shown on the right-hand side of Figure 5-5. The red and blue circles represent the NLOS and LOS satellites, respectively. The length of the white line connecting the center of 3D LiDAR and the satellite is D_{thres}. Therefore, satellite visibility can be classified using Algorithm 5-1. In our implementation, less than 10 ms was spent on classifying the visibility of each satellite.

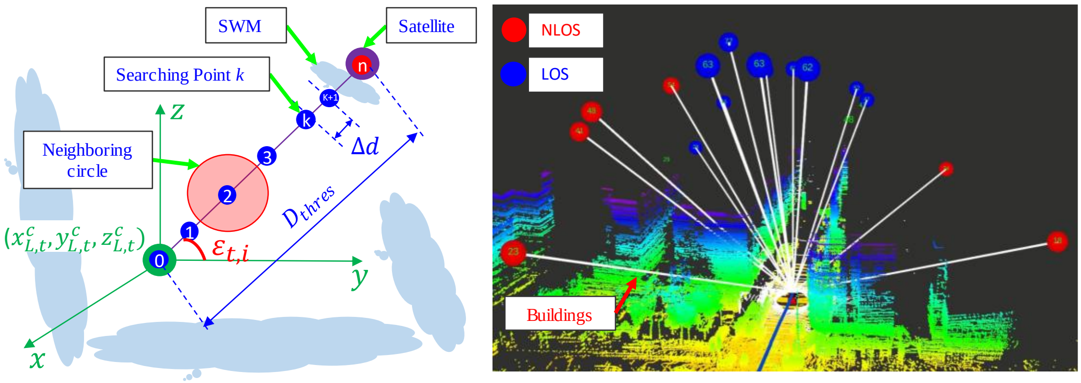

Figure 5-5. Illustration of NLOS detection based on SWM. A detailed video can be found here

Algorithm 5−1: NLOS Detection based on SWM

Inputs: Point clouds $\mathbf{M}_t$, $\varepsilon_{t,i}$, $\alpha_{t,i}$.

Outputs: Satellite visibility $\mathbf{Vis}_{t,i}$

Step 1: Initialize the searching point at $(x^c_{L,t}, y^c_{L,t}, z^c_{L,t})$, the searching direction denoted by $\varepsilon_{t,i}$ and $\alpha_{t,i}$, transfer the $\mathbf{M}_t$ into *kdTree* and get $\mathbf{M}_{t,tree}$

Step 2: Given a constant incremental value $\Delta\mathrm{d}_{\mathrm{pix}}$, the searching point is updated as follows:

$$x^s_{L,t,k} = x^s_{L,t,k-1} + \Delta\mathrm{d}_{\mathrm{pix}}\sin(\alpha_{t,i})\cos(\varepsilon_{t,i}) \qquad (5\text{-}4)$$

$$y^s_{L,t,k} = y^s_{L,t,k-1} + \Delta\mathrm{d}_{\mathrm{pix}}\cos(\alpha_{t,i})\cos(\varepsilon_{t,i}) \qquad (5\text{-}5)$$

$$z^s_{L,t,k} = z^s_{L,t,k-1} + \Delta\mathrm{d}_{\mathrm{pix}}\sin(\varepsilon_{t,i}) \qquad (5\text{-}6)$$

Step 3: if $k\Delta\mathrm{d}_{\mathrm{pix}} < D_{thres}$, find the nearest neighbor points (NNPs) of a given point $(x^s_{L,t,k}, y^s_{L,t,k}, y^s_{L,t,k})$ and get N_k NNPs.

Step 4: Repeat Step 2~3, until $N_k > N_{thres}$. Then the satellite is NLOS ($\mathbf{Vis}_{t,i}$=0), else LOS ($\mathbf{Vis}_{t,i}$=1)

The numbers inside the circles on the right-hand side of Figure 5-5 denote the elevation angle of the corresponding satellite. We can see that the NLOS satellite with an elevation angle of 54 degrees was detected. As shown in Figure 5-2, the maximum mask elevation angle can reach 76 degrees. In practice, the maximum mask elevation angle based on SWM was significantly correlated with the width of the street. The narrower the street was, the higher the mask elevation angle is achieved. Although the proposed SWM has effectively enhanced the FOV of LiDAR sensing compared with our previous work described in [10, 67, 117, 118], the SWM still could not fully reconstruct scenarios with very tall buildings. However, according to recent research described in [10], NLOS satellites with low elevation angles produce the majority of GNSS positioning error.

5.4.2 Model Calibration: NLOS Correction Based on SWM

This section presents the details of NLOS correction (model calibration) based on an SWM (Figure 5-4). The typical NLOS error model proposed in [3] is shown in Figure 5-6. The expected signal transmission route is expressed as a dashed blue line in Figure 5-6. The distance from the receiver to the building is represented by τ. The elevation angle of the GNSS signal is represented by θ_{ele}. We assume that:

(1) The surrounding buildings and dynamic objects which can cause potential NLOS receptions are vertical.

(2) GNSS signal reflections satisfy the law of reflection.

(3) The NLOS signals are reflected only once.

Thus, we can get $\theta_a = \theta_b$. Moreover, the direction of real signal transmission is parallel to the direction of expected signal transmission. Finally, we have $\theta_a = \theta_b = \theta_0 = \theta_{ele}$ .The route distance difference, γ, between the reflected signal and the expected signal is as follows:

$$\gamma = \gamma_1 + \gamma_2 \tag{5-7}$$

$$\gamma_1 = \tau \, sec\theta_{ele} \tag{5-8}$$

$$\gamma_2 = \gamma_1 \cos{(2\theta_{ele})} \tag{5-9}$$

Thus, NLOS error can be calculated based on the azimuth angle, elevation angle, and the distance from the receiver to the building causing the reflection. In general, two steps are needed to proceed with NLOS correction: NLOS detection (presented in Section 4.1) and NLOS error calculation based on the detected reflector. When correcting for NLOS satellites, the major difficulties lie in finding the reflectors which reflect the GNSS signals, to further estimate the distance between the GNSS receiver and the reflector.

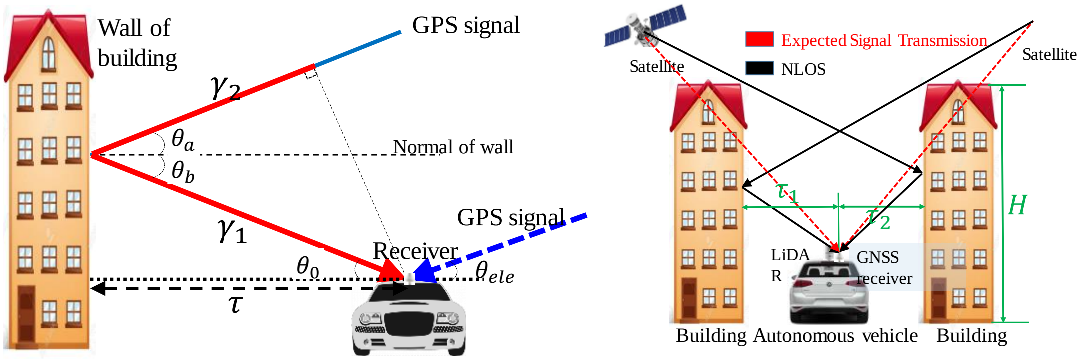

Figure 5-6. Illustration of the NLOS error model and NLOS signal transmission route. The signal is reflected by the building and subsequently received by the receiver [3]

A ray-tracing [126] technique is commonly used to simulate NLOS signal transmission routes for finding the NLOS reflectors in range-based 3DMA GNSS [16-19]. This approach can incur a high computational load. However, unlike the 3D building models, the SWM described in this paper does not produce continuous building surfaces and clear building boundaries. The SWM only provides large amounts of dense, discrete, unorganized point clouds; there are about 10 million points inside an SWM. Instead of applying the ray-tracing technique to find the reflectors inside the SWM, we directly search for the reflectors from the SWM, using an efficient *kdTree* structure.

The details of the reflector detection algorithm are presented in Algorithm 5-2. The inputs of the algorithm include the point clouds $\mathbf{M}_t$, the elevation angle at epoch *t* for NLOS satellite *i* ($\varepsilon_{t,i}$), the azimuth angle at epoch *t* for NLOS satellite *i* ($\alpha_{t,i}$), and the azimuth resolution, α_{res}, as in Algorithm 5-1. The output is the closest reflecting point, p_r^s, which is the most probable reflector for NLOS satellite i.

Step1. A search point is initialized at the center of the 3D LiDAR. The search direction is determined based on the satellite elevation, $\varepsilon_{t,i}$, and α_s. The SWM is transformed into a *kdTree* structure for use in the neighbor search, as in Algorithm 5-1.

Step2. Based on the assumption that $\theta_a = \theta_{ele}$ [20], the reflected signal should have the same elevation angle as the expected directed signal (Figure 5-6). Therefore, we traverse all the azimuths from 0 to 360 degrees, with an azimuth resolution of α_{res}

and elevation angle of $\varepsilon_{t,i}$, to find all the possible routes of NLOS transmissions. For example, for a given direction specified by $\varepsilon_{t,i}$ and α_s, the line-of-sight between the GNSS receiver and satellite is identified based on Algorithm 5-1.

Step3. If the line-of-sight is blocked by a point, p_j, (Step 2 in Algorithm 5-2) and the line-of-sight connecting the point p_j and the satellite i is not blocked (Step 3 in Algorithm 5-2), point p_j is considered as a possible reflector and is saved to $\mathbf{Q}_t$.

Step 4. The α_s proceed to the next azimuth based on Step 4. By repeating steps 2 and 3, all possible reflectors are identified, based on the assumption that $\theta_a = \theta_{ele}$. Figure 5-7 (a)- (d) shows the result of possible reflector detection for 1-2 NLOS satellites. We observe that multiple possible reflectors are found using Steps 1 to 4. The red circles in Figure 5-7 denote the NLOS satellite, and the red lines denote possible NLOS reflection and transmission routes. According to [3], the reflector with the shortest distance is usually the best candidate.

Step 5. A unique reflector can be detected based on the shortest distance assumption (Step 5), as shown in Figure 5-7 (e). Therefore, the reflecting point for a given satellite i is detected as p_r^s, and the distance (τ) needed in formula (5-8) can be calculated accordingly.

Algorithm 5-2: Reflecting Point Detection (RPD) based on SWM

Inputs: Point clouds $\mathbf{M}_t$, $\varepsilon_{t,i}$, $\alpha_{t,i}$ and azimuth resolution as α_{res}.

Outputs: Reflecting point p_r^s.

Step 1: Initialize the searching point at $(x_{L,t}^c, y_{L,t}^c, z_{L,t}^c)$, the searching direction denoted by $\varepsilon_{t,i}$ and $\alpha_{t,i}$, transfer the $\mathbf{M}_t$ into *kdTree* and get $\mathbf{M}_{t,tree}$. Initialize reflecting points array $\mathbf{Q}_t$. $\alpha_s = 0$.

Step 2: Get the first point p_j inside the map blocking the searching direction denoted by $\varepsilon_{t,i}$ and $\alpha_s$using Algorithm 1. if p_j is found, go to Step 3, otherwise go to Step 4.

Step 3: If the direction connecting the point and satellite is visible, save p_j to $\mathbf{Q}_t$

Step 4: α_s= α_s + α_{res}, repeat Step 1 to 2 until α_s>360°.

Step 5: find the most likely reflector p_j from $\mathbf{Q}_t$ with the shortest distance between the GNSS receiver and the reflector. Save p_j to p_r^s.

The proposed NLOS reflector detection method does not rely on the accuracy of the detection of building surfaces. The short distance assumption applied in Step 5 of Algorithm 5-2 can effectively prevent overcorrection, as only the closest reflector is identified as the unique reflector. Due to the sparsity of the SWM, although it is still denser than the 3D real-time point clouds, there are still some satellites whose reflectors cannot be found using the SWM. Therefore, we remodel NLOS satellites whose reflectors are not found, using the approach described in the next section, based on the framework shown in Figure 5-4.

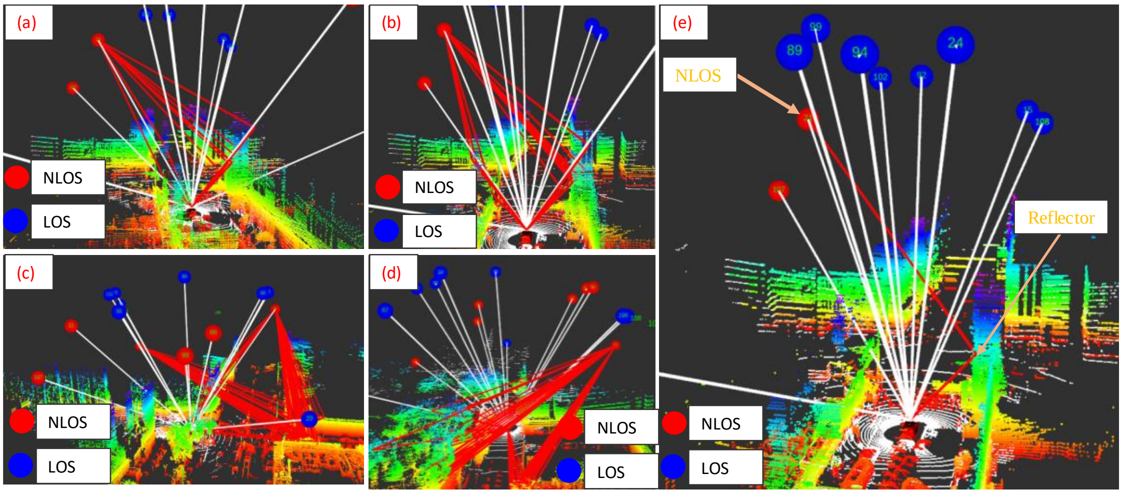

Figure 5-7. Demonstration of the NLOS signal reflector detection. Red and blue circles represent NLOS and LOS satellites, respectively. White lines denote LOS transmission routes. Red lines represent NLOS transmission routes. Multiple possible NLOS transmission routes are shown in (a)-(d). The most probably NLOS transmission route is shown in (e), based on the shortest route assumption adopted in Algorithm 5-2

5.4.3 Model Repair: NLOS Remodeling

If the reflector for an NLOS satellite is not detected, we remodel it as described in this section, instead of simply excluding it (Figure 5-4). According to [3], satellites with lower elevation angles and smaller signal to noise ratios (SNR) have a higher possibility of contamination by NLOS errors [3]. Pseudorange uncertainty modeling based on the satellite elevation angle and SNR was reported in [65, 127]. The weighting scheme in [65] produces satisfactory performance in open areas. However, the scheme may not work in dense urban areas, as the NLOS can have high elevation angles and SNR, as can be seen in our previous work [117]. This weighting scheme treats the LOS and

NLOS in the same manner, which is not preferable when the NLOS has already been detected. The weighting scheme in [127] employs a scaling factor to assign the LOS and NLOS different weightings. Inspired by this approach, we modeled the uncertainty of LOS and NLOS using the weighting scheme described in [65]. A scaling factor was added onto the scheme to treat the LOS and NLOS differently. Assume that $\mathbf{SV}_i$ represents the information from satellite i and $\mathbf{SV}_i = \{\varepsilon_i, \alpha_i, SNR_i, \rho_i\}$, α_i denotes the satellite azimuth angle, ε_i represents the satellite elevation angle, SNR_i indicates the satellite SNR, and ρ_i denotes the pseudorange measurement. The weighting scheme presented in [65] is as follows:

$$W_{LOS}^{(i)}(\varepsilon_i, SNR_i) = \frac{1}{sin^2\varepsilon_i}\left(10^{-\frac{(SNR_i-T)}{a}}\left(\left(\frac{A}{10^{-\frac{(F-T)}{a}}}-1\right)\frac{(SNR_i-T)}{F-T}+1\right)\right) \tag{5-10}$$

the parameter T indicates the SNR threshold. Parameters a, A and F are experimentally determined. The weighting matrix W is a diagonal matrix constructed by the weighting $W^{(i)}(ele_i, SNR_i)$. The weighting scheme treats LOS and NLOS using the same formula. We added a scaling factor, K, to adapt the weighting scheme to treat LOS and NLOS differently, as follows:

$$W_{NLOS}^{(i)}(el_i, SNR_i) = K \cdot W_{LOS}^{(i)}(el_i, SNR_i) \tag{5-11}$$

when the received signal is LOS, K is equal to 1. When the signal is NLOS, K changes, and is experimentally determined. In this work, the variance, σ_p, of a pseudorange measurement is computed as $\sigma_p = 1/W^{(i)}(el_i, SNR_i)$. In this case, satellite visibility is classified, and the corresponding uncertainty is modeled.

5.5 GNSS/INS Integration Using Factor Graph Optimization

In general, the goal of the multi-sensor integration is to find the optimal posterior state given measurements from sensors. Therefore, the sensor integration problem can be formulated as a typical maximum *a posteriori* (MAP) problem [108]. The measurements include two parts, the GNSS and INS measurements. Assuming that the GNSS and INS measurements are independent, we can formulate the GNSS/INS integration problem as:

$$\widehat{\mathrm{X}} = \arg\max \prod_{t,i} P(\boldsymbol{z}_{t,i}|\boldsymbol{x}_t) \prod_t P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) \tag{5-12}$$

where $\boldsymbol{z}_{t,i}$ represents the GNSS raw measurements at epoch *t*, and $\boldsymbol{x}_k$ represents the system state at epoch *t*. The index of measurements at a given epoch *t* is denoted as *i*, so one epoch can have multiple pseudorange measurements. The control input (INS measurements) is denoted by $\boldsymbol{u}_t$, and the optimal system state set is $\widehat{\mathbf{X}}$ [108]. A conventional Bayes filter-based method finds the best estimation of the current state considering only the previous state and the control input and observation measurements at the current epoch. This approach fails to take full advantage of historical information. Conversely, FGO-based sensor integration [128] has been used to transfer the MAP problem into a non-linear optimization problem.

In FGO-based integration, all sensor measurements are treated as factors [128] associated with specific states [128]. According to [129], the MAP problem can be expressed as:

$$\widehat{\mathbf{X}} = \arg\max_{\mathbf{X}}(\prod_j \zeta_j(\boldsymbol{x}_j)) \tag{5-13}$$

$$\text{with } \zeta_j(\boldsymbol{x}_j) \propto \exp(-||h_j(\boldsymbol{x}_j) - \boldsymbol{z}_j||^2_{\boldsymbol{\Sigma}_j}),$$

where $\zeta_j(\boldsymbol{x}_j)$ is a factor associated with the measurements $\boldsymbol{z}_j$, which can be derived from both GNSS and INS measurements. The $\boldsymbol{\Sigma}_j$ denotes the uncertainty associated with the given measurement $\boldsymbol{z}_j$.The state x_j is associated with the measurements $\boldsymbol{z}_j$. An observation function, $h_j(*)$, is associated with $\boldsymbol{z}_j$. The state set that needs to be estimated is denoted $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \dots, \boldsymbol{x}_k, \dots\}$. Assuming that all sensor noise has a Gaussian distribution, the negative logarithm of $\zeta_j(\boldsymbol{x}_j)$ is proportional to the error function [129] associated with the measurements. Therefore, Equation (5-13) can be transformed as follows:

$$\widehat{\mathbf{X}} = \arg\min_{\mathbf{X}}(\prod_j ||h_j(\boldsymbol{x}_j) - \boldsymbol{z}_j||^2_{\boldsymbol{\Sigma}_j}) \tag{5-14}$$

The FGO transforms the (5-13) into a standard non-linear least squares problem, as shown in (5-14), and is used to obtain the optimal state set, **X,** by minimizing the derived error function (5-14).

The graph structure of the GNSS/INS integration is shown in Figure 5-8. The state-space of the system is represented as:

$$\boldsymbol{x}_t = (\mathbf{X}^{ecef}_{t,r}, \mathbf{V}^{ecef}_{t,r}, \mathbf{B}^{body}_{t,r}, \boldsymbol{\delta}^{clock}_{t,r})^T \quad , \tag{5-15}$$

where $\boldsymbol{x}_t$ denotes the system state. Note that the $\boldsymbol{x}_t$ here is different from that in Figure 5-3. The position of the GNSS receiver in the ECEF coordinate [100] (denoted by the subscript, *r*) at given epoch *t* is represented by $\mathbf{X}_{t,r}^{ecef} = (x_{t,r}^{ecef}, y_{t,r}^{ecef}, z_{t,r}^{ecef})$. The velocities of the GNSS receiver in ECEF coordinate are denoted by $\mathbf{V}_{t,r}^{ecef} = (vx_{t,r}^{ecef}, vy_{t,r}^{ecef}, vz_{t,r}^{ecef})$, respectively. The bias of the accelerometer in the body (INS) frame is denoted by $\mathbf{B}_{t,ins}^{body} = (a_{t,x}^{body}, a_{t,y}^{body}, a_{t,z}^{body})$. The GNSS receiver clock bias is $\boldsymbol{\delta}_{t,r}^{clock}$. In Figure 5-8, the black-shaded rectangle represents the INS factor. The green- and red-shaded rectangles denote the LOS and NLOS satellite factors, respectively. The blue-shaded box represents the state transition factor.

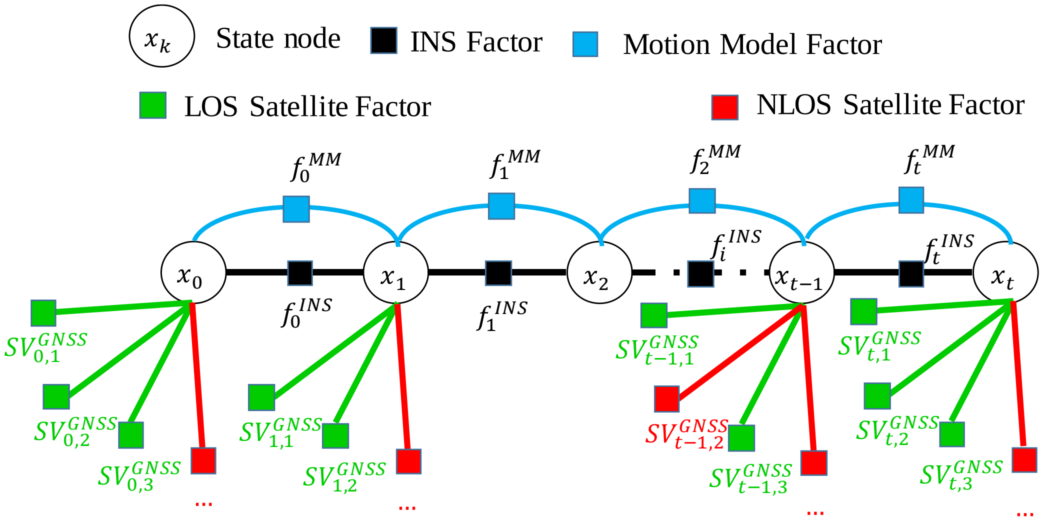

Figure 5-8. Graph structure of tightly-coupled GNSS/INS integration using FGO

5.5.1 Motion Model Factor

We used a constant velocity model [130] to constrain the two consecutive states. Based on the constant velocity model, the motion model can be expressed as:

$$\boldsymbol{x}_t = h^{MM}(\boldsymbol{x}_{t-1}) + N(0, \boldsymbol{\Sigma}_t^{MM}), \tag{5-16}$$

where $h^{MM}(*)$ represents the motion model function. Based on the constant velocity motion model, the motion model function can be expressed as follows:

$$h^{MM}(\boldsymbol{x}_{k-1}) = \begin{bmatrix} x_{t-1,r}^{ecef} + vx_{t-1,r}^{ecef} \cdot \Delta t \\ y_{t-1,r}^{ecef} + vy_{t-1,r}^{ecef} \cdot \Delta t \\ z_{t-1,r}^{ecef} + vz_{t-1,r}^{ecef} \cdot \Delta t \\ \mathbf{B}_{t-1,ins}^{body\ \ T} \\ \boldsymbol{\delta}_{t-1,r}^{clock} \end{bmatrix}, \tag{5-17}$$

where Δt is the time difference between the two states. The covariance matrix associated with the motion model is $\mathbf{\Sigma}_t^{MM}$. Therefore, the error function ($\mathbf{e}_t^{MM}$) of the motion model factor can be expressed as:

$$||\mathbf{e}_t^{MM}||^2_{\mathbf{\Sigma}_t^{MM}} = ||\boldsymbol{x}_t - h^{MM}(\boldsymbol{x}_{t-1})||^2_{\mathbf{\Sigma}_t^{MM}} \tag{5-18}$$

5.5.2 INS Factor

We make use of only the raw linear acceleration ($\mathbf{A}_t^{raw}$) and the attitude ($\mathbf{R}_{LB,t}$) from the AHRS. Therefore, the measurement from the accelerometer is expressed as follows:

$$\mathbf{A}_t^{raw} = (ax_t^{body}, ay_t^{body} az_t^{body})^T \ , \tag{5-19}$$

where $ax_t^{body}, ay_t^{body},$ and az_t^{body} represent the acceleration measurements in the body frame. As the estimated state, $\boldsymbol{x}_t$, is in the global frame (ECEF), we need to transform the acceleration measurements from the body frame to the global frame based on the orientation and its position in the last epoch. The transformed acceleration measurements $\mathbf{A}_t^{ecef} = (ax_t^{ecef}, ay_t^{ecef}, az_t^{ecef})^T$ are as follows [100]:

$$\mathbf{A}_t^{ecef} = \mathbf{R}_{GL,t}\mathbf{R}_{LB,t}(\mathbf{A}_t^{raw} - \mathbf{B}_{t,ins}^{body}) \ \ , \tag{5-20}$$

where $\mathbf{R}_{LB,t}$ is the transformation matrix used to transform the acceleration measurements from the body to the local frames, and can be expressed as follows:

$$\mathbf{R}_{LB,t} = \mathbf{R}_{LB}^{z}(\alpha)\mathbf{R}_{LB}^{y}(\beta)\mathbf{R}_{LB}^{x}(\gamma) \tag{5-21}$$

$$\text{with } \mathbf{R}_{LB}^{z}(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{LB}^{y}(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

$$\mathbf{R}_{LB}^{x}(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix}$$

where α, β, and γ denote the yaw, pitch and roll angles, respectively. The rotation matrices corresponding to the yaw, pitch, and roll angles are denoted $\mathbf{R}_{LB}^{z}(\alpha)$, $\mathbf{R}_{LB}^{y}(\beta)$ and $\mathbf{R}_{LB}^{x}(\gamma)$, respectively. The transformation matrix used to transform the acceleration measurements from the local frame to the global frame based on the $\boldsymbol{x}_k$ is denoted $\mathbf{R}_{GL,t}$ and can be expressed as follows:

$$\mathbf{R}_{GL} = \begin{bmatrix} -\sin(\emptyset_{lon}) & -\sin(\emptyset_{lat})\cos(\emptyset_{lon}) & \cos(\emptyset_{lat})\cos(\emptyset_{lon}) \\ \cos(\emptyset_{lon}) & -\sin(\emptyset_{lat})\sin(\emptyset_{lon}) & \cos(\emptyset_{lat})\sin(\emptyset_{lon}) \\ 0 & \cos(\emptyset_{lat}) & \sin(\emptyset_{lat}) \end{bmatrix} \tag{5-22}$$

where $\boldsymbol{\emptyset_{lon}}$ and $\boldsymbol{\emptyset_{lat}}$ represent the longitude and latitude based on the WGS84 geodetic system [100], which can be derived from $\boldsymbol{x}_k$. The measurement model for the linear acceleration is as follows:

$$\boldsymbol{x}_t = h^{INS}\left(\boldsymbol{x}_{t-1}, \mathbf{A}_t^{ecef}\right) + N(0, \boldsymbol{\Sigma}_t^{INS}), \tag{5-23}$$

with the measurement function $h^{INS}\left(\boldsymbol{x}_{t-1}, \mathbf{A}_t^{ecef}\right)$ as follows:

$$h^{INS}\left(\boldsymbol{x}_{t-1}, \mathbf{A}_t^{ecef}\right) = \begin{bmatrix} vx_{t-1,r}^{ecef} + ax_t^{ecef} \cdot \Delta t \\ vy_{t-1,r}^{ecef} + ay_t^{ecef} \cdot \Delta t \\ vz_{k-1,r}^{ecef} + az_k^{ecef} \cdot \Delta t \end{bmatrix}, \tag{5-24}$$

where the covariance matrix for the INS factor is $\boldsymbol{\Sigma}_k^{INS}$. We can formulate the error function for INS acceleration measurements as follows:

$$||\mathbf{e}_t^{INS}||^2_{\boldsymbol{\Sigma}_t^{INS}} = ||\boldsymbol{x}_t - h^{INS}\left(\boldsymbol{x}_{t-1}, \mathbf{A}_t^{ecef}\right)||^2_{\boldsymbol{\Sigma}_t^{INS}}, \tag{5-25}$$

where $\boldsymbol{\Sigma}_t^{INS}$ is constant, and is based on the specification of INS.

5.5.3 GNSS Pseudorange Factor

The GNSS pseudorange measurements, $\mathbf{SV}_t$, include the healthy pseudorange, $\mathbf{SV}_t^C$, the re-modeled pseudorange, $\mathbf{SV}_t^M$, and the corrected pseudorange, $\mathbf{SV}_t^C$ at epoch *t* can be expressed as follows:

$$\mathbf{SV}_t = \left\{\mathbf{SV}_{t,1}, \mathbf{SV}_{t,2}, \dots, \mathbf{SV}_{t,i}, \dots\ \mathbf{SV}_{t,N}\right\} \tag{5-26}$$

The position of the GNSS receiver is $\mathbf{X}_{t,r}^{ecef} = (x_{t,r}^{ecef}, y_{t,r}^{ecef}, z_{t,r}^{ecef})$. The position of a satellite $\mathbf{SV}_{t,i}$ is represented by $\boldsymbol{x}_{SV,i}^{xyz} = (x_{SV}^{ecef}, y_{SV}^{ecef}, z_{SV}^{ecef})^T$. Therefore, we can obtain the predicted GNSS pseudorange measurement for satellite $\mathbf{SV}_{t,i}$ as:

$$h^p\left(\mathbf{SV}_{t,i}, \mathbf{X}_{t,r}^{ecef}, \boldsymbol{\delta}_{t,r}^{clock}\right) = ||\boldsymbol{x}_{SV,i}^{xyz} - \mathbf{X}_{t,r}^{ecef}|| + \boldsymbol{\delta}_{t,r}^{clock}, \tag{5-27}$$

where $h^p(*)$ is the measurement function of the pseudorange. In this work, the earth rotation and atmospheric delay were calculated in advance using the model detailed in

[65]. The measured pseudorange from the GNSS receiver is expressed as $\rho_{SV,i}$ and is given by $h^p\left(\mathbf{SV}_{t,i}, \mathbf{X}_{t,r}^{ecef}, \boldsymbol{\delta}_{t,r}^{clock}\right)$ with additional Gaussian noise. We have the following formulation:

$$\rho_{SV,i} = h^p\left(\mathbf{SV}_{t,i}, \mathbf{X}_{t,r}^{ecef}, \boldsymbol{\delta}_{t,r}^{clock}\right) + N(0, \boldsymbol{\Sigma}_{t,i}^{SV}), \tag{5-28}$$

where $\boldsymbol{\Sigma}_{t,i}^{SV}$ is the uncertainty of the satellite measurement $\rho_{SV,i}$. We can calculate the error function for a given satellite measurement, $\rho_{SV,i}$, as follows:

$$||\mathbf{e}_{t,i}^{P}||_{\sigma_p{}^2}^2 = ||\rho_{SV,i} - h^p\left(\mathbf{SV}_{t,i}, \mathbf{X}_{t,r}^{ecef}, \boldsymbol{\delta}_{t,r}^{clock}\right)||_{\boldsymbol{\Sigma}_{t,i}^{SV}}^2 \tag{5-29}$$

5.5.4 Efficient Incremental Optimization

In this section, we formulate three kinds of factors including the motion model factor, the INS (accelerometer) factor, and the GNSS pseudorange factor. The optimal state set $\mathbf{X}$ can be solved as follows:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \textstyle\sum_{i,k} ||\mathbf{e}_{t,i}^{P}||_{\boldsymbol{\Sigma}_{t,i}^{SV}}^2 + ||\mathbf{e}_{t}^{MM}||_{\boldsymbol{\Sigma}_{t}^{MM}}^2 + ||\mathbf{e}_{t}^{INS}||_{\boldsymbol{\Sigma}_{t}^{INS}}^2 \tag{5-30}$$

To solve the optimization problem, we used ISAM2 [129] in GTSAM [131].

5.6 Experimental Results

5.6.1 Experimental Setup

Two experiments were conducted in typical urban canyons in Hong Kong on 12 October 2019 (Figure 5-9). The figure on the left shows the test vehicle with all the sensors installed in a compact sensor kit. The middle and right figures show the urban canyons used. Both of the urban scenarios contain static buildings, trees and dynamic objects, such as double-decker buses and cars. We are aware of the limitation of the method described of the proposed method, that the sliding window map cannot sense the roof of buildings above 40 meters high in dense urban environments. We first carried out the experiment in a typical urban canyon in Hong Kong (Urban Canyon 1 in Figure 5-9). Then we performed another experiment in a highly urbanized area in Hong Kong (Urban Canyon 2 in Figure 5-9), in which the buildings are significantly higher, and which is one of the densest areas in Hong Kong, to study the limitations of the proposed method. Some NLOS satellites reflected by buildings taller than 40 meters may not be detected using SWM. In both experiments, a u-blox M8T GNSS receiver was used to collect raw GPS/BeiDou measurements at a frequency of 1 Hz. A 3D LiDAR sensor (Velodyne 32) was employed to collect raw 3D point clouds at a frequency of 10 Hz. The Xsens Ti-10 IMU was employed to collect data at a frequency of 100 Hz. Besides, the NovAtel SPAN-CPT, a GNSS (GPS, GLONASS, and Beidou) RTK/INS (fiber-optic gyroscopes, FOG) integrated navigation system was used to provide ground truth of positioning. The gyro bias in-run stability of the FOG is 1 degree per hour, and its random walk is 0.067 degrees per hour. The baseline between the rover and the GNSS base station is about 7 km. All the data were collected and synchronized using a robot operation system (ROS) [66]. The coordinate systems between all the sensors were calibrated before the experiments. The experimentally determined parameters used in this paper are shown in Table 5-1.

Table 5-1. Parameter values used

Parameters	N_{sw}	$\Delta\mathrm{d}_{\mathrm{pix}}$	α_{res}	N_{thres}
Values	50	2	1°	10
Parameters	K	a	A	F
Values	1.65	30	32	10

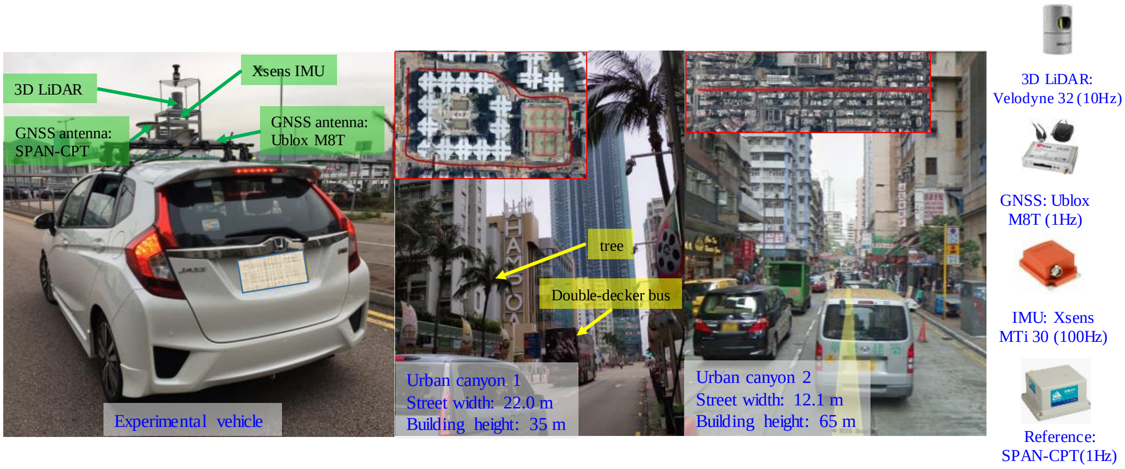

Figure 5-9. Experimental vehicle and tested scenarios of Urban Canyons 1 and 2

We first analyzed the performance of GNSS standalone positioning by comparing five single point positioning (SPP) methods, as shown below. The objective of this analysis was to validate the effectiveness of the proposed method in improving the GNSS standalone positioning.

(a) **u-blox:** the GNSS positioning solution from the u-blox M8T receiver.

(b) **WLS**: weighted least squares (WLS) method [65].

(c) **WLS-NE**: weighted least squares (WLS) method [65] with all NLOS satellites excluded.

(d) **R-WLS**: WLS method with the aid of the re-weighting scheme in equation (11); all NLOS satellites were re-weighted.

(e) **CR-WLS** (proposed SPP): WLS method with the aid of (1) the NLOS correction proposed in Section 4.2 if the reflector was detected, and (2) the re-weighting scheme in equation (11) if the reflector was not detected.

Three GNSS/INS integrated positioning methods were also compared:

(1) **EKF**: Standard EKF-based tightly coupled GNSS/INS integration based on [132].

(2) **FG**: Factor graph-based tightly-coupled GNSS/INS integration[118].

(3) **FG-3DLA** (proposed integration): Factor graph-based tightly coupled GNSS/INS integration with the help of 3D LiDAR aided GNSS, as described in this paper.

5.6.2 Evaluation of Urban Canyon 1 Experiment

(1) GNSS Standalone Positioning

The results of the GNSS standalone positioning experiments using the five methods are shown in Table 5-2. The first column shows the 2D positioning error of the u-blox receiver. The positioning result is based on standard NMEA [100] messages from the u-blox receiver. A mean error of 31.02 meters was obtained, with a standard deviation of 37.69 meters. The maximum error reached 177.59 meters due to excessive reflections from surrounding buildings. The GNSS solution was available throughout the experiment. The second column shows the positioning result using the raw pseudorange measurements from the u-blox receiver and positioning based on WLS. The weighting scheme was taken from [65] and is based on the satellite elevation angle and the signal to noise ratio (SNR). The positioning error decreased to 9.57 meters with a standard deviation of 7.32 meters. The maximum error also decreased to less than 50 meters. Based on the NLOS detection results shown in Section 5.4.1, the positioning error increased to 11.63 meters after excluding all detected NLOS satellites, a result that is even worse than that of the WLS. This situation arose because excessive NLOS exclusion can significantly distort the perceived geometric distribution of the satellites. Our previous results, described in [10, 117], showed a similar phenomenon. The standard deviation also increased compared with that of the WLS. Availability decreased slightly from 100% (WLS) to 96.01%. Complete NLOS exclusion is therefore not preferable in urban canyons. The fourth column of the table presents the results of R-WLS where all the NLOS satellites were remodeled based on the weighting scheme described in Section 5.4.3, instead of excluding the NLOS satellites detected. The 2D mean error was reduced from 9.57 meters (WLS) to 9.01 meters. Both the standard deviation and maximum errors decreased slightly. The last column shows the 2D positioning error of CR-WLS. The 2D positioning error decreased to 7.92 meters, with a standard deviation of 5.27 meters. Availability is also guaranteed using the proposed method (CR-WLS). The improved GNSS standalone positioning results demonstrate the effectiveness of the proposed method in mitigating the effects of NLOS signals.

Table 5-3 depicts the accuracy of NLOS satellite detection. As mentioned in Section 5.4.1, the proposed SWM cannot fully construct all environments, so some NLOS satellites with high elevation angles cannot be detected. Therefore, we evaluated the NLOS detection performance at three elevation angle ranges. The second row in Table 5-3 shows the percentage of NLOS satellites that belonged to a certain elevation angle range. The NLOS satellites with elevation angles between 0° and 30° made up 43.8% of all NLOS satellites. Of these NLOS satellites, 92% were detected using the method described in Section 5.4.1. The NLOS detection accuracy for NLOS satellites (28.9%) with elevation angles between 30° and 60° was 35%. Similar NLOS detection accuracy (27.35%) was obtained for NLOS satellites with elevation angles between 60° and 90°. Although the NLOS satellites with high elevation angles were not detected effectively, the proposed method is a new and general solution for NLOS detection. Due to the decreased cost of 3D LiDARs, multiple 3D LiDARs [133] is a common sensor setup for safety-critical ADV, to guarantee robustness. The use of multiple 3D LiDARs can significantly enhance the FOV of the proposed SWM. Therefore, NLOS satellites with high elevation angles can be detected by autonomous driving vehicles using multiple 3D LiDARs [133].

Table 5-2. Positioning performance of GNSS SPP in Urban Canyon 1

GNSS Positioning	u-blox	WLS	WLS-NE	R-WLS	CR-WLS
Mean error	31.02 m	9.57 m	11.63 m	9.01 m	7.92 m
Std	37.69 m	7.32 m	13.05 m	6.90 m	5.27 m
Maximum error	177.59 m	46.29 m	52.93 m	43.59 m	41.75 m
Availability	100 %	100%	96.01 %	100%	100%

Table 5-3. Performance of NLOS classification between different elevation ranges in Urban Canyon 1

All data	Elevation (0°–30°)	Elevation (30°–60°)	Elevation (60°–90°)
Percentage of NLOS Satellites	43.8 %	28.9%	27.35 %
Accuracy of NLOS Detection at Different Elevation Angles	92 %	35 %	21%

Tables 5-4 and 5-5 show the values of NLOS correction using the proposed method in two selected epochs. In Table 5-4, NLOS satellite 8, with an elevation angle of 23.49° and C/N_0 of 15 dB-Hz, was detected and the NLOS correction was 10.08 meters. The fourth column shows the exact NLOS delay, labeled using a ray-tracing technique [16] based on ground truth positioning provided by the reference system SPAN-CPT. We can see that the exact NLOS delay (15.55 meters) was slightly larger than the NLOS delay estimated using the proposed method. The major reason for the difference is that the proposed method finds the reflectors based on the shortest distance assumption. Therefore, the reflector may not be perfectly detected. In general, we find that NLOS satellites with lower elevation angles usually cause larger NLOS delay, as shown in column four of Table 5-4. The other epoch data shown in Table 5-5 show a slightly different trend. Satellite 30, with an elevation angle of 56.22 degrees, caused the largest NLOS delay, of 48.52 meters. According to (5-7) and Figure 5-6, NLOS delay is determined by the elevation angle and the distance between the GNSS receiver and the reflector. The main reason for the large NLOS delay caused by satellite 30 is the long distance between the GNSS receiver and the reflector. The majority of NLOS satellites were detected and corrected using the proposed method. Figure 5-10 shows a case in which the NLOS satellites were blocked by a traffic signal pole instead of 3D buildings. Conventionally, the 3DMA GNSS only considers static buildings. However, infrastructure such as traffic signal pole and even guard bars can also cause NLOS receptions. Satellites 7, 99 and 112 were all blocked by the signal pole. With increased complexity and density of infrastructure [134], which is not included in conventional 3D building models, NLOS receptions caused by such structures should also be considered. We believe that this is a significant contribution of the proposed method.

Table 5-4. NLOS pseudorange correction in Urban Canyon 1 (Epoch 33661)

Satellite PRN	Elevation Angle (degree)	C/N$_0$ (dB-Hz)	Actual Pseudorange Correction (ground truth)	Estimated Pseudorange Correction
8	23.49	15	15.55 m	10.08 m
17	23.13	18	13.73 m	8.14 m
11	62.45	24	3.87 m	7.59 m

Table 5-5. NLOS pseudorange correction in Urban Canyon 1 (Epoch 33730)

Satellite PRN	Elevation Angle (degree)	C/N$_0$ (dB-Hz)	Actual Pseudorange Correction (ground truth)	Estimated Pseudorange Correction
22	26.91	19	12.02 m	10.17 m
28	28.60	18	16.41 m	9.47 m
30	56.22	30	48.52 m	27.31 m

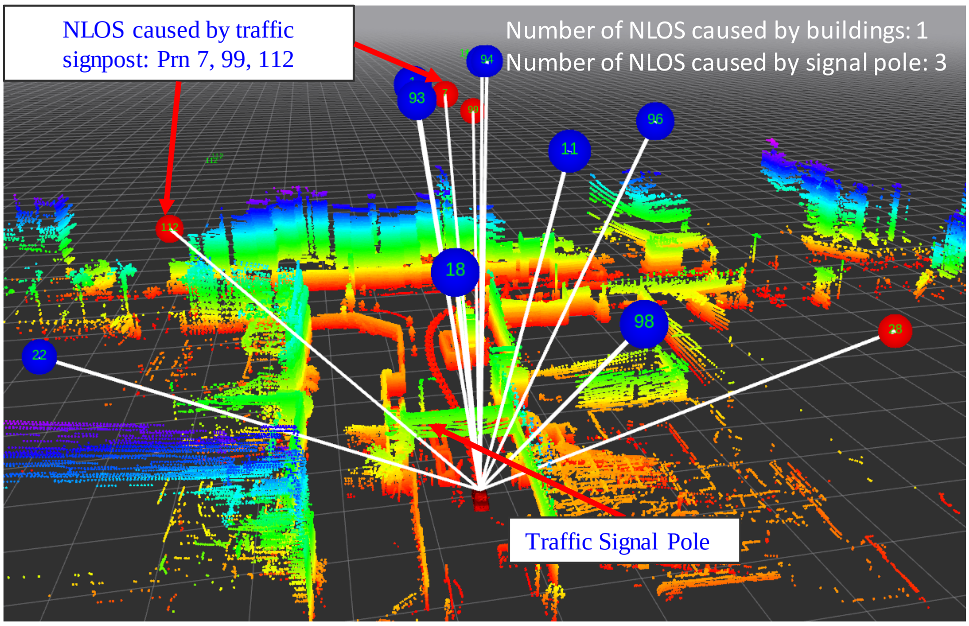
NLOS caused by traffic signpost: Prn 7, 99, 112
Number of NLOS caused by buildings: 1
Number of NLOS caused by signal pole: 3
94
93
96
11
18
98
22
28
Traffic Signal Pole

Figure 5-10. Illustration of NLOS receptions blocked by an overhead traffic signal pole instead of 3D buildings. The blue and red circles denote the LOS and NLOS satellites, respectively. The numbers inside the circles denote the satellite PRN

In short, the best performance of GNSS standalone positioning was obtained using CR-WLS. These improved results show that the proposed method can mitigate the effects of NLOS receptions by remodeling and correcting NLOS signals. Due to the complementarity of GNSS and INS, the remodeling and correction of GNSS measurements can contribute to the GNSS/INS integration.

(2) GNSS/INS Integrated Positioning

In this section, we present the results of GNSS/INS integration. Table 5-6 shows the 2D positioning errors identified using the three kinds of GNSS/INS integration listed. A 2D mean error of 8.03 meters was obtained using EKF, with a maximum error of 44.55 meters. Significantly improved positioning accuracy was obtained after the application of the state-of-the-art FGO, with the mean error decreasing from 8.03 to 3.64 meters. Both the standard deviation and the maximum error decreased. Our recent research, described in [118], extensively evaluated the performance of GNSS/INS integration using EKF and factor graphs. The improvement caused by the factor graph is due to the iteration and re-linearization employed during factor graph optimization [118]. Unlike conventional EKF based GNSS/INS integration, the FGO makes use of historical measurements during optimization, which exploits the connectivity between historical states and measurements. Unfortunately, the improvements from the FGO are still limited if the GNSS measurements w not well modeled. The maximum error still reached 23.56 meters. The positioning error still fluctuates dramatically between epochs 190 and 205 (left-hand side of Figure 5-11). The main reason for this variability is the high number of unmodeled NLOS measurements. With the use of the proposed method, the 2D mean error decreased from 3.64 meters (FG) to 2.8 (FG-3DLA) meters. The standard deviation was also reduced to 1.62 meters. The maximum 2D error was reduced from 23.56 meters (FG) to 9.71 meters. These improved results show the effectiveness of the proposed method. Although GNSS standalone positioning using the proposed CR-WLS still reaches 7.92 meters, GNSS/INS integration using FGO can effectively make the best use of the pseudorange correction (model calibration in Section 5.4.2) and uncertainty modeling (model repair in Section 5.4.3). After applying the 3D LiDAR aided GNSS positioning, the performance of GNSS/INS integration using the state-of-the-art factor graph was pushed significantly higher.

Table 5-6 Positioning performance of GNSS/INS integration in Urban Canyon 1

GNSS/INS	EKF	FG	FG-3DLA
Mean error	8.03 m	3.64 m	2.80 m
Std. dev.	7.60 m	3.19 m	1.62 m
Maximum error	44.55	23.56 m	9.71m

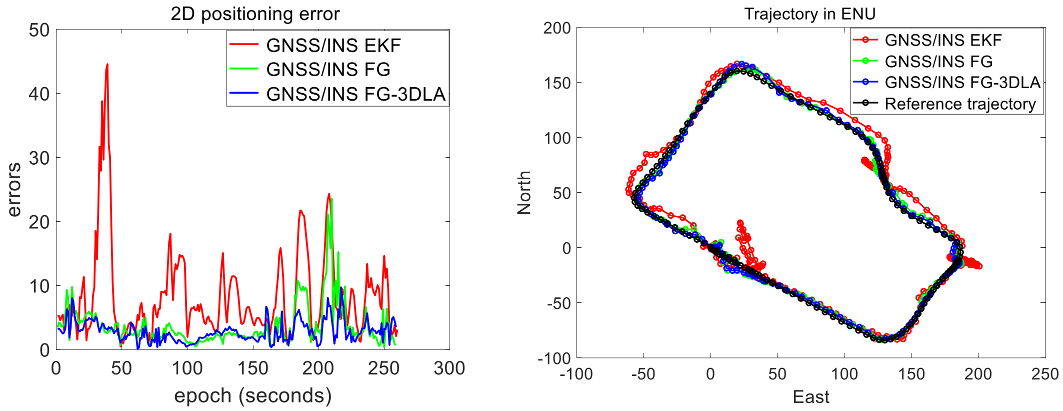

Fig. 11. 2D positioning error and trajectories of the GNSS/INS integrations in Urban Canyon 1

5.6.3 Evaluation of Urban Canyon 2 Experiment

To investigate the performance of the proposed method, another experiment was conducted in a denser urban canyon. (Figure 5-9). As with Experiment 1, NLOS satellites with high elevation angles cannot be fully detected using the SWM. We also wanted to explore what would happen in a denser urban canyon, using the proposed method.

(1) GNSS Standalone Positioning

As with Experiment 1, the results of the GNSS standalone positioning experiment are presented to show the effectiveness of the proposed method in GNSS positioning. A positioning error of 30.68 meters was obtained using the u-blox receiver with a maximum error of 92.32 meters. A GNSS positioning error of 23.79 meters was obtained using WLS based on the raw pseudorange measurements from the u-blox receiver. The maximum error increased slightly to 104.83 meters, compared with the GNSS positioning using data directly from the u-blox receiver. After excluding all detected NLOS satellites from the GNSS positioning (WLS-NE), both the mean and standard deviation increased to 25.14 and 23.73 meters, respectively. The availability of GNSS positioning data decreased to 95.52%, due to the lack of satellites for GNSS positioning, which again shows that complete NLOS exclusion is not optimal in urban canyons. With the help of NLOS remodeling (Section 4.3), the 2D error decreased to 19.61 meters by R-WLS. One hundred percent availability is guaranteed. The GNSS positioning error was further decreased to 17.09 meters using the CR-WLS method. The improvement in the results shows the effectiveness of the proposed method for

GNSS standalone positioning. The maximum error still reached 71.28 meters, because not all NLOS satellites can be detected and mitigated.

To further investigate these results, we also examined the percentage of NLOS satellites within certain elevation angle ranges. The percentage trend is almost opposite the trend in Experiment 1. The majority (44%) of the NLOS satellites belonged to the 60°–90° group in Experiment 2. However, the majority of NLOS satellites belonged to the 0°–30° group in Experiment 1. In Experiment 2 the buildings were higher, and the streets narrower, than in Experiment 1 (Figure 5-9). Apart from undetected NLOS satellites, the error caused by multipath effects was not mitigated. Even so, improved GNSS standalone positioning results were obtained, with the 2D error decreasing from 23.79 meters (WLS) to 17.09 (CR-WLS) meters. The NLOS detection accuracy for the satellites in the low elevation angle group (0°–30°) was still more than 90%, similar to Experiment 1. The NLOS detection accuracy for the satellites of the high elevation angle group (60°–90°) was limited (12%). However, the proposed method can easily be adapted to ADV with multiple 3D LiDARs to further detect NLOS satellites with high elevation angles. Since the proposed NLOS detection method relies on the orientation from the AHRS, we also present the effect of orientation error on NLOS detection accuracy. The last row of Table 8 shows the accuracy of NLOS detection at different angle ranges, using the ground truth orientation provided by the SPAN-CPT.

Table 5-7. Positioning performance of GNSS SPP in Urban Canyon 2

GNSS Positioning	u-blox	WLS	WLS-NE	R-WLS	CR-WLS
Mean error	30.68 m	23.79 m	25.14 m	19.61 m	17.09 m
Std. dev.	26.53 m	18.22 m	23.73 m	19.80 m	20.95 m
Maximum error	92.32	104.83 m	109.30	86.14 m	71.28 m
Availability	100%	100%	95.52%	100%	100%

Table 5-8. Performance of NLOS classification between different elevation ranges in Urban Canyon 2

All data	Elevation (0°–30°)	Elevation (30°–60°)	Elevation (60°–90°)
Percentage of NLOS Satellites	17.7 %	38.3%	44.0 %

Accuracy of NLOS Detection at Different Angle	90.7 %	46.0%	12.0%
Accuracy of NLOS Detection at Different Angle (SPAN-CPT)	91.3 %	47.1%	12.5%

(2) GNSS/INS Integrated Positioning

In this section, we present the results of GNSS/INS integration. Table 5-9 shows the 2D positioning errors using the three kinds of GNSS/INS integration, similar to Experiment 1. A 2D mean error of 20.89 meters was obtained using EKF, with a maximum error of 90.27 meters. Slightly improved positioning accuracy was obtained after applying the state-of-the-art factor graph, with the mean error decreasing from 20.89 to 18.54 meters. Both the standard deviation and the maximum error decreased. The improvement caused by the state-of-the-art factor graph was not as large as that in Experiment 1, because the percentage of polluted GNSS signals (multipath effects and NLOS receptions) is significantly larger in Urban Canyon 2, which has taller buildings and narrower streets. The improvements from the use of FGO were still limited since the GNSS measurements are not well modeled.

With the use of the proposed method, the 2D mean error decreased from 18.54 meters (FG) to 13.32 (FG-3DLA) meters. The standard deviation was also reduced to 10.11 meters. The maximum 2D error was reduced from 77.10 meters (FG) to 43.82 meters. The improved results show the effectiveness of the proposed method. Figure 5-13 shows the positioning errors and trajectories generated in Experiment 2. Positioning was improved almost throughout the experiment. From epoch 350 to epoch 450 the positioning error decreased. Interestingly, the positioning error from FG-3DLA near epoch 300 was even larger than the one from FG which is mainly caused due to the misclassification of satellite visibility. Although the GNSS standalone positioning using the proposed CR-WLS still reached 17.09 meters, the GNSS/INS integration using FGO could make the best use of the pseudorange correction (model calibration in Section 5.4.2) and uncertainty modeling (model repair in Section 5.4.3) with the positioning error decreasing to 13.32 meters.

Table 5-9. Positioning performance of GNSS/INS integration in Urban Canyon 1

GNSS/INS	EKF	FG	FG-3DLA
Mean error	20.89 m	18.54 m	13.32 m
Std. dev	23.44 m	19.18 m	10.11 m
Maximum error	90.27 m	77.10 m	43.82 m

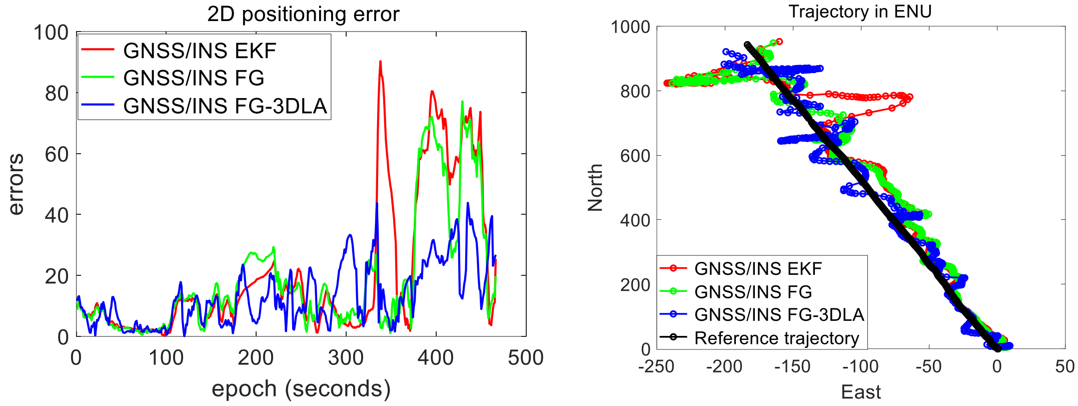

Figure 5-13. 2D positioning error and trajectories of the GNSS/INS integrations in Urban Canyon 1

Although the mean positioning error was significantly improved compared with the 30.68 meters obtained using u-blox, it still reached 13.32 meters. The remaining error arises from two major sources: 1) undetected NLOS satellites; and 2) unexpected multipath effects. Table 5-10 shows the pseudorange errors caused by the multipath effects and NLOS. Satellite 15 introduced the maximum pseudorange error of 37.92 meters among the six satellites. Multipath effects can also cause errors of with similar magnitude; for example, satellite 21 had a pseudorange error of 34.88 meters. Therefore, unmodeled multipath is a major factor causing the remaining 13.32 meters of positioning error. Fortunately, the multipath can be further mitigated using a higher level GNSS antenna, which is acceptable for autonomous driving vehicles.

Table 5-10. Pseudorange errors in Urban Canyon 2 (Epoch 401793)

Satellite PRN	Elevation Angle (degree)	C/N$_0$ (dB-Hz)	Type	Pseudorange Error
15	51.6	31	NLOS	37.29 m
21	48.70	26	Multipath	34.88 m
89	63.1	27	NLOS	5.71 m
92	61.63	33	Multipath	5.49 m
94	62.32	32	Multipath	5.14 m
102	60.98	34	Multipath	7.77 m

6. CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

GNSS positioning is currently still the major source of globally referenced positioning for intelligent transportation systems (ITS). However, accurate GNSS positioning in urban canyons is still a challenging problem. NLOS receptions currently remain the major problems for GNSS positioning in urban canyons. Therefore, effectively identifying and mitigating the effects of NLOS receptions is a significant step in achieving and popularizing accurate GNSS positioning solutions, such as SPP, real-time kinematic (RTK), and precise point positioning, in urban canyons. Since the performance of GNSS positioning relies heavily on environmental conditions, the state-of-the-art range-based 3DMA method proposes to effectively mitigate NLOS effects, based on offline environment descriptions known as 3D building models. However, with the increasing complexity and dynamics of city infrastructures, 3D building models cannot fully describe the real-time environment. Reconstructing the real-time environment based on onboard sensing is a promising method for identifying potentially polluted GNSS signals. Unlike the state-of-the-art 3DMA GNSS method, this thesis explores to proposes to improving the GNSS positioning with the assistant of onboard sensing. First, an object detection aided NLOS exclusion algorithm based on 3D LiDAR is proposed to mitigate the impacts of NLOS signals caused by surrounding dynamic objects. This is so far the first work that considers the NLOS caused by the dynamic objects. Secondly, instead of directly excluding the NLOS signals which have a drawback of distorting the geometry distribution of satellites, an NLOS correction method is proposed based on a 3D LiDAR sensor. The method does not require the 3D building models and the initial guess of the position of the GNSS receiver. The potential of the proposed method in GNSS/LiDAR integration is studied subsequently and the results show the effectiveness of the proposed method. As both of the two methods share the drawback of (1) the limited FOV of 3D LiDAR causes limited environment sensing capability; (2) the performance of NLOS detection relies heavily on the accuracy of object detection, such as double-decker bus detection in [67] and building detection in [10]; and (3) the reflector detection method described in [67] [10] [117] can only work when a building surface is detected and is sufficiently regular.

The work presented in Section 5 effectively relaxes these three drawbacks where a novel 3D LiDAR aided the GNSS positioning method which makes use of an onboard 3D LiDAR sensor to reconstruct the surrounding environment. Potential NLOS receptions caused by static buildings, dynamic objects, and even semi-static infrastructure (traffic signpost in Figure 5-10) can be detected, remodeled, and even corrected. This proposed method is a continuation of the previous work described in [67] [10] [117]. Overall, this thesis continuously focuses on solving the problem of GNSS NLOS in urban canyons by proposing a 3D LiDAR aided GNSS positioning method. A general solution is proposed to mitigate the effects of NLOS receptions and the proposed method can be easily adapted to the systems with multiple 3D LiDARs, and NLOS satellites with high elevation angles can be detected accordingly. We believe that the proposed method in this thesis can have a positive impact on both the academic and industrial fields.

6.2 Future Work

Using a low-cost sensor setup to obtain high accuracy is an important factor that could lead to the population of autonomous driving vehicles. This thesis has effectively improved the performance of the GNSS positioning in highly urbanized areas. However, the positioning error could still reach more than 10 meters in the evaluated areas. The future works will focus on two parts:

(1) More onboard sensors, such as cameras, vehicular odometers, will be integrated with the GNSS solution with the help of the proposed method. Multi-sensor fusion is the most promising solution for achieving centimeter-level positioning for autonomous driving vehicles.

(2) The proposed method can only mitigate the impacts of the GNSS NLOS receptions. The performance degradation caused by the multipath effects is not mitigated based on the proposed 3D LiDAR aided GNSS positioning method. In future work, we will combine the robust model studied in [102-104] to model the multipath effects. Moreover, the Doppler and carrier-phase measurements will be integrated with the pseudorange measurements using the proposed 3D LiDAR aided GNSS method.

(3) With the fast development of the communication technique, the vehicle to everything (V2X) technique enables collaborative positioning based on

information from multiple agents. In the future, we will also explore the potential of V2X in improving the GNSS positioning and even the overall positioning accuracy.

REFERENCES

[1] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics,* vol. 25, no. 8, pp. 425-466, 2008.

[2] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences,* vol. 368, no. 1928, pp. 4649-4672, 2010.

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

[4] P. D. Groves, Z. Jiang, L. Wang, and M. K. Ziebart, "Intelligent urban positioning using multi-constellation GNSS with 3D mapping and NLOS signal detection," 2012.

[5] 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,* 2019.

[6] S. H. Kong, "Statistical Analysis of Urban GPS Multipaths and Pseudo-Range Measurement Errors," *IEEE Transactions on Aerospace & Electronic Systems,* vol. 47, no. 2, pp. 1101-1113, 2011.

[7] J. Breßler, P. Reisdorf, M. Obst, and G. Wanielik, "GNSS positioning in non-line-of-sight context—A survey," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, 2016, pp. 1147-1154: IEEE.

[8] C. Pinana-Diaz, R. Toledo-Moreo, D. Betaille, and A. F. Gomez-Skarmeta, "GPS multipath detection and exclusion with elevation-enhanced maps," in *Intelligent transportation systems (ITSC), 2011 14th International IEEE Conference on*, 2011, pp. 19-24: IEEE.

[9] S. Peyraud *et al.*, "About non-line-of-sight satellite detection and exclusion in a 3D map-aided localization algorithm," *Sensors,* vol. 13, no. 1, pp. 829-847, 2013.

[10] W. Wen, G. Zhang, and L.-T. Hsu, "Correcting GNSS NLOS by 3D LiDAR and Building Height," presented at the ION GNSS+, 2018, Miami, Florida, USA., 2018.

[11] L. Wang, P. D. Groves, and M. K. Ziebart, "Urban positioning on a smartphone: Real-time shadow matching using GNSS and 3D city models," 2013: The Institute of Navigation.

[12] L. Wang, P. D. Groves, and M. K. Ziebart, "GNSS shadow matching: Improving urban positioning accuracy using a 3D city model with optimized visibility scoring scheme," *Navigation,* vol. 60, no. 3, pp. 195-207, 2013.

[13] L. Wang, P. D. Groves, and M. K. Ziebart, "Smartphone shadow matching for better cross-street GNSS positioning in urban environments," *The Journal of Navigation,* vol. 68, no. 3, pp. 411-433, 2015.

[14] P. D. Groves and M. Adjrad, "Likelihood-based GNSS positioning using LOS/NLOS predictions from 3D mapping and pseudoranges," *GPS Solutions,* journal article vol. 21, no. 4, pp. 1805-1816, October 01 2017.

[15] M. Adjrad and P. D. Groves, "Intelligent Urban Positioning: Integration of Shadow Matching with 3D-Mapping-Aided GNSS Ranging," *Journal of Navigation,* vol. 71, no. 1, pp. 1-20, 2018.

[16] L.-T. Hsu, Y. Gu, and S. Kamijo, "3D building model-based pedestrian positioning method using GPS/GLONASS/QZSS and its reliability calculation," (in English), *GPS Solutions*, vol. 20, no. 3, pp. 413–428, 2016.

[17] S. Miura, L.-T. Hsu, F. Chen, and S. Kamijo, "GPS error correction with pseudorange evaluation using three-dimensional maps," *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, no. 6, pp. 3104-3115, 2015.

[18] M. Obst, S. Bauer, P. Reisdorf, and G. Wanielik, "Multipath detection with 3D digital maps for robust multi-constellation GNSS/INS vehicle localization in urban areas," in *Intelligent Vehicles Symposium (IV), 2012 IEEE,* 2012, pp. 184-190: IEEE.

[19] T. Suzuki and N. Kubo, "Correcting GNSS multipath errors using a 3D surface model and particle filter," *Proc. ION GNSS+ 2013,* 2013.

[20] H.-F. Ng, G. Zhang, and L.-T. Hsu, "GNSS NLOS Pseudorange Correction based on Skymask for Smartphone Applications," presented at the ION GNSS+, 2019, Miami, Florida, USA, 2019.

[21] J. Levinson and S. Thrun, "Robust Vehicle Localization in Urban Environments Using Probabilistic Maps," (in English), *2010 Ieee International Conference on Robotics and Automation (Icra),* pp. 4372-4378, 2010.

[22] Y. Gu, L.-T. Hsu, and S. Kamijo, "GNSS/onboard inertial sensor integration with the aid of 3-D building map for lane-level vehicle self-localization in urban canyon," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 6, pp. 4274-4287, 2016.

[23] A. Fernandez *et al.,* "GNSS/INS/LiDAR integration in urban environment: Algorithm description and results from ATENEA test campaign," in *Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing,(NAVITEC), 2012 6th ESA Workshop on*, 2012, pp. 1-8: IEEE.

[24] L.-T. Hsu, H. Tokura, N. Kubo, Y. Gu, and S. Kamijo, "Multiple faulty GNSS measurement exclusion based on consistency check in urban canyons," *IEEE Sensors Journal,* vol. 17, no. 6, pp. 1909-1917, 2017.

[25] Y. Suh and R. Shibasaki, "Evaluation of satellite-based navigation services in complex urban environments using a three-dimensional GIS," *IEICE transactions on communications,* vol. 90, no. 7, pp. 1816-1825, 2007.

[26] Y.-W. Lee, Y.-C. Suh, and R. Shibasaki, "A simulation system for GNSS multipath mitigation using spatial statistical methods," *Computers & Geosciences,* vol. 34, no. 11, pp. 1597-1609, 2008.

[27] L.-T. Hsu, F. Chen, and S. Kamijo, "Evaluation of multi-GNSSs and GPS with 3D map methods for pedestrian positioning in an urban canyon environment," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences,* vol. 98, no. 1, pp. 284-293, 2015.

[28] P. D. Groves, "Shadow matching: A new GNSS positioning technique for urban canyons," *The journal of Navigation,* vol. 64, no. 3, pp. 417-430, 2011.

[29] L. Wang, P. D. Groves, and M. K. Ziebart, "Multi-constellation GNSS performance evaluation for urban canyons using large virtual reality city models," *The Journal of Navigation,* vol. 65, no. 3, pp. 459-476, 2012.

[30] L.-T. Hsu, Y. Gu, and S. Kamijo, "3D building model-based pedestrian positioning method using GPS/GLONASS/QZSS and its reliability calculation," *GPS solutions,* vol. 20, no. 3, pp. 413-428, 2016.

[31] L.-T. Hsu, Y. Gu, Y. Huang, and S. Kamijo, "Urban pedestrian navigation using smartphone-based dead reckoning and 3-D map-aided GNSS," *IEEE Sensors Journal,* vol. 16, no. 5, pp. 1281-1293, 2016.

[32] M. Adjrad and P. D. Groves, "Intelligent Urban Positioning: Integration of Shadow Matching with 3D-Mapping-Aided GNSS Ranging," *The Journal of Navigation,* vol. 71, no. 1, pp. 1-20, 2018.

[33] M. Adjrad and P. D. Groves, "Enhancing least squares GNSS positioning with 3D mapping without accurate prior knowledge," *Navigation,* vol. 64, no. 1, pp. 75-91, 2017.

[34] D. Betaille, F. Peyret, M. Ortiz, S. Miquel, and F. Godan, "Improving accuracy and integrity with a probabilistic Urban Trench modeling," *Navigation,* vol. 63, no. 3, pp. 283-294, 2016.

[35] D. Maier and A. Kleiner, "Improved GPS Sensor Model for Mobile Robots in Urban Terrain," (in English)*, 2010 Ieee International Conference on Robotics and Automation (Icra),* pp. 4385-4390, 2010.

[36] D. Maier and A. Kleiner, "Improved GPS sensor model for mobile robots in urban terrain," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 4385-4390: IEEE.

[37] K. Ali, X. Chen, F. Dovis, D. De Castro, and A. J. Fernández, "GNSS signal multipath error characterization in urban environments using LiDAR data aiding," in *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on*, 2012, pp. 1-5: IEEE.

[38] J. I. Meguro, T. Murata, J. I. Takiguchi, Y. Amano, and T. Hashizume, "GPS multipath mitigation for urban area using omnidirectional infrared camera," *IEEE Transactions on Intelligent Transportation Systems,* vol. 10, no. 1, pp. 22-30, 2009.

[39] T. Suzuki, M. Kitamura, Y. Amano, and T. Hashizume, "High-accuracy GPS and GLONASS positioning by multipath mitigation using omnidirectional infrared camera," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 311-316.

[40] O. O. Okunribido, S. J. Shimbles, M. Magnusson, and M. Pope, "City bus driving and low back pain: A study of the exposures to posture demands, manual materials handling and whole-body vibration," (in English), *Applied Ergonomics,* vol. 38, no. 1, pp. 29-38, Jan 2007.

[41] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, 2011, pp. 163-168: IEEE.

[42] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013, pp. 763-770: IEEE.

[43] M. Himmelsbach, F. V. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3d point clouds for ground vehicles," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 2010, pp. 560-565: IEEE.

[44] W. Li, Q. Guo, M. K. Jakubowski, and M. Kelly, "A new method for segmenting individual trees from the lidar point cloud," *Photogrammetric Engineering & Remote Sensing,* vol. 78, no. 1, pp. 75-84, 2012.

[45] B. Douillard *et al.*, "On the segmentation of 3D LIDAR point clouds," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 2798-2805: IEEE.

[46] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 221-228: IEEE.

[47] J. Zhang, X. Lin, and X. Ning, "SVM-based classification of segmented airborne LiDAR point clouds in urban areas," *Remote Sensing,* vol. 5, no. 8, pp. 3749-3775, 2013.

[48] J. Marshall, "Creating and viewing skyplots," *GPS solutions,* vol. 6, no. 1-2, pp. 118-120, 2002.

[49] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz,* vol. 24, no. 4, pp. 345-348, 2010.

[50] G. Barequet and S. Har-Peled, "Efficiently approximating the minimum-volume bounding box of a point set in three dimensions," *Journal of Algorithms,* vol. 38, no. 1, pp. 91-109, 2001.

[51] A. M. Herrera, H. F. Suhandri, E. Realini, M. Reguzzoni, and M. C. J. G. s. de Lacy, "goGPS: open-source MATLAB software," vol. 20, no. 3, pp. 595-603, 2016.

[52] C. Rödel, S. Stadler, A. Meschtscherjakov, and M. Tscheligi, "Towards autonomous cars: the effect of autonomy levels on acceptance and user experience," in *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 2014, pp. 1-8: ACM.

[53] Y. Gao, S. Liu, M. M. Atia, and A. Noureldin, "INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm," *Sensors,* vol. 15, no. 9, pp. 23286-23302, 2015.

[54] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 4372-4378: IEEE.

[55] X. Meng, H. Wang, and B. Liu, "A Robust Vehicle Localization Approach Based on GNSS/IMU/DMI/LiDAR Sensor Fusion for Autonomous Vehicles," *Sensors,* vol. 17, no. 9, p. 2140, 2017.

[56] A. Fernández *et al.*, "ATENEA: Advanced techniques for deeply integrated GNSS/INS/LiDAR navigation," in *Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 2010 5th ESA Workshop on*, 2010, pp. 1-8: IEEE.

[57] A. Fernández, P. Silva, and I. Colomina, "Real-time navigation and mapping with mobile mapping systems using LiDAR/Camera/INS/GNSS advanced hybridization algorithms: description and test results," in *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2014,* 2014, pp. 896-903.

[58] P. D. Groves, "Shadow Matching: A New GNSS Positioning Technique for Urban Canyons," *The Journal of Navigation,* vol. 64, no. 03, pp. 417-430, 2011.

[59] L. Wang, P. D. Groves, and M. K. Ziebart, "GNSS Shadow Matching: Improving Urban Positioning Accuracy Using a 3D City Model with Optimized Visibility Scoring Scheme," *NAVIGATION, Journal of The Institute of Navigation,* vol. 60, no. 3, pp. 195-207, 2013.

[60] J. S. Sánchez, A. Gerhmann, P. Thevenon, P. Brocard, A. B. Afia, and O. Julien, "Use of a FishEye camera for GNSS NLOS exclusion and characterization in urban environments," in *ION ITM 2016, International Technical Meeting*, 2016: ION.

[61] A. Shetty and G. X. Gao, "Covariance Estimation for GPS-LiDAR Sensor Fusion for UAVs," in *ON GNSS+ 2017, Portland, OR, USA*, 2017.

[62] J. Levinson, M. Montemerlo, and S. Thrun, "Map-Based Precision Vehicle Localization in Urban Environments," in *Robotics: Science and Systems*, 2007, vol. 4, p. 1: Citeseer.

[63] M. Greenspan and M. Yurick, "Approximate kd tree search for efficient ICP," in *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, 2003, pp. 442-448: IEEE.

[64] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.

[65] A. M. Herrera, H. F. Suhandri, E. Realini, M. Reguzzoni, and M. C. de Lacy, "goGPS: open-source MATLAB software," *GPS solutions,* vol. 20, no. 3, pp. 595-603, 2016.

[66] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, 2009, vol. 3, no. 3.2, p. 5: Kobe, Japan.

[67] 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 *Position, Location and Navigation Symposium (PLANS), 2018 IEEE/ION*, 2018, pp. 158-165: IEEE.

[68] W. Wen, L.-T. Hsu, and G. Zhang, "Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong," *Sensors,* vol. 18, no. 11, p. 3928, 2018.

[69] G. Wan, X. Yang, R. Cai, H. Li, H. Wang, and S. Song, "Robust and Precise Vehicle Localization based on Multi-sensor Fusion in Diverse City Scenes," 2017.

[70] S. Thrun, "Probabilistic robotics," *Communications of the ACM,* vol. 45, no. 3, pp. 52-57, 2002.

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

[72] X. Zhang, L. Miao, and H. Shao, "Tracking Architecture Based on Dual-Filter with State Feedback and Its Application in Ultra-Tight GPS/INS Integration," *Sensors,* vol. 16, no. 5, p. 627, 2016.

[73] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter based SLAM," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 473-479: IEEE.

[74] J. Zhu, N. Zheng, Z. Yuan, Q. Zhang, X. Zhang, and Y. He, "A SLAM algorithm based on the central difference Kalman filter," in *Intelligent Vehicles Symposium, 2009 IEEE*, 2009, pp. 123-128: IEEE.

[75] C. Cadena and J. Neira, "SLAM in O (logn) with the Combined Kalman-Information Filter," *Robotics and Autonomous Systems,* vol. 58, no. 11, pp. 1207-1219, 2010.

[76] S. Thrun and Y. Liu, "Multi-robot SLAM with sparse extended information filers," in *Robotics Research. The Eleventh International Symposium*, 2005, pp. 254-266: Springer.

[77] J.-S. Gutmann, E. Eade, P. Fong, and M. Munich, "A constant-time algorithm for vector field slam using an exactly sparse extended information filter," *Robotics: Science and Systems VI,* p. 193, 2011.

[78] D. Törnqvist, T. B. Schön, R. Karlsson, and F. Gustafsson, "Particle filter SLAM with high dimensional vehicle model," *Journal of Intelligent and Robotic Systems,* vol. 55, no. 4-5, pp. 249-266, 2009.

[79] N. Fairfield, G. Kantor, and D. Wettergreen, "Towards particle filter SLAM with three dimensional evidence grids in a flooded subterranean environment," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 3575-3580: IEEE.

[80] R. Sim, P. Elinas, and J. J. Little, "A study of the Rao-Blackwellised particle filter for efficient and accurate vision-based SLAM," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 303-318, 2007.

[81] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[82] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181-1203, 2006.

[83] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous robots*, vol. 4, no. 4, pp. 333-349, 1997.

[84] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 2262-2269: IEEE.

[85] T. Sakai, K. Koide, J. Miura, and S. Oishi, "Large-scale 3D outdoor mapping and on-line localization using 3D-2D matching," in *System Integration (SII), 2017 IEEE/SICE International Symposium on*, 2017, pp. 829-834: IEEE.

[86] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31-43, 2010.

[87] P. François, B. David, and M. Florian, "Non-Line-Of-Sight GNSS signal detection using an on-board 3D model of buildings," in *ITS Telecommunications (ITST), 2011 11th International Conference on*, 2011, pp. 280-286: IEEE.

[88] F. Peyret, D. Bétaille, P. Carolina, R. Toledo-Moreo, A. F. Gómez-Skarmeta, and M. Ortiz, "Gnss autonomous localization: Nlos satellite detection based on 3-d maps," *IEEE Robotics & Automation Magazine*, vol. 21, no. 1, pp. 57-63, 2014.

[89] D. Bétaille, F. Peyret, and M. Voyer, "Applying standard digital map data in map-aided, lane-level GNSS location," *The Journal of navigation*, vol. 68, no. 5, pp. 827-847, 2015.

[90] M. Hentschel, O. Wulf, and B. Wagner, "A GPS and laser-based localization for urban and non-urban outdoor environments," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 149-154: IEEE.

[91] A. P. Shetty, "GPS-LiDAR sensor fusion aided by 3D city models for UAVs," 2017.

[92] 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," *arXiv preprint arXiv:1804.10917*, 2018.

[93] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, 2014, vol. 2.

[94] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm," *Image and Vision Computing*, vol. 23, no. 3, pp. 299-309, 2005.

[95] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *Journal of Field Robotics,* vol. 24, no. 10, pp. 803-827, 2007.

[96] M. Magnusson, "The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection," Örebro universitet, 2009.

[97] M. Magnusson, H. Andreasson, A. Nuchter, and A. J. Lilienthal, "Appearance-based loop detection from 3D laser data using the normal distributions transform," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 2009, pp. 23-28: IEEE.

[98] *GNSS Satellites system*. Available: https://www.gislounge.com/global-navigation-satellite-systems/

[99] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 3607-3613: IEEE.

[100] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.

[101] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354-3361: IEEE.

[102] T. Pfeifer and P. Protzel, "Expectation-maximization for adaptive mixture models in graph optimization," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3151-3157: IEEE.

[103] T. Pfeifer and P. Protzel, "Robust sensor fusion with self-tuning mixture models," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3678-3685: IEEE.

[104] T. Pfeifer, S. Lange, and P. Protzel, "Dynamic Covariance Estimation—A parameter free approach to robust Sensor Fusion," in *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2017, pp. 359-365: IEEE.

[105] N. Sünderhauf and P. Protzel, "Towards robust graphical models for GNSS-based localization in urban environments," in *International Multi-Conference on Systems, Signals & Devices*, 2012, pp. 1-6: IEEE.

[106] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends® in Robotics,* vol. 6, no. 1-2, pp. 1-139, 2017.

[107] N. Sünderhauf, M. Obst, S. Lange, G. Wanielik, and P. Protzel, "Switchable constraints and incremental smoothing for online mitigation of non-line-of-sight and multipath effects," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 262-268: IEEE.

[108] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[109] R. M. Watson and J. N. Gross, "Robust navigation in GNSS degraded environment using graph optimization," *arXiv preprint arXiv:1806.08899,* 2018.

[110] R. M. Watson, J. N. Gross, C. N. Taylor, and R. C. Leishman, "Uncertainty Model Estimation in an Augmented Data Space for Robust State Estimation," *arXiv preprint arXiv:1908.04372,* 2019.

[111] R. M. Watson, J. N. Gross, C. N. Taylor, and R. C. Leishman, "Robust Incremental State Estimation through Covariance Adaptation," *arXiv preprint arXiv:1910.05382,* 2019.

[112] S. Kato, M. Kitamura, T. Suzuki, and Y. Amano, "Nlos satellite detection using a fish-eye camera for improving gnss positioning accuracy in urban area," *Journal of robotics and mechatronics*, vol. 28, no. 1, pp. 31-39, 2016.

[113] P. V. Gakne, "Improving the Accuracy of GNSS Receivers in Urban Canyons using an Upward-Facing Camera," PhD Thesis, Geomatics Engineering, University of Calgary. doi: dx. doi. org …, 2018.

[114] P. V. Gakne and K. O'Keefe, "Tightly-Coupled GNSS/Vision Using a Sky-Pointing Camera for Vehicle Navigation in Urban Areas," *Sensors,* vol. 18, no. 4, p. 1244, 2018.

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

[116] B. X. Wen W., Kan Y.C., Hsu, L.T, "Tightly Coupled GNSS/INS Integration Via Factor Graph and Aided by Fish-eye Camera (Accepted)," *IEEE Transactions on Vehicular Technology,* 2019.

[117] X. Bai, W. Wen, G. Zhang, and L.-T. Hsu, "Real-time GNSS NLOS Detection and Correction Aided by Sky-Pointing Camera and 3D LiDAR," presented at the Proceedings of ION Pacific PNT 2019, Honolulu, HA, USA, 2019.

[118] W. Wen, Y.-C. Kan, and L.-T. Hsu, "Performance Comparison of GNSS/INS Integrations Based on EKF and Factor Graph Optimization," presented at the ION GNSS+ 2019, Florida, 2019.

[119] F. Moosmann and C. Stiller, "Velodyne slam," in *2011 ieee intelligent vehicles symposium (iv),* 2011, pp. 393-398: IEEE.

[120] L.-T. Hsu, Y. Gu, and S. Kamijo, "NLOS correction/exclusion for GNSS measurement using RAIM and city building models," *Sensors,* vol. 15, no. 7, pp. 17329-17349, 2015.

[121] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE robotics & automation magazine,* vol. 13, no. 3, pp. 108-117, 2006.

[122] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, 2014, vol. 2, p. 9.

[123] 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.

[124] J. D. Jurado and J. F. Raquet, "Towards an online sensor model validation and estimation framework," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS),* 2018, pp. 1319-1325: IEEE.

[125] W. Wen, L.-T. Hsu, and G. J. S. Zhang, "Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong," vol. 18, no. 11, p. 3928, 2018.

[126] A. S. Glassner, *An introduction to ray tracing*. Elsevier, 1989.

[127] S. Tay and J. Marais, "Weighting models for GPS Pseudorange observations for land transportation in urban canyons," in *6th European Workshop on GNSS Signals and Signal Processing*, 2013, p. 4p.

[128] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Factor graph based incremental smoothing in inertial navigation systems," in *2012 15th International Conference on Information Fusion*, 2012, pp. 2154-2161: IEEE.

[129] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research,* vol. 31, no. 2, pp. 216-235, 2012.

[130] N. Sünderhauf, M. Obst, G. Wanielik, and P. Protzel, "Multipath mitigation in GNSS-based localization using robust optimization," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 784-789: IEEE.

[131] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Institute of Technology2012.

[132] G. Falco, M. Pini, and G. Marucco, "Loose and tight GNSS/INS integrations: Comparison of performance assessed in real urban scenarios," *Sensors,* vol. 17, no. 2, p. 255, 2017.

[133] J. Jiao, Y. Yu, Q. Liao, H. Ye, and M. Liu, "Automatic Calibration of Multiple 3D LiDARs in Urban Environments," *arXiv preprint arXiv:1905.04912,* 2019.

[134] C. M. Silva, B. M. Masini, G. Ferrari, and I. Thibault, "A survey on infrastructure-based vehicular networks," *Mobile Information Systems,* vol. 2017, 2017.