ACCURATE AND SAFETY-QUANTIFIABLE LOCALIZATION METHODS FOR AUTONOMOUS ROBOTIC SYSTEMS

XI ZHENG

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University

Department of Aeronautical and Aviation Engineering

Accurate and Safety-quantifiable Localization Methods for Autonomous Robotic Systems

Xi ZHENG

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

May 2025

Certificate of Originality

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

________________________(Signature)

Xi Zheng (Name of student)

Abstract

Accurate and safety-quantifiable localization is critical for safety-critical autonomous robotic systems, such as unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs). While visual-inertial odometry (VIO) provides high-frequency pose estimation, it inherently suffers from cumulative drift. Meanwhile, current pose estimation approaches for localization predominantly rely on local nonlinear optimization techniques, which lack guarantees of global optimality and are vulnerable to outliers. Additionally, a safety-critical robotic system is required not only to achieve accurate pose estimation but also to predict and quantify the associated localization uncertainty by providing an error bound for each pose. These limitations severely compromise the accuracy, reliability, and safety of localization, despite its pivotal role in safety-critical applications.

To address these challenges, this thesis develops a comprehensive three-stage framework for robust and safety-quantifiable localization. The first study introduces a novel line feature-based visual localization method that leverages prior 3D maps while incorporating safety quantification strategy. Building upon VIO for high-frequency local pose initialization, the proposed method establishes geometric constraints between 2D image lines and 3D lines belonging prior map through an innovative foot-point based association. The safety quantification mechanism, inspired by global navigation satellite systems (GNSS) receiver autonomous integrity monitoring (RAIM) methodology, employs a weighted sum of squares residual with

Chi-squared testing for outlier rejection, complemented by a derived protection level scheme that bounds potential errors in both position and orientation domains.

The second study advances the first work through a tightly-coupled factor graph formulation that fully integrates 3D prior line maps with VIO measurements, addressing the complementarity limitations of loosely-coupled approaches. This framework incorporates a robust cross-modality line association model between 3D maps and 2D images, enhanced by an efficient line-tracking strategy for outlier rejection. A theoretically grounded line feature cost model is developed for factor graph optimization, supported by the first rigorous observability analysis of such systems. The analysis reveals full observability of global translations with only the yaw angle around gravity remaining unobservable, providing critical insights for safety-critical applications.

The final study overcomes fundamental limitations of local optimization methods by adopting a global estimation framework that unifies outlier rejection with estimation error prediction. The solution employs Truncated Least Squares (TLS) for robust estimation, with the non-convexity challenge addressed through a Graduated Non-Convexity (GNC) strategy combined with Gaussian-Newton optimization. This approach systematically approximates and solves progressively more challenging subproblems to converge to globally optimal solutions. Furthermore, the framework incorporates an error propagation model that explicitly accounts for measurement uncertainties and residual outliers, enabling predictive error bounding to guarantee estimation safety throughout the localization process.

This integrated three-stage approach systematically addresses the accuracy, robustness, and safety quantification challenges in modern localization systems, while providing theoretical guarantees and practical implementation strategies suitable for safety-critical robotic applications. The feasibility and practicality of the proposed frameworks have been validated through simulations and real-world datasets. To benefit the research community, we have open-sourced the complete implementation, including algorithms and datasets.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my chief supervisor Dr. Weisong Wen and co-supervisor Dr. Li-Ta Hsu, for their invaluable guidance, patience, and encouragement throughout this research journey. Their expertise, critical feedback, and unwavering support were instrumental in shaping this thesis.

I extend my sincere appreciation to my thesis committee members, Professor Hui Kong from University of Macau and Professor Kedong Wang from Beihang University for their insightful suggestions and constructive criticism, which significantly improved the quality of this work. I am deeply appreciate the valuable time they generously provided to review and examine my thesis.

I would like to give my hearty thanks to the Associate Professor Fei Gao from Zhejiang University for providing me the opportunity to exchange and cooperation.

I am indebted to My BoE Chair Professor Wei Liu, the Department of Aeronautical and Aviation Engineering, and the Hong Kong Polytechnic University for their support and for providing the research environment and resources essential to my work.

My thanks go to Dr. Guohao Zhang, Dr. Xiwei Bai, Dr. Feng Huang, Dr. Hoi-Fung Ng and Dr. Penghui Xu for giving me valuable advices to help me through my PhD journey. I would also like to thank Dr. Hongmin Zhang, Liyuan Zhang, Ruizhi Hu, Xikun Liu, Xiao Xia, Yihan Zhong, Zekun Zhang, Ruijie Xu, Xin Wang, Yuanyuan Zhang, Yuan Li and all of you in the lab, for the unforgettable time we had together. I wish you all the best for the future.

Finally, I owe my deepest appreciation to two pillars of my life. To my partner, Dr. Xiangyong Wen, for walking beside me through every step of this journey. Your unwavering love, support, and understanding have been my greatest treasures. Together we've weathered life's joys and challenges, and no one in this world knows me better than you. For all this and more, I will always be with you. To my parents for their unwavering care and selfless devotion throughout my life. Their unconditional support for all my decisions has been my greatest source of strength, and their exemplary lives have served as my guiding inspiration. They remain my most cherished role models, and my lifelong aspiration is to reciprocate their care.

List of Publications

Journal

[1] **X. Zheng**, W. Wen, and L.-T. Hsu, "Safety-quantifiable line feature-based monocular visual localization with 3d prior map," *IEEE Transactions on Intelligent Transportation Systems,* 2025. (Project)

[2] **X. Zheng**, W. Wen, and L.-T. Hsu, "Tightly-couple visual/inertial/map integration with observability analysis for reliable localization of intelligent vehicles," IEEE Transactions on Intelligent Vehicles, 2024 (Project)

[3] **X. Zheng**, X. Wen, W. Wen, F. Gao, and L. -T. Hsu, "A Global and Safety-Guaranteed State Estimation Method for Spatial Transformation," *IEEE Transactions on Robotics* (in submission).

Conference

[1] **X. Zheng**, W. Wen, and L.-T. Hsu, Tightly-coupled line feature-aided visual inertial localization within lightweight 3d prior map for intelligent vehicles," in 2023 IEEE 26th Inter-national Conference on Intelligent Transportation Systems (ITSC). IEEE, 2023, pp. 6019–6026

[2] X. Wen, Y. Wang, **X. Zheng**, K. Wang, C. Xu, and F. Gao, "Simultaneous time synchronization and mutual localization for multi-robot system," in 2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024, pp. 2603–2609

[3] X. Wen, X. Huang, Y. Wang, W. Wen, **X. Zheng***, F. Gao, " Multi-UAV

Planner for Dynamic Target Localization by Maximizing Common Field-of-View," in *2025 IEEE International Conference on Intelligent Robots and Systems (IROS)*. (Corresponding author, in submission).

Symbols And Acronyms

Symbols and Acronyms

Symbols

$(\cdot)^w$	The world frame, the origin coincides with the prior map starting point
$(\cdot)^b$	The body frame of robots, the origin can be defined on the inertial measurement unit (IMU)
$(\cdot)^c$	The camera frame
$\boldsymbol{x}$	The state variables
$\mathcal{X}$	The state domain
$\mathcal{M}$	The measurements set
$\mathcal{I}$	The inlier measurement set
s	The scale of robots
$\boldsymbol{R}$	The rotation matrix of robots, the rotation can also be expressed by quaternion $\boldsymbol{q}$
$\boldsymbol{t}$	The translation vector of robots the translation also be expressed by $\boldsymbol{p}$
$\boldsymbol{R}_b^a$	The rotation from frame b to frame a
$\boldsymbol{p}_b^a$	The translation from frame b to frame a
$\boldsymbol{\xi}$	The Lie algebra of translation and rotation $[\boldsymbol{R}\|\boldsymbol{t}]$
$\boldsymbol{K}$	The camera intrinsic parameter

$\boldsymbol{P}(x,y,z)$	A 3D point in space
$\boldsymbol{p}(\mu,\nu)$	A 2D pixel point in image
$\boldsymbol{b}_a$	IMU accelerometer bias
$\boldsymbol{b}_g$	gyroscope bias
$\otimes$	The symbol of the multiplication operation of quaternions
χ^2	The notation of Chi-Squared distribution
$\mathcal{N}$	The notation of Gaussian distribution
$\lVert\cdot\rVert^2$	The 2-norm of a vector or matrix in Euclidean space
$\boldsymbol{r}$	The cost function in optimization problem
$\rho(\cdot)$	The loss function in optimization problem

Acronyms

UGV	unmanned ground vehicle
UAV	unmanned aerial vehicl
GNSS	global navigation satellite system
LiDAR	light detection and ranging
IMU	inertial measurement unit
VIO	visual-inertial odometry
HD	high-definition
SLAM	simultaneous localization and mapping
VINS	visual-inertial navigation system
FGO	factor graph optimization
GT	ground truth
QCQP	quadratically constrained quadratic program
SDP	semidefinite programming
SOS	sum-of-squares
GNC	graduated non-convexity
RANSAC	random sample consensus
RAIM	receiver autonomous integrity monitoring
FDE	fault detection and exclusion
PL	protection level
TLS	truncated least squares
BA	bundle adjustment
ICP	iterative closest point
NDT	distribution transform
DoF	degree of freedom
GMM	gaussian mixture model

EKF	extended Kalman filter
LM	Levenberg–Marquardt
GN	Gauss-Newton
BnB	branch-and-bound
MC	maximum consensus
WLS	weighted least squares
FoV	field of view
FLD	fast line detection
BOP	The benchmark of 6D pose estimation
CV	computer vision
ROS	Robot Operating System
RMSE	Root Mean Square Error
ATE	Absolute Translation Error

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation & Overview

Autonomous robotic systems are increasingly deployed in complex, dynamic, and safety-critical environments, where both accurate localization and reliable safety assurance are indispensable [3]. While modern localization systems—including global navigation satellite system (GNSS), LiDAR, camera, and their combinations—have achieved substantial progress, yet they continue to face significant challenges in practice. For instance, GNSS signals can be degraded or blocked by urban canyons or indoor structures, resulting in intermittent or inaccurate positioning [4]. LiDAR-based methods, while offering high-precision 3D mapping, are expensive and generate large volumes of data that impose heavy computational and storage burdens [5]. Vision-based sensors provide rich texture and semantic information, but are susceptible to illumination changes, motion blur, and low-texture regions [6]. Even advanced multi-sensor fusion approaches, although improving robustness, still face challenges such as cross-modality alignment and heterogeneous measurement integration [7]. These limitations not only reduce localization accuracy but also introduce risks that cannot be ignored in safety-critical applications.

Consequently, current localization frameworks often focus primarily on improving accuracy, with limited mechanisms to quantify the reliability or safety of the estimated states. In safety-critical domains such as aviation, localization is treated as a safety-of-life function: systems continuously monitor estimation errors, detect faults, and provide explicit upper bounds on worst-case errors (e.g., protection levels in GNSS) [8]. In contrast, most robotic perception systems lack comparable mechanisms, leaving a gap between high-precision localization and verifiable safety assurance. Without explicit quantification of uncertainty and robust outlier handling, autonomous robots are vulnerable to sensor faults, environmental disturbances, and modeling mismatches, which can lead to unsafe behavior or accidents in real-world deployment.

These challenges motivate the need for a unified framework that addresses accurate localization, outlier rejections, and real-time safety quantification. Accordingly, this thesis explores the following directions. (1) Improving localization accuracy through multi-modal constraints, including visual–inertial estimation, geometric information from prior maps, and globally optimization strategies such as GNC-based method [9]. (2) Safety-aware state estimation, which combines robust outlier rejection mechanisms—such as chi-squared tests, robust loss functions, and dynamic outlier filtering—with analytical error propagation analysis to derive conservative bounds on estimation uncertainty [10]. This approach enables the system to mitigate the impact of sensor faults, mismatches, and environmental disturbances, while simultaneously predicting potential errors in its estimated states to quantify and ensure the safety of the localization system in real time. Together, these contributions aim to support autonomous systems that are not only accurate and robust, but also verifiably safe under real-world uncertainties.

1.2 Localization Solutions

Accurate, cost-effective, and reliable localization is the fundamental function of safety-critical autonomous robotic systems, such as unmanned ground vehicles (UGVs) [11, 12] and unmanned aerial vehicles (UAVs) [13, 14]. By employing perception and state estimation, the localization module delivers position and orientation information regarding the robot's ego-motion within its environment. This information underpins essential functions such as mapping, planning, and control, which are critical for the robot to autonomously, efficiently, and safely execute its assigned tasks. The global navigation satellite system (GNSS) [15] is widely used for providing globally-referenced positioning for autonomous systems in outdoor environments. However, GNSS signals are susceptible to blockage and multipath reflection from urban environments and natural obstructions, such as tall buildings and dense tree canopies, which can substantially degrade positioning accuracy [4]. Furthermore, GNSS signals are nearly ineffective in indoor environments. The aforementioned limitations significantly restrict the applicability of GNSS in various scenarios. Consequently, alternative sensors, such as the point cloud-based light detection and ranging (LiDAR) [16, 17] and vision-based camera [18, 19], have become indispensable components in autonomous robotic systems.

Although LiDAR and vision-based systems have achieved exceptional performance in robotic localization, their intrinsic sensor limitations restrict deployment across diverse operational environments. LiDAR sensors provide dense 3D point cloud that enable high-precision pose estimation through point cloud registration algorithms [20]. However, single-modality LiDAR point clouds suffer from limited color and semantic information and can impose significant computational burdens due to their size. More critically, LiDAR-only localization may fail in geometrically degenerate environments, such as featureless or repetitive scenes. Integrating vision with LiDAR provides complementary texture and semantic cues, improving the robustness and accuracy of the localization system [1]. Furthermore, the susceptibility of visual sensors to illumination variations and their tendency to fail in low-texture environments substantially limit their applicability [21, 22]. With the increasing expectation of automation, intelligence, and generalization of robotic systems, single-sensor solutions have become inadequate for reliable localization in changing environments. To address this challenge, multi-sensor fusion—combining LiDAR, camera, and inertial measurement unit (IMU)—has attracted increasing attention. This approach leverages complementary sensor strengths to compensate for individual shortcomings, enabling robust and high-precision pose estimation [23].

To mitigate the computational burden associated with processing real-time LiDAR point clouds and reduce the payload demands of heavy LiDAR systems during robot operation, some researchers have turned their attention to an alternative fusion approach that utilizes pre-built 3D point cloud maps as prior information instead of real-time LiDAR data. This fusion of 3D prior map, vision, and IMU localization system is called 3D prior map-aided visual-inertial odometry (VIO), which is a hybrid localization framework that enhances traditional VIO systems by incorporating pre-built 3D environmental maps as spatial constraints. This approach aligns real-time visual features and IMU measurements with the prior map information, mitigating cumulative drift inherent in pure VIO while reducing computational and payload demands compared to real-time LiDAR processing. It achieves robust, globally consistent pose estimation in GNSS-denied or perceptually degraded environments by leveraging offline map priors [24, 25, 26, 27].

Nevertheless, this solution still faces several challenges that demand further investigation. One limitation is the substantial storage requirement imposed by dense prior maps, which can hinder deployment in resource-constrained systems. Additionally, the reliability of localization critically depends on the accuracy and consistency of the prior map—any degradation or outdated information in the map may lead to significant localization errors. To address these issues, researchers have explored high-definition (HD) maps as a promising alternative. These maps offer high-level precision and rich semantic annotations, including detailed road geometry, lane markings, and traffic signs, thereby enhancing both localization robustness and scene understanding [28, 29]. On the other hand, integrating real-time visual features with 3D prior maps introduces another layer of complexity due to the heterogeneous measurements from LiDAR and cameras. The differences in data representation (3D points vs 2D images) and inherent sensor noise complicate the cross-modality alignment problem between the two data sources [30].

To fill the gap, some studies leverage mutual geometric line features between 2D images and 3D maps for cross-modality registration. Originally adopted in visual simultaneous localization and mapping (SLAM), line features serve as supplementary correspondences for motion estimation [31, 32, 33]. Line features provide distinct advantages in low-texture environments where point-based features (e.g., corners) are scarce, as they robustly encode structural geometry [34]. Additionally, lines usually offer more stable constraints than points in both 3D space and 2D images, since they consist of multiple point correspondences [35]. Studies such as [31, 32] demonstrate that line features can effectively exploit the potential of the geometry information within the environment. Building on this, the work in [36] introduces a line feature-based visual localization method that associates 2D image lines with 3D map lines extracted from prior point, with the initial guess of the pose from the visual-inertial navigation system (VINS) [1]. However, the approach in [36] relies on line-to-line distance constraints, which do not fully utilize the geometric relationships between features.

The first work in this study presents an improved line feature-based visual localization framework that combines robust point-to-point line-based geometric constraints with a selective correspondence matching strategy. By loosely coupling VIO state estimates with line-based map-matching constraints in a factor graph optimization (FGO), the method achieves reliable localization within prior maps [3]. Nevertheless, the shortcomings of loosely coupled frameworks are obvious. (1) The sequential processing (from VIO to prior map matching) leads to cumulative error propagation, since the drift in VIO state estimation directly influences the subsequent map alignment stage. (2) As the prior map matching depends on the VIO-estimated pose as initialization, the accumulated drift also degrades the effectiveness of geometric associations between 2D image features and 3D prior map features, thereby limiting the achievable accuracy improvement from cross-modal constraints. (3) The overall system remains sensitive to initialization errors, particularly in challenging scenarios with poor visual features or illumination disturbances, where the VIO estimation itself may become unreliable.

To address these limitations, the second work of this study proposes a tightly coupled framework that integrates classical VIO optimization factors with cross-modal line feature constraints for unified pose estimation. Furthermore, we conduct a comprehensive observability analysis of the proposed system to elucidate the intrinsic relationships between state variables and measurement constraints. To our knowledge, this represents the first rigorous observability study of prior map-aided VIO systems [37].

1.3 Safety-Guaranteed Pose Estimation

Returning to our original objective of developing high-precision, reliable, and safety-critical robotic pose estimation systems, the aforementioned two works still exhibit room for improvement. A key limitation stems from our adoption of weighted least-squares-based factor graph optimization in both fusion frameworks. This local optimization framework fundamentally struggles with pose estimation's non-convex nature, converging to suboptimal local minima when handling noisy observations [38]. Moreover, pose estimation applications hold substantial real-world relevance, underscoring the necessity to address their security implications.

Next, we will examine two key perspectives essential for achieving this goal. (C1) Enhancing the accuracy and stability of pose estimation, which is vital for making informed decisions in dynamic environments. A robust pose estimation process minimizes the risk of errors that lead to unsafe operational conditions. (C2) Forecasting the discrepancy between the estimated pose and the ground truth (GT). Accurate pose error prediction enables systems to implement timely corrective actions, thereby bolstering confidence in the reliability of transformation estimations. Concerning (C1), several elements impact the performance of pose estimation. Observations fed into the estimator often contain faulty measurements (i.e. outliers). These observations typically include measurement noise, which is generally modeled as a Gaussian distribution with zero mean [39]. Meanwhile, the inherent non-linearity and non-convexity of the pose estimation problem mean that a local optimizer cannot guarantee convergence to the global optimal solution [38]. For (C2), it is recognized that pose estimation is subject to error. Our objective is to predict this error by providing an *error bound* for each frame of poses, thereby quantifying the uncertainty (i.e. error) involved [40]. This thought allows the algorithm to deliver predicted pose errors, which serve as safety warnings for the system, even in cases where pose estimation accuracy is compromised. In summary, the insensitivity of pose estimation performance to outliers and model uncertainties is termed robustness, while the system's ability to prevent hazardous situations by self-evaluating uncertainty (localization error) through the Protection Level is defined as safety.

Regarding the global optimal solution, although researchers have reformulated the original state estimation problem as a quadratically constrained quadratic program (QCQP) and employed optimization techniques such as semidefinite programming (SDP) and sum-of-squares (SOS) convex relaxations to achieve certifiably optimal solutions [41, 42, 43, 44], these globally optimal methods remain sensitive to measurement noise and outliers and generally require the construction of specific models tailored to the specific problem. Consequently, their practical applicability in real-world scenarios is limited [45]. To achieve globally optimal estimation in the presence of outliers, we introduce a robust loss function into the original least-squares formulation and leverage graduated non-convexity (GNC) strategy, thereby enabling outlier-resistant global pose estimation [38].

Current research on localization estimation rarely addresses the critical issue of error prediction. However, error prediction is critical, as relying solely on optimization results is insufficient for assessing their reliability, especially in safety-critical applications. Therefore, further exploration is required not only to achieve accurate pose estimation but also to predict and quantify associated errors. This would enable the system to provide early warnings about potential uncertainties, thereby enhancing operational safety and robustness.

When the localization errors can be estimated and monitored simultaneously, it becomes possible to evaluate whether a navigation system remains within safe operational bounds [46]. To address this, we introduce the concept of **safety quantification**, a mechanism that can predict and monitor localization errors within a navigation system. By quantifying these errors, it becomes possible to assess the system's safety in real-time. This approach enables a systematic evaluation of whether the navigation system operates within safe parameters, ensuring both reliability and accuracy. Safety quantification provides a framework for understanding and interpreting the prediction of localization errors, thereby enhancing the overall safety of navigation systems. Reliably quantifying the safety of the derived localization solution is significant for safety-critical autonomous systems before their massive production and deployment. For autonomous robotic and driving vehicles, large localization error can affect the subsequent planning and decision-making steps, which may lead to serious safety accidents. Consequently, localization error prediction is important and necessary to reduce the probability of safety accidents and guarantee the correctness of decision-making. Unfortunately, the safety quantification of the localization solution has not been effectively studied, which still has a big gap in the front.

As mentioned above, the localization accuracy of a navigation system is inevitably affected by observation noise. In addition, the outliers in the observations as erroneous constraints are also an important factor influencing the accuracy. Therefore, the positioning accuracy of a navigation system is compromised by two key error sources. *(1) The measurements noise:* typically modeled as zero-mean Gaussian-distributed random noise, it introduces inherent uncertainty into observations. *(2) Faulty measurements (outliers):* These erroneous constraints deviate significantly from true values and can severely degrade accuracy. Quantifying the error of a localization solution thus hinges on reliably characterizing the errors contributed by the both two components. In robotic localization, several outlier rejection algorithms have existed. The popular methods are random sample consensus (RANSAC) and its variants [47, 48]. However, even after the outlier exclusion operation, system cannot 100% guarantee that there are no outliers in the existing observations. So, outliers remaining in the observations still have a negative impact on localization accuracy, which is commonly ignored.

In civil aviation, Receiver Autonomous Integrity Monitoring (RAIM) is a widely recognized method for assessing the performance of GNSS by predicting positioning errors [8]. RAIM theory posits that positioning errors stem not only from measurement noise but also from the quality of received satellite signals. Crucially, even after fault detection and exclusion (FDE), residual outliers may persist in the satellite data [49]. To predict positioning error, RAIM computes a statistical metric called the protection level (PL), which estimates the maximum potential positioning error through the principle of error propagation in pose estimation. Specifically, PL bounds the error contributions from both measurement noise and undetected outliers, providing real-time alerts when GNSS reliability falls below navigation safety thresholds [50]. Extensively validated in theory and practice, RAIM has proven effective in aviation for error prediction [10]. Inspired by this approach, we propose adapting RAIM principles to the autonomous robotics domain, leveraging its robustness to quantify pose estimation uncertainties.

According to the two conditions presented above for building safety-critical robotic pose estimation systems, the third work in this study proposes a comprehensive and safety-guaranteed state estimation method that can efficiently reject outliers, estimate poses, and predict estimation errors.

1.4 Research Objectives

The primary objective of this study is to construct an accurate, safe, and reliable universal robotic pose estimation system. In terms of improving accuracy, this research first integrates prior point cloud maps with classical VIO to formulate a new least-squares problem, which is then solved through FGO. Furthermore, considering the impact of outliers and non-convexity in pose estimation, we employ a robust estimator with GNC strategy, which is capable of simultaneously performing outlier rejection and global optimization, thereby improving both the localization accuracy and stability of the robotic system. Regarding safety assurance, this work draws inspiration from the RAIM mechanism in the field of GNSS and rigorously derives the localization errors under the aforementioned two pose estimation frameworks (FGO and GNC) based on the error propagation theory, respectively. The specific research objectives are as follows.

Objective 1: The first objective of this work is to mitigate the cumulative localization error inherent in classical VIO. To address this, prior 3D point cloud map is integrated as an additional data source into the VIO framework to enhance localization performance. Specifically, for the cross-modality matching between prior maps and real-time images, the geometric line structure coexisting in the 3D map and 2D image is utilized to build the line feature-based correspondences. With the initial guess of pose provided by VIO, the matched cross-model correspondences are used as new constraints in a loosely coupled optimization model to refine pose estimation.

Furthermore, inspired by the RAIM mechanism in GNSS, this study introduces the concept of safety quantification into robotic localization systems. By analyzing error sources and error propagation during pose estimation, the system predicts the final localization error and defines this specific error bound as safety quantification to ensure operational reliability.

Objective 2: Due to the inherent limitations of the loosely coupled integration between VIO and prior maps in improving localization accuracy, the second objective focuses on constructing a tightly coupled framework. This framework jointly optimizes IMU pre-integration factors, visual point feature factors, and cross-modal line feature constraints for robust pose estimation. And cross-modal line correspondences are established through line matching and a novel line tracking strategy designed to reject outliers.

Meanwhile, an observability analysis of the proposed system is conducted to elucidate the intrinsic relationships between state variables and system measurements, thereby enhancing the theoretical understanding of the framework's performance characteristics.

Objective 3: Because of the non-convex nature of the pose estimation problem, local nonlinear optimization methods are prone to converge to local minima, particularly in the presence of outlier measurements. This issue not only degrades robotic localization accuracy but also compromises the ability of the safety quantification mechanism to reliably predict localization errors. To address these challenges, the third objective proposes a globally optimal and safety-guaranteed pose estimation framework capable of simultaneously rejecting outliers, estimating system states, and predicting localization errors.

Specifically, a robust truncated least squares (TLS) cost function is adopted to handle measurement outliers, while a GNC strategy progressively convexifies the original non-convex optimization problem. Then, an error propagation analysis based on Gaussian noise assumptions and this new optimization model is derived to predict a state error bound for uncertainty quantification. This proposed system achieves both globally optimal state estimates and theoretically grounded safety guarantees while maintaining real-time performance.

1.5 Research Contributions

The contributions of this research and the academic articles are summarized as follows.

Chapter 3 proposes a safety-quantifiable line feature-based visual localization method, which fuses the classical VIO with prior point cloud maps in a loosely coupled form. The detailed contributions of this work are listed as:

1. This study proposes a new line feature-based optimization model, which re-

fines the conventional 2D-3D line constraints from minimizing the line-to-line distance into a point-to-point reprojection model. This new constraint improves the stability of line features in state estimation.

2. A RAIM-inspired safety quantification pipeline is proposed in this study, which predicts the localization errors for the positioning and orientation domains.

3. This is the first proposed integral framework including visual localization and safety quantification for autonomous systems. The effectiveness of the proposed pipeline is verified with challenging datasets collected by both UAV and UGV systems. Moreover, a detailed analysis of the localization safety towards the line features degeneracy is presented.

This work is published in:

- **X. Zheng**, W. Wen, and L. -T. Hsu, ”Safety-quantifiable Line Feature-based Monocular Visual Localization with 3D Prior Map,” in *IEEE Transactions on Intelligent Transportation Systems*. IEEE, 2025. Accept. [3] Project

Chapter 4 develops a tightly coupled framework that associates the classic VIO optimization factors with the proposed cross-model line feature factor for pose estimation. The contributions of this work are listed as follows,

1. An optical flow-related fast line feature tracking method is used to remove correspondence outliers and improve the reliability of cross-modality line feature matching. The IMU pre-integration factors, visual point feature factor, and line feature factor are constructed into a tightly coupled optimization structure to improve the accuracy of state estimation.

2. This work analyzes for the first time the observability of the prior map-aided visual localization framework. The proposed system is constructed into a nonlinear affine model and the observability properties are conducted by finite Lie derivatives through basis element factorization. The results show that the yaw angle around gravity is the only unobservable direction of the proposed system.

3. The proposed framework is evaluated in both simulation and real-world experiments. The results demonstrate the effectiveness of the proposed method.

This study is published in:

- **X. Zheng**, W. Wen, and L. -T. Hsu, "Tightly-coupled Visual/Inertial/Map Integration with Observability Analysis for Reliable Localization of Intelligent Vehicles," in *IEEE Transactions on Intelligent Vehicles*. IEEE, 2024. [37] Project

- **X. Zheng**, W. Wen, and L. -T. Hsu, "Tightly-coupled Line Feature-aided Visual Inertial Localization within Lightweight 3D Prior Map for Intelligent Vehicles," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023. [51]

Chapter 5 proposes a global and safety-guaranteed state estimation method, which can efficiently realize outlier rejection, pose estimation, and estimation error prediction. The contributions of this study are summarized as follows,

1. A universal pose estimation framework that integrates TLS-GNC for outlier rejection and global optimization, while also incorporating an error bound prediction method. This dual mechanism offers enhanced reliability for spatial transformation estimation.

2. For outlier rejection, a threshold setting method based on the assumption of Gaussian noise distribution is introduced, which could dynamically adjust the threshold according to the characteristics of the observations.

3. We comprehensively consider both system measurement errors and outliers remaining in the measurement set, deriving the error propagation within the proposed optimization framework to predict the state estimation error.

4. We validate the proposed framework through both simulations and real-world datasets, showing superior performance in outlier rejection, estimation accuracy, and error bound prediction compared to existing methods. The code and datasets are publicly available, contributing to transparency and reproducibility in the field.

The work is submitted in:

- **X. Zheng**, X. Y. Wen, W. Wen, F. Gao, and L. -T. Hsu, "A Global and Safety-Guaranteed State Estimation Method for Spatial Transformation," *IEEE Transactions on Robotics*.

1.6 Thesis Outline

The outline of this thesis is as follows.

Chapter 2 systematically reviews autonomous robot localization through five critical perspectives: (1) Prior Map-aided Visual Localization analyzes cross-modality matching techniques, exposing underutilization of line features in existing geometric approaches. (2) Pose Estimation Solvers contrasts local and global optimization methods, revealing their accuracy-efficiency tradeoffs. (3) Outlier Rejection evaluates robustness strategies from classical RANSAC to robust loss function. (4)

Observability Analysis dissects unobservable directions in VIO systems, emphasizing the need for rigorous analysis in map-aided cases. (5) State Error Prediction demonstrates critical gaps in error bound prediction for pose estimation compared to GNSS RAIM standards. These literature reviews collectively motivate the methodological innovations in subsequent chapters.

Chapter 3 develops a loosely-coupled architecture for monocular visual localization that bridges 3D prior maps and VIO through line feature constraints while introducing the first safety quantification framework for this new system. The system comprises: (1) a visual localization module establishing geometric constraints between 2D image lines and 3D prior map lines via a foot-point-based optimization model to eliminate VIO drift. (2) a safety quantification module that combines Chi-squared test-based outlier rejection with GNSS-inspired protection level (PL) computation—extended for the first time to six-DoF error prediction (position and orientation) under multiple-fault assumptions. The effectiveness of the proposed safety-quantifiable localization system is verified using the datasets collected by UAV and UGV in indoor and outdoor environments, respectively.

Chapter 4 presents a novel tightly-coupled optimization framework that integrates IMU pre-integration, visual point features, and cross-modal line constraints from lightweight prior maps into a unified state estimator. The system introduces three key innovations: (i) a computationally efficient line representation using endpoints with a fast line tracking strategy for outlier rejection, (ii) a theoretically-grounded line feature cost model validated through rigorous proof, and (iii) the first complete observability analysis for map-aided VIO systems, revealing full global translatability (x,y,z) with only yaw remaining unobservable. The proposed system is evaluated on both simulated outdoor and real-world indoor environments, and the results demonstrate the effectiveness of the proposed methods.

Chapter 5 proposes a comprehensive framework for robust state estimation that addresses two fundamental challenges in safety-critical robotic systems: (1) achieving globally optimal solutions in the presence of significant measurement outliers, (2) providing rigorous uncertainty quantification (error bound) for the estimated states. The proposed methodology combines a graduated non-convexity (GNC) approach with truncated least squares (TLS) to form a robust optimization framework (GNC-TLS) that guarantees convergence to globally optimal solutions while maintaining computational efficiency for real-time implementation. The framework also introduces an error propagation theory that systematically accounts for measurement noise covariance, undetected outliers under the adopted GNC-TLS optimizer to predict state error bounds. The feasibility and practicality of the proposed framework have been validated through simulations and real-world datasets

Chapter 6 concludes this thesis and provides directions for future research.

Chapter 2

Literature Review

This chapter first reviews relevant research on prior map-aided visual localization, focusing on heterogeneous matching schemes between 2D images and 3D point cloud maps. Subsequently, we discuss various solvers—ranging from local to global—for addressing the pose estimation problem, along with outlier rejection strategies in measurements. Additionally, this chapter introduces the research status of observability analysis in visual SLAM and error prediction in state estimation.

2.1 Prior Map-aided Visual Localization

The fundamental principle of cross-modality registration lies in employing different techniques to extract and integrate features from heterogeneous data, which are used to construct constraints for state estimation. Photometric-based methods typically project 3D points into synthetic images and align them with intensity images by maximizing normalized mutual information [52] or minimizing the normalized information distance [53]. However, the nonlinear projection from sparse point clouds to images is susceptible to discretization errors, which can degrade localization accuracy. In contrast, geometric-based methods usually employ bundle adjustment (BA) [54] to reconstruct a 3D point cloud and then align it with a reference point cloud using registration techniques, such as iterative closest point

(ICP) [55], and normal distribution transform (NDT) [56, 57]. Hybrid photometric-geometric methods compute depth images from both point clouds and stereo cameras, then align them for localization [25, 58]. However, large-scale BA operation is time-consuming, and the reconstructed point clouds, derived from image features, are often too sparse for robust alignment. Meanwhile, the range of the depth recovery from the stereo camera is limited because of the requirement for dense texture information.

The other research stream proposed to match the 2D representative features with the 3D point cloud by estimating the scale and the pose of the system simultaneously. For example, Caselitz *et al.* [59] propose a 7-DoF ICP scheme to estimate the scale and pose between 3D LiDAR map and sparse points calculated from the BA method. Liu *et al.* [24] exploit global contextual information between 2D images and 3D point maps to handle the matching ambiguity and solve it on a Markov graph structure by a random walk algorithm. These methods only exploit the potential of corner features but do not fully explore the geometry information (such as lines) from environments. It cannot be ignored that HD map-based localization is favored by academia and industry in autonomous driving [60]. Specifically, Schreiber *et al.* [61] match road lane markings and curbs detected on stereo camera with an HD-map for localization by Kalman Filter, where the maps are created by a sensor setup containing a high precision GNSS unit, Velodyne laser scanner, and cameras. Ye *et al.* [62] render vertex and normal maps from 3D surfel maps via principal component analysis, projecting them into depth maps to provide depth priors for tracked direct point features. Pose estimation is then performed by optimizing direct photometric error in a 6-DoF framework. Huang *et al.* [63] represent dense 3D prior maps as Gaussian mixture models (GMMs), enabling association with online visual measurements to construct hybrid structural factors for joint pose optimization.

In addition, some researchers apply semantic information to get cross-modal correspondences between 2D image matching with 3D prior maps to solve localization problems. Specifically, Liang *et al.* construct a hybrid constraint with Dirichlet distribution between semantic and structure information from prior maps, and then apply the Expectation-Maximization algorithm to jointly optimize data associations and poses [64]. Zhang *et al.* utilize semantic information to build a point-to-plane ICP algorithm for cross-model registration and execute the localization task by a decoupling optimization strategy [65]. Qin *et al.* [66] propose a lightweight HD-map composed of semantic elements such as lane lines, crosswalks, and ground signs on the road. Vehicles are localized in the semantic map by online image segmentation and the ICP method. Zhou *et al.* [67] develop a unified pose graph-based estimator including crowd-sourced mapping and a low-cost map-aided localization system based on the semantic road marking.

2.2 Pose Estimation Solvers

Visual localization, point cloud registration, and data fusion in robotics and autonomous driving systems can be formulated as pose estimation problems. The state variables contained in pose estimation problem are scaling, rotation, and translation [68], hence this problem is also called as **spatial transformation** issue.

The most straightforward method for spatial transformation involves randomly selecting a finite number of correspondences to compute the analytical solution of state variables. This approach is known as *minimal solvers*, where only the minimum number of observations is used to solve states [69]. A notable example is the computation of the fundamental matrix $\boldsymbol{F}$ in computer vision, which uses the 8-point algorithm to estimate the camera pose transformation between two frames [70]. However, due to the lack of redundancy in measurements, minimal solvers exhibit poor noise immunity and are typically employed to provide an initial guess for subsequent state optimization [6]. To cope with inevitable observation noises, it is usually assumed to follow a Gaussian distribution. In this context, spatial transformation becomes a state estimation problem, aiming to find the optimal states from noisy data. These solutions are called *non-minimal solvers*. Beyond the extended Kalman filter (EKF) and its variants, such as OpenVINS (MSCKF) [22], observability-based consistent EKF (OC-EKF) [71], and invariant EKF (InEKF) [72], nonlinear optimization based solutions have gained broader attention in recent studies.

From the optimization perspective, state estimation is usually formulated as a least squares problem,

$$\min_{\boldsymbol{x} \in \mathcal{X}} \sum_{i \in \mathcal{M}} \left\| \boldsymbol{r}(\boldsymbol{z}_i, \boldsymbol{x}) \right\|^2, \tag{2.1}$$

where $\boldsymbol{x}$ is the state variables, $\mathcal{X}$ is the state domain, $\mathcal{M}$ is the measurements set, $\boldsymbol{z}_i$ is the i-th measurement, $\boldsymbol{r}$ means the residual error (e.g., in computer vision, residual refers to the pixel reprojection error. For point cloud registration, residual means the distance between correspondences). Meanwhile, from the probabilistic view, the optimal solution of the weighted least squares problem is equivalent to the maximum likelihood estimation of the state on the Gaussian noise assumption [73]. The solution of this least squares problem can be categorized into *local* and *global* optimization based on the requirement of initial guesses or not.

Local optimization can be classified into two categories based on their search modes: line search and trust region approaches. In line search, the search direction is first fixed, and then seeks the incremental step is determined along this direction. Representing methods are the steepest descent method and Gauss-Newton method [74]. In contrast, trust region methods [75] define a region for searching, and then step directions is chosen to find the optimal within this region. Examples include the

Dog-leg and Levenberg–Marquardt (LM) algorithm (damped least-squares method) [76]. In practice, widely used open-source C++ optimization libraries such as g2o (general graphic optimization) [77], GTSAM (factor graph optimization) [78], and Ceres [79] are commonly applied to solve spatial transformation problems. These libraries implement various local optimization algorithms, with the distinction that they adopt different approaches to model optimization problems and handle large-scale matrix computations. For instance, g2o represents the least squares problem as a graph, where state variables and constraints are denoted as vertices and edges [80]. Thanks to the sparse connectivity of the graph, g2o efficiently handles large matrix marginalization and elimination calculations. GTSAM introduces a novel data structure called *Bayes tree*, which supports incremental reordering and dynamic relinearization, enhancing the efficiency of nonlinear optimization [81, 82]. On the other hand, Ceres is also built on a graph-based architecture and is highly regarded for its accessibility and flexibility. It has been widely adopted in classical SLAM frameworks such as VINS-Mono [1] and OKVIS [83].

However, local optimization methods require initial guesses of states and are prone to getting trapped in local minima, which leads to erroneous results under noise disturbances. This is especially problematic in safety-critical applications [84]. To address these limitations, researchers have increasingly explored global optimization methods for solving transformation estimation problems, aiming for algorithms that can converge to a globally optimal without requiring initial guesses. For example, Yang *et al.* [85] use the Branch-and-bound (BnB) framework to solve point cloud registration. Parra *et al.* [86] improve the bound function of BnB to accelerate the rotation search issue in 3D alignment. However, the time consumption of BnB increases exponentially with the size of the observation set [87]. Besides, the exploration of global optimization gradually shift to the convex relaxation of specific transformation problems, alongside discussions around the certification of optimality. Carlone *et al.* reformulate the pose graph optimization (PGO) problem and achieve reliable, global solutions by Lagrangian duality [88, 89]. Rosen *et al.* [43] propose SE-Sync, a Riemannian truncated-Newton trust-region method designed to solve pose graph optimization globally and certifiably. Briales *et al.* [42] formulate the relative pose problem as a quadratically constrained quadratic program (QCQP) and then convert it into a semidefinite program (SDP) by Shor's relaxation to obtain the global optimum. Wu *et al.* [90] formulate the distance-based relative state estimation as an edge-based SDP problem and use distributed optimization methods to globally solve it in a scalable way.

However, global optimization algorithms based on convex relaxation suffer from high computational costs, making it difficult to meet real-time requirements when handling large-scale operations with numerous observations. To address this, several compromise frameworks with higher practicality have been proposed. Wu *et al.* [91] propose a specific formulation for planar PGO problem with efficient implementation of GNC. Zhou *et al.* [92] and Yang *et al.* [38] employ GNC to efficiently solve point cloud registration and spatial perception questions in a globally optimal manner, where the loss function used in the former is Geman-McClure, and in the latter is truncated least squares cost. TEASER++ proposed by Yang *et al.* also utilize the GNC mechanism to replace the SDP relaxation to accelerate calculations [44]. Based on this, we argue that GNC can achieve global convergence while maintaining real-time performance in practical applications. Therefore, the study in Chapter 5 adopts the GNC framework to achieve global convergence in nonlinear optimization tasks.

2.3 Outlier Rejection

Due to noise interference, false detection, and incorrect data association, outliers are inevitable in practical applications, which can seriously degrade the accuracy of state estimation. Outlier rejection can be regarded as a maximum consensus problem (MC), its mathematical model is,

$$\max_{\boldsymbol{x}\in\mathcal{X},\mathcal{I}\subseteq\mathcal{M}} |\mathcal{I}| \quad s.t. \quad |\boldsymbol{r}(\boldsymbol{z}_i,\boldsymbol{x})| < \Gamma, \quad \forall i \in \mathcal{I} \tag{2.2}$$

where $\mathcal{I}$ is the inlier measurement set, Γ is the inlier threshold. (2.2) expects to find the maximum number of inliers from the observation set, thereby conforming the consensus that residuals fall below a defined threshold [93]. RANSAC [48] and BnB [94, 95] methods can be used to solve this problem (2.2). However, RANSAC is ineffective with high outlier rates. BnB suffers from exponential growth in computational complexity as the dataset size increases. Besides, some researchers use chi-square tests and the greedy algorithm to iteratively remove outliers under the assumption of Gaussian noise [96]. Nevertheless, these approaches often prove inefficient due to their algorithmic limitations.

Another school of solution is M-estimation [97], which appends a robust loss function to the original optimization cost in (2.1), shown as,

$$\min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i\in\mathcal{M}} \rho(\|\boldsymbol{r}(\boldsymbol{z}_i,\boldsymbol{x})\|^2), \tag{2.3}$$

Here $\rho(\cdot)$ can be Huber [98], Geman-McClure, and truncated least squares (TLS) cost [99]. For instance, Qin *et al.* [1] utilize Huber loss function to improve the robustness of visual-inertial SLAM system. The Geman-McClure penalty is used in [92] to deal with outliers in point cloud registration. Yang *et al.* [100] employ TLS cost for robust registration under high outlier rates. Antonante *et al.* [101] compare the performance of generalized MC and TLS in outlier scenarios, demonstrating that TLS more efficiently rejects outliers and produces more accurate estimates [102].

Building on this, the present work in Chapter 5 adopts TLS as the strategy for outlier rejection. It is important to note that the optimization problem in (2.3) remains a maximum likelihood estimation question, with the goal of finding the optimal least squares solution [101]. In essence, loss function in (2.3) restructures the model in (2.1) to incorporate outlier rejection, enabling simultaneous optimization of both outlier rejection and state estimation. However, the introduction of $\rho(\cdot)$ exacerbates the nonconvexity of the optimization problem, making the limitations of local solvers more prominent. As a result, the study in Chapter 5 leverages a global solver framework to jointly optimize state estimation and outlier rejection.

2.4 Observability Analysis of Visual Localization

Observability plays a crucial role in state estimation, determining the ability to estimate the states of a dynamic system based on its measurements, ensuring that the state of the system can be effectively understood [103, 104]. In other words, An unobservable state variable produces indistinguishable trajectories under the system's measurement constraints [105]. Observability analysis not only evaluates estimation performance but also identifies degenerate motions that introduce additional unobservable directions, critically affecting estimator accuracy [106, 107]. Furthermore, it enables online self-calibration of camera-IMU systems by ensuring the observability of calibration parameters [108, 109, 110].

Specifically, Martinelli *et al.* [111] derive a closed-form VINS solution that analytically expresses IMU biases, 3D velocity, and absolute roll and pitch angles are observable states. Hesch *et al.* [112] introduce basis elements in observability analysis to factorize the observability matrix. The factorized matrix is used to determine the unobservable mode of nonlinear VIO systems by computing a finite number of Lie derivatives, which shows that the monocular VINS has four unobservable DoFs, corresponding to the three global translations and global yaw angle around the gravity vector. Guo *et al.* [113] extend this approach to IMU-RGBD systems and study the observability properties based on Hesch's method, incorporating both point and plane features. Further advancing the analysis, Yang *et al.* [107] comprehensively examine the observability of VINS with various geometric features, including points, lines, planes, and their combinations in linearized systems.

Observability analysis provides critical insights into a system's limitations by revealing its unobservable directions, which directly constrain state estimation performance. A rigorous formulation of map-aided VIO is essential for ensuring reliable localization in safety-critical robotic applications.

2.5 State Error Prediction

Error analysis is a well-established discipline in mathematics, focusing on the types and quantities of errors (uncertainties) [114]. However, in safety-critical applications such as robotics and autonomous driving, the emphasis has been on enhancing the accuracy of state estimation, with limited attention given to providing performance guarantees or conducting detailed error analyses. In computer vision, Szeliski *et al.* use the classical Cramér-Rao lower bound to characterize estimator uncertainty, which can be calculated via the inverse of Hessian matrix [115]. The work in [116] adopts the optimum of sparse semidefinite relaxation as the lower bound to certify global optimality. However, these methods fail to offer a referable upper bound for the estimation errors. Yang *et al.*[44] use the triangle inequality to roughly compute an error bound for the worst-case state estimation, which often leads to a bound that is overly conservative and far from the true error, thus reducing its practical value. In his recent work, the error bounds of the object pose estimation are taken as optimization variables, followed by a convex relaxation to solve the probabilistic worst-bound [40]. Yet, error bounds obtained from his method are still much larger than true errors.

In GNSS, researchers have applied error analysis to global positioning system and developed the RAIM theory for positioning safety [117]. A key metric in RAIM, known as protection level (PL), models the error propagation within weighted least squares (WLS) under Gaussian noise assumption, providing a way to compute error bounds for state estimation [10]. Recently, this theory has found increasing application in robotics. For example, Li *et al.* use PL to predict the visual localization errors of UAVs based on ORB-SLAM2 [118]. Joerger *et al.* leverage RAIM theory to monitor the safety of robots and autonomous vehicles using EKF and particle filter with LiDAR/IMU platform [119, 120, 121]. The widespread application of PL in GNSS and its successful application in robotics demonstrate its versatility across different domains. In this study, we integrate the PL concept from GNSS into the proposed state optimization framework, proposing a theoretical derivation for predicting error bounds to quantify uncertainty and ensure the safety of state estimation.

Chapter 3

Safety-quantifiable Line Feature-based Monocular Visual Localization with 3D Prior Map

3.1 System Overview

The proposed system pipeline is shown in Fig. 3.1. The system mainly includes two parts: the visual localization module and the safety quantification module. Firstly, a high-precision 3D point cloud as a prior map is generated in advance, in which all 3D lines are extracted and stored offline [122]. Then the 3D lines are projected into image frames based on the initial field of view (FoV) of the cam-

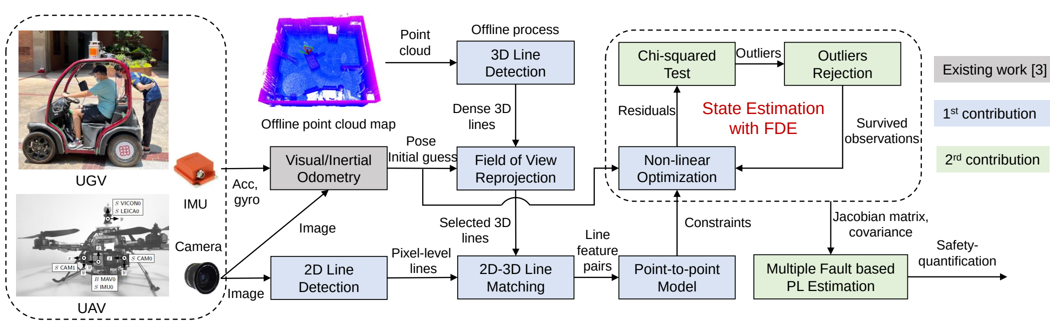


Figure 3.1: Framework of the proposed system: 3D lines are pre-extracted from the prior map, and 2D lines are detected online in the camera frame. With the initial state from VINS [1], 2D-3D line pairs are optimized. Outliers are rejected via a Chi-squared test (FDE), and passing cases compute a protection level (PL) to quantify state estimation safety.

era that is provided by the pose estimation from the VINS [1]. On the other hand, the images captured by an onboard camera are used to detect 2D lines in real-time (Chapter 3.2.2). The two types of lines are matched into line feature correspondences, which serve as the predictions and measurements of the state optimization model for absolute visual localization (Chapter 3.2.3) in this prior map. Secondly, the safety quantification is implemented based on a reliable FDE and protection level calculation. Specifically, a Chi-squared test is applied to detect outliers based on the weighted least-squares model iteratively (Chapter 3.3.2). After removing the detected outliers, the surviving observations are re-input into the optimization model for state estimation until the Chi-squared test passes. Then the protection level is estimated based on the multiple fault assumption by quantitatively evaluating the safety of the visual localization solution (Chapter 3.3.4).

The notations are defined as follows. $(\cdot)^w$ denotes the world frame, and the origin coincides with the prior map starting point. $(\cdot)^b$ represents the body frame, and the origin is defined on the inertial measurement unit (IMU). $(\cdot)^c$ is the camera frame. The translation vector and rotation matrix are represented by $\boldsymbol{R}$ and $\boldsymbol{t}$, respectively, and also by their corresponding Lie algebra $\boldsymbol{\xi}$ [73]. $\boldsymbol{R}_w^b, \boldsymbol{t}_w^b$ means the transformation from the world frame to the body frame, $\boldsymbol{R}_b^c, \boldsymbol{t}_b^c$ is from the body frame to the camera frame. b_k is the subscript of the body frame in k^{th} image. The state variables in this work are the pose of UGV or UAV at the world frame while taking the k^{th} image, expressed by $\boldsymbol{R}_{b_k}^w$ and $\boldsymbol{t}_{b_k}^w$. $\boldsymbol{P}(x,y,z)$ is a 3D point and $\boldsymbol{p}(\mu,\nu)$ means a 2D image pixel point. The notation of Chi-Squared distribution is χ^2.

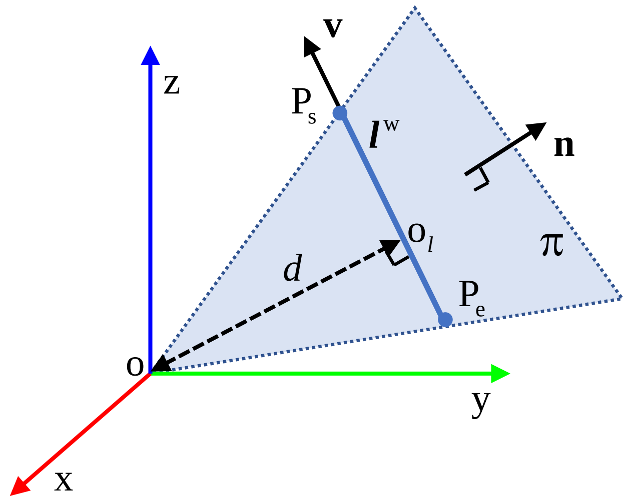


Figure 3.2: The line representation diagram.

3.2 State Estimation: Visual Localization with Prior Map

3.2.1 Line Feature Representation

In visual-based state estimation, a line in 3D space is typically represented using *Plücker* coordinates and orthonormal representations [107, 123, 124]. The former is employed for line initialization and projection, the latter is applied for optimization in state estimation. As illustrated in Fig. 3.2, the *Plücker* coordinate of a 3D line $\boldsymbol{l}^w$ can be expressed as a 6-vector $\mathcal{L} = (\mathbf{n}^\top, \mathbf{v}^\top)^\top$. The vector $\mathbf{v}$ denotes the direction of the line, and $\mathbf{n}$ is the normal vector of the plane π that includes both the line $\boldsymbol{l}^w$ and the origin o. If there are two known points $\boldsymbol{P}_s$, $\boldsymbol{P}_e$ on $\boldsymbol{l}^w$, the *Plücker* coordinate of the line can be expressed as

$$\begin{bmatrix} \mathbf{n} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}_s \times \boldsymbol{P}_e \\ \boldsymbol{P}_e - \boldsymbol{P}_s \end{bmatrix}. \tag{3.1}$$

However, infinite lines in 3D space possess four DoFs, necessitating an additional constraint on the Plücker coordinate:

$$\mathbf{n}^\top \mathbf{v} = \mathbf{0}. \tag{3.2}$$

This additional constraint complicates optimization calculations. Therefore, the orthonormal representation $(\mathbf{U}, \mathbf{W})$ is introduced to facilitate the computation of partial derivatives and nonlinear optimization. This representation can be obtained through the orthonormal decomposition of the *Plücker* coordinate [123],

$$[\mathbf{n} \mid \mathbf{v}] = \mathbf{U} \begin{bmatrix} \omega_1 & 0 \\ 0 & \omega_2 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \omega_1 & -\omega_2 \\ \omega_2 & \omega_1 \end{bmatrix}. \tag{3.3}$$

where ω_1, ω_2 are scalar parameters and $\mathbf{U} \in \mathbb{R}^{3\times 2}$ is a 3D orthonormal basis for the line, satisfying $\mathbf{U}^\top \mathbf{U} = \mathbf{I}_2$. The columns of $\mathbf{U}$ span the plane defined by the line's Plücker coordinates and automatically enforce the orthogonality constraint. Accordingly, the *Plücker* coordinate can be transformed by orthonormal representation,

$$\mathcal{L} = (\omega_1 \mathbf{u}_1^\top, \omega_2 \mathbf{u}_2^\top)^\top, \tag{3.4}$$

where, $\mathbf{u}_1, \mathbf{u}_2$ are the i-th column of matrix $\mathbf{U}$.

From (3.1) and (3.3), it is evident that both *Plücker* and orthonormal representations can be derived from any two points $\boldsymbol{P}_s$, $\boldsymbol{P}_e$ on the line. Thus, two points uniquely determine a spatial line. The difference between the line segments used in this article and the above is that the lines we used are extracted from prior maps. We can obtain the line segments in 3D space directly without calculating their positions by multi-view geometry principles, which means that the two endpoints of the lines have explicit 3D coordinates rather than being infinite. Therefore, this study uses the two explicit endpoints $\boldsymbol{P}_s$, $\boldsymbol{P}_e$ to represent the spatial line segment and conducts subsequent operations based on these endpoints. Compared to the previously men-

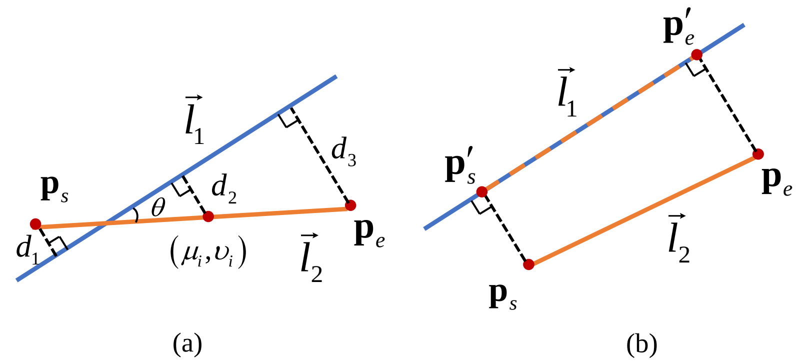


Figure 3.3: An illustration of line similarity criteria. (a) The line distance is the sum lengths of the dashed lines, denoted by $\sum d_i$. The angle of two lines $\theta = \angle(\vec{l}_1, \vec{l}_2)$ (b) the line overlap rate refers to the percentage of the orange dashed line on the blue line.

tioned two methods, this representation is clearer and more direct in terms of both projection and optimization.

3.2.2 2D-3D Cross-Modality Line Matching

In this study, the 2D lines are detected by a fast line detection (FLD) method [125], and are indicated by two endpoints $(\boldsymbol{p}_s, \boldsymbol{p}_e)$ in 2D images. Line features existing in 3D point clouds are extracted by a segment-based 3D line detection method [122]. In the method, the 3D points are segmented by planes and projected into a virtual image, then the 2D lines detected on this image are re-projected into 3D space to get line segments. Once the 3D lines are obtained, the dense point cloud can be replaced by line segments that are represented by two 3D endpoints $(\boldsymbol{P}_s, \boldsymbol{P}_e)$, and only these lines need to be loaded during visual localization.

To get line correspondences, we first project the 3D lines into the camera imaging plane according to the initial pose of the camera from VINS. Then, these lines are matched with the 2D lines captured from the actual world in this image depending on the defined similarity criteria. The criteria in this work are mainly determined by the distance, angle, and overlap between two line segments [30, 36], which are

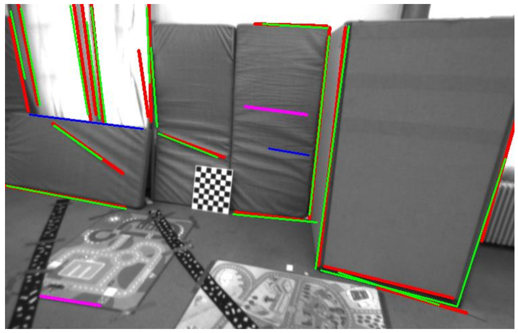


Figure 3.4: The red lines in the image are the detected 2D lines by FLD method. And the red and green line pairs are the valid matching associations. The blue and purple lines are the rejected outliers.

shown in Fig. 3.3. Specifically, the general function of a 2D line l is

$$Ax + By + C = 0, \tag{3.5}$$

the line distance D is defined as the sum of distances from n equally interval sampling points $(\mu_i, \nu_i), i = 1 \cdots n$ (containing endpoints) on one line segment to the other (shown in Fig. 3.3 (a)), the equation is,

$$D = \sum_{i=1}^{n} d_i = \sum_{i=1}^{n} \frac{|A\mu_i + B\nu_i + C|}{\sqrt{A^2 + B^2}}. \tag{3.6}$$

By assigning appropriate thresholds, which are experimentally determined, the algorithm can obtain line correspondences that meet requirements. The line detection and matching results at a single camera frame are shown in Figure 3.4.

3.2.3 Point-to-point line-based Optimization Model

Typically, the cost function for line features is formulated by summing the distances between corresponding lines. In this work, we reformulate the optimization problem by converting line-distance minimization into a point reprojection error model [73].

Assuming there is a line pair l_1, l_2, the general function of l_1 is shown in (3.5), and the distance between them can be calculated by the sum distance of the points on l_2 from l_1, which is consistent with (3.6). Take one point $p(\mu,\nu)$ on l_2 as an example, the distance from $\boldsymbol{p}$ to l_1 is equivalent to the distance from the point $\boldsymbol{p}$ to its vertical foot point $\hat{\boldsymbol{p}}^f(\hat{\mu}^f,\hat{\nu}^f)$ on l_1. This point-to-point distance is essentially the same as the reprojection error in visual pose estimation, both of them are used to represent the difference between the coordinates of two-pixel points.

Based on this, the objective function can be built by minimizing the square sum of the point pair error. This is a classical weighted least squares (WLS) problem and can be formulated as,

$$\min \sum \left\| \hat{\boldsymbol{p}}^f - \boldsymbol{p} \right\|_{\boldsymbol{Q}}^2, \tag{3.7}$$

where $\hat{\boldsymbol{p}}^f$ and $\boldsymbol{p}$ are the measurement and prediction of this function, respectively. The $\boldsymbol{Q}$ is the covariance of observations and is experimentally determined. In particular, $\boldsymbol{Q}$ typically arises from camera calibration errors, as well as feature detection errors caused by camera over-exposure and motion blur. For point features, these errors are commonly in the range of 1–3 pixels, which can be used as a guideline for setting $\boldsymbol{Q}$ [50]. The coordinate of the foot point $\hat{\boldsymbol{p}}^f(\hat{\mu}^f,\hat{\nu}^f)$ can be computed based on (3.5) as below [124]

$$\hat{\mu}^f = \frac{B^2\mu - AB\nu - AC}{A^2+B^2}, \quad \hat{\nu}^f = \frac{A^2\nu - AB\mu - BC}{A^2+B^2}. \tag{3.8}$$

The specific process of constructing an optimization model is illustrated in Fig. 3.5. Suppose that there is a 3D line segment l^w in 3D space at the time while the k^{th} image is taken by the camera, and one of its endpoints is $\boldsymbol{P}(x,y,z)$. The transformation of the camera during this time from the world frame is $\boldsymbol{T}_w^c = [\boldsymbol{R}|\boldsymbol{t}]$ (corresponding to Lie algebra $\boldsymbol{\xi}$ [126]), the 3D line l^w is projected into the image plane of the camera shown as l^c, which is matched with the line segment $\hat{l}^c$ in this image. Based on the pre-calibrated extrinsic matrix $\boldsymbol{T}_w^c$, this projection model can

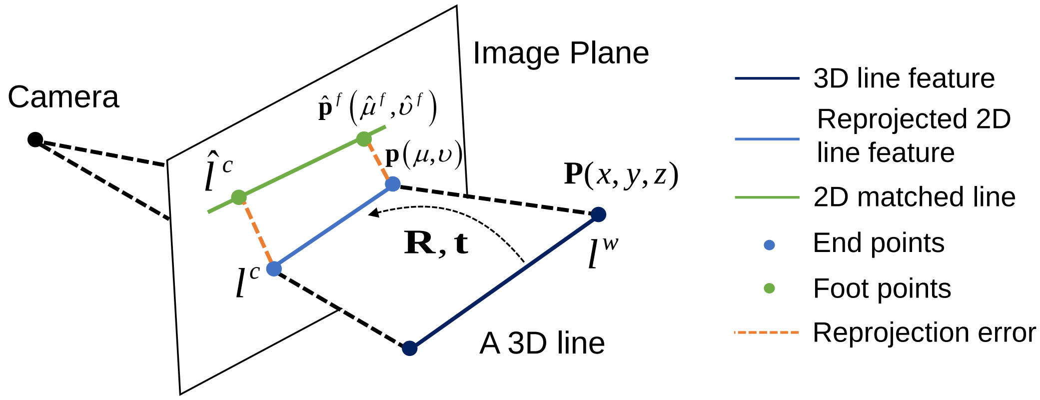


Figure 3.5: An illustration of line reprojection error from the 3D prior map to the 2D image. A 3D point $\boldsymbol{P}(x,y,z)$ on l^w is projected into a camera image with $\boldsymbol{p}(\mu,\nu)$ based on the transformation $\boldsymbol{R},\boldsymbol{t}$. The foot point $\hat{\boldsymbol{p}}^f(\hat{\mu}^f,\hat{\nu}^f)$ of $\boldsymbol{p}(\mu,\nu)$ on l^c is found to build a pixel cost function that is minimized.

be expressed as

$$s\begin{bmatrix}\mu\\ \nu\\ 1\end{bmatrix} = \boldsymbol{K}\boldsymbol{T}\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix} = \boldsymbol{K}\exp\left(\boldsymbol{\xi}^{\wedge}\right)\boldsymbol{P}, \tag{3.9}$$

where $\boldsymbol{p}(\mu,\nu)$ is the pixel location of the endpoint $\boldsymbol{P}(x,y,z)$ in an image plane. s is the scale and $\boldsymbol{K}$ is the intrinsic matrix of the camera [70]. The symbol $\wedge$ is a vector to skew-symmetric conversion. Combined with the state variables we defined in Chapter 3.1, the transformation $\boldsymbol{T}_w^c$ can be decomposed as

$$\begin{aligned}\boldsymbol{R}_w^c &= \boldsymbol{R}_{b_k}^c\boldsymbol{R}_w^{b_k}\\ \boldsymbol{t}_w^c &= \boldsymbol{R}_{b_k}^c\boldsymbol{t}_w^{b_k}+\boldsymbol{t}_{b_k}^c\end{aligned}. \tag{3.10}$$

where $\boldsymbol{R}_{b_k}^c$ and $\boldsymbol{t}_{b_k}^c$ refer to the transformation between the camera and IMU at the k^{th} image, which are invariant parameters because the camera and IMU are fixedly mounted on autonomous robots, so we have $\boldsymbol{T}_{b_k}^c = \boldsymbol{T}_{b_0}^c = \boldsymbol{T}_b^c$, and this transformation can be obtained by the joint camera-IMU calibration [127]. In our system, the initial guess of states $\boldsymbol{T}_w^{b_k}$ is provided by VINS [1].

In the image plane, After obtaining the prediction $\boldsymbol{p}(\mu,\nu)$ based on (3.9), the measurement $\hat{\boldsymbol{p}}^f(\hat{\mu}^f,\hat{\nu}^f)$ on the matched line pair l^c can be computed by (4.16). So the error function depended on the point-to-point reprojection error can be shown as

$$\boldsymbol{r} = \left\| \hat{\boldsymbol{p}}^f - \boldsymbol{p} \right\| = (\hat{\mu}^f - \mu)^2 + (\hat{\nu}^f - \nu)^2. \tag{3.11}$$

Therefore, the cost function (3.7) can be rewritten by

$$\boldsymbol{\xi}^* = \min_{\boldsymbol{\xi}} \sum_{k=1}^{N} \left\| \hat{\boldsymbol{p}}_k^f(\hat{\mu}^f(\boldsymbol{\xi}),\hat{\nu}^f(\boldsymbol{\xi})) - \frac{1}{s_k}(\boldsymbol{K}\exp(\boldsymbol{\xi}^\wedge)\,\boldsymbol{P}_k) \right\|_{\boldsymbol{Q}}^2, \tag{3.12}$$

where N is the number of line pairs on the current image. The proposed method selects two endpoints from each set of pairs as observations for the optimization model. Moreover, the measurements $\hat{\boldsymbol{p}}^f$ are variable related to the state $\boldsymbol{\xi}$.

3.2.4 Optimization Solver

For the non-linear optimization problem in this study, the Gauss-Newton (GN) method is employed and is implemented by the Ceres solver [79] for solving. First, the error function is linearized by Taylor expansion about $\boldsymbol{x}$ shown as

$$\boldsymbol{r} = \boldsymbol{f}(\boldsymbol{x} + \Delta\boldsymbol{x}) \approx \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{J}^\top \Delta\boldsymbol{x}. \tag{3.13}$$

where $\boldsymbol{x}$ is the state variable, and $\boldsymbol{J}$ is the Jacobian matrix. If the Jacobian matrix is known, given the initial guess of states $\boldsymbol{x}$, the optimization increment $\Delta\boldsymbol{x}$ can be obtained by

$$\boldsymbol{H}\Delta\boldsymbol{x} = \boldsymbol{g} \tag{3.14}$$

where $\boldsymbol{H} = \boldsymbol{J}^\top\boldsymbol{J}$, $\boldsymbol{g} = -\boldsymbol{J}^\top\boldsymbol{f}(\boldsymbol{x})$ [128]. Then the states are updated by $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \Delta\boldsymbol{x}$, until the algorithm converges, where i is the iterative index.

In terms of the Jacobian matrix in this work, the proposed method utilizes the perturbation model in Lie algebra to derive it [73]. Based on the chain rule and

(3.11), the derivation of each error term with respect to the state variable can be calculated by

$$\boldsymbol{J} = \frac{\partial \boldsymbol{r}}{\partial \delta \boldsymbol{\xi}} = \lim_{\delta \xi \to 0} \frac{\boldsymbol{r}\left(\delta \boldsymbol{\xi} \oplus \boldsymbol{\xi}\right) - \boldsymbol{r}(\boldsymbol{\xi})}{\delta \boldsymbol{\xi}} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{p}} \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{P}'} \frac{\partial \boldsymbol{P}'}{\partial \delta \boldsymbol{\xi}}. \tag{3.15}$$

Here $\oplus$ refers to the disturbance left multiplication in Lie algebra, $\delta \boldsymbol{\xi}$ is the perturbation value. The $\boldsymbol{P}'$ is a point in the camera frame, which is transformed by the 3D space point $\boldsymbol{P}$ in the world frame, referring to Equation (3.9), we have

$$\boldsymbol{P}' = (\boldsymbol{T}\boldsymbol{P})_{1:3} = \exp\left(\boldsymbol{\xi}^{\wedge}\right) \boldsymbol{P}_{1:3} = [X', Y', Z']^{\top}. \tag{3.16}$$

The specific calculation procedure for the three components of the Jacobian matrix in (3.15) is shown below. Based on the chain rule in (3.15), we need to solve each partial derivative of this equation. The first item based on (4.16) and (3.11) can be computed by

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{p}} = \begin{bmatrix} \frac{\partial \boldsymbol{r}}{\partial \mu} & \frac{\partial \boldsymbol{r}}{\partial \nu} \end{bmatrix} = \frac{2}{A^2 + B^2} \begin{bmatrix} A^2(\mu^f - \mu) + AB(\nu^f - \nu) \\ AB(\mu^f - \mu) + B^2(\nu^f - \nu) \end{bmatrix}^{\top}. \tag{3.17}$$

For the second item, according to (3.9) and (3.16), $\boldsymbol{P}'$ is a projected point in the camera frame and its pixel location $\boldsymbol{p}(\mu, \nu)$ on the image plane is

$$s\boldsymbol{p} = \boldsymbol{K}\boldsymbol{P}', \tag{3.18}$$

$$\mu = f_x \frac{X'}{Z'} + c_x, \quad \nu = f_y \frac{Y'}{Z'} + c_y, \tag{3.19}$$

f_x, f_y, c_x and c_y is the intrinsic parameters of the camera,

$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{P}'} = -\begin{bmatrix} \frac{\partial \mu}{\partial X'} & \frac{\partial \mu}{\partial Y'} & \frac{\partial \mu}{\partial Z'} \\ \frac{\partial \nu}{\partial X'} & \frac{\partial \nu}{\partial Y'} & \frac{\partial \nu}{\partial Z'} \end{bmatrix} = -\begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix}. \tag{3.20}$$

For the third part, $\boldsymbol{P}' = \boldsymbol{T}\boldsymbol{P}$ and give $\boldsymbol{T}$ a left perturbation $\Delta \boldsymbol{T} = \exp\left(\delta \boldsymbol{\xi}^{\wedge}\right)$, whose

Lie algebra is $\delta\boldsymbol{\xi} = [\delta\boldsymbol{\rho}, \delta\boldsymbol{\phi}]^T$ [73], then,

$$
\begin{aligned}
\frac{\partial(\boldsymbol{T}\boldsymbol{P})}{\partial\delta\boldsymbol{\xi}} &= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\exp(\delta\boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge)\boldsymbol{P} - \exp(\boldsymbol{\xi}^\wedge)\boldsymbol{P}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{(\boldsymbol{I}+\delta\boldsymbol{\xi}^\wedge)\exp(\boldsymbol{\xi}^\wedge)\boldsymbol{P} - \exp(\boldsymbol{\xi}^\wedge)\boldsymbol{P}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\delta\boldsymbol{\xi}^\wedge\exp(\boldsymbol{\xi}^\wedge)\boldsymbol{P}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\begin{bmatrix} \delta\boldsymbol{\phi}^\wedge & \delta\boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{R}\boldsymbol{P}+\boldsymbol{t} \\ 1 \end{bmatrix}}{\delta\boldsymbol{\xi}} \\
&= \lim_{\delta\boldsymbol{\xi}\to\mathbf{0}} \frac{\begin{bmatrix} \delta\boldsymbol{\phi}^\wedge(\boldsymbol{R}\boldsymbol{P}+\boldsymbol{t})+\delta\boldsymbol{\rho} \\ \mathbf{0}^T \end{bmatrix}}{[\delta\boldsymbol{\rho}, \delta\boldsymbol{\phi}]^T} \\
&= \begin{bmatrix} \boldsymbol{I} & -(\boldsymbol{R}\boldsymbol{P}+\boldsymbol{t})^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix},
\end{aligned}
\tag{3.21}
$$

taking out the first 3 dimensions, we have

$$
\frac{\partial\boldsymbol{P}'}{\partial\delta\boldsymbol{\xi}} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{P}'^\wedge \end{bmatrix}, \tag{3.22}
$$

where,

$$
-\boldsymbol{P}'^\wedge = \begin{bmatrix} 0 & Z' & -Y' \\ -Z' & 0 & X' \\ Y' & -X' & 0 \end{bmatrix}. \tag{3.23}
$$

Merging formula (3.17) (3.20) and (3.21), the final Jacobian matrix can be solved as

$$
\boldsymbol{J} = \frac{\partial\boldsymbol{r}}{\partial\delta\boldsymbol{\xi}} = -\frac{2}{A^2+B^2}[J_1, J_2, J_3, J_4, J_5, J_6], \tag{3.24}
$$

among that,

$$\begin{aligned}
J_1 &= m\frac{f_x}{Z'}, \\
J_2 &= n\frac{f_y}{Z'}, \\
J_3 &= -\frac{mf_xX' + nf_yY'}{Z'^2} \\
J_4 &= -nf_y - \frac{mf_xX' + nf_yY'}{Z'^2}Y' \\
J_5 &= mf_x + \frac{mf_xX' + nf_yY'}{Z'^2}X' \\
J_6 &= -m\frac{f_xY'}{Z'^2} + n\frac{f_yX'}{Z'^2},
\end{aligned} \tag{3.25}$$

with,

$$\begin{aligned}
m &= A^2(\hat{\mu}^f - \mu) + AB(\hat{\nu}^f - \nu) \\
n &= AB(\hat{\mu}^f - \mu) + B^2(\hat{\nu}^f - \nu)
\end{aligned}. \tag{3.26}$$

Therefore, the localization problem could be solved iteratively based on the devised Jacobian matrices.

3.3 Safety Quantification of Visual Localization with Prior Map

3.3.1 Observation Model

Considering the state estimation problem in the previous chapter, a general observation equation could be constructed as follows,

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v}, \tag{3.27}$$

where $\boldsymbol{x}$ is the state variable and $\boldsymbol{v}$ is the noise item. $\boldsymbol{z}$ is the measurements, corresponding to the foot point $\hat{\boldsymbol{p}}^f$. $\boldsymbol{h}(\cdot)$ is a nonlinear measurement model and corresponds to (3.9) in Chapter 3.2.3, which can be linearized by a first-order Taylor expansion as shown in Chapter 3.2.4 to get an approximated linear model,

$$\hat{\boldsymbol{z}} = \boldsymbol{J}\Delta\boldsymbol{x} + \boldsymbol{v} \tag{3.28}$$

where $\boldsymbol{J}$ is the Jacobian matrix. $\hat{\boldsymbol{z}} = \boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}_0)$ is the shifted measurement according to the operating point $\boldsymbol{x}_0$ (initial guess of state variables). Here, we assume that the error introduced by the linearization is negligible. As in typical visual localization scenarios, the frame frequency is usually high and the motion between successive frames is relatively small, making the operating point close to the state convergence value. [129]. However, it is worth noting that this assumption may not always hold in cases of abrupt or large camera motions, where the linearization error could become non-negligible.

Generally, the noise item $\boldsymbol{v}$ is assumed to be a Gaussian distribution with zero mean and a known covariance matrix $\boldsymbol{Q}$,

$$\boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{Q}). \tag{3.29}$$

However, in this section a fixed bias $\boldsymbol{b}$ is added into the error model as

$$\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{Q}). \tag{3.30}$$

The bias $\boldsymbol{b}$ represents another source of error, called outliers. The details will be discussed in the section 3.3.4.

For the linear problem in (3.28), we can solve it by the weighted least squares method, and the solution is

$$\Delta\hat{\boldsymbol{x}} = (\boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{J})^{-1} \boldsymbol{J}^\top \boldsymbol{W} \hat{\boldsymbol{z}}, \tag{3.31}$$

where $\boldsymbol{W}$ is the weighted matrix or information matrix and is the inverse of noise covariance matrix $\boldsymbol{Q}$, Note that the Jacobian matrix $\boldsymbol{J}$ could be updated in each iteration.

3.3.2 Outlier Rejection

Referring to Zhu's work [50], we could briefly classify the errors associated with the raw measurements (line features in this work) in the proposed framework as follows. The first category is the photometric noise caused by camera calibration error, camera over-exposure, and motion blur. These errors will affect the detection accuracy of line features and are expressed as the covariance matrix $\boldsymbol{Q}$ of the error model in (3.30). The second category is the fixed estimation deviations introduced by outliers, and we use the bias $\boldsymbol{b}$ in (3.30) to represent the effect of the outliers on the state estimation. However, it is undeniable that the presence of outliers is associated with the first category, as the category can cause feature misdetection and mismatching.

Due to the existence of noise and bias in measurements, the results of state estimation could contain residual $\boldsymbol{\varepsilon}$. Combining (3.28) and (3.31), the residual $\boldsymbol{\varepsilon}$ can be written as

$$\begin{aligned} \boldsymbol{\varepsilon} &= \hat{\boldsymbol{z}} - \boldsymbol{J}\Delta\hat{\boldsymbol{x}} = (\boldsymbol{I}-\boldsymbol{P})\hat{\boldsymbol{z}} \\ \boldsymbol{P} &= \boldsymbol{J}(\boldsymbol{J}^\top \boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^\top \boldsymbol{W} \end{aligned} . \tag{3.32}$$

In the actual calculation, the residual is obtained from the optimization solver in the previous section.

To evaluate the quality of measurements, a weighted sum of the squared errors (WSSE) [49] based on the residual is defined as

$$\boldsymbol{\varepsilon}^\top \boldsymbol{W}\boldsymbol{\varepsilon} = [(\boldsymbol{I}-\boldsymbol{P})]\hat{\boldsymbol{z}}^\top \boldsymbol{W}[(\boldsymbol{I}-\boldsymbol{P})\hat{\boldsymbol{z}}], \tag{3.33}$$

which can be further simplified as

$$\begin{aligned} \boldsymbol{\varepsilon}^\top \boldsymbol{W} \boldsymbol{\varepsilon} \Rightarrow \hat{\boldsymbol{z}}^\top \boldsymbol{W}(\boldsymbol{I}-\boldsymbol{P})\hat{\boldsymbol{z}} = \hat{\boldsymbol{z}}^\top \boldsymbol{S}\hat{\boldsymbol{z}} \\ \boldsymbol{S} = \boldsymbol{W}(\boldsymbol{I}-\boldsymbol{J}(\boldsymbol{J}^\top \boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^\top \boldsymbol{W}) \end{aligned}. \tag{3.34}$$

with $\boldsymbol{P}, \boldsymbol{I}-\boldsymbol{P}$ are idempotent matrices [130]. Next, a Chi-squared test for WSSE is used to determine whether there are outliers in the measurements:

$$\hat{\boldsymbol{z}}^\top \boldsymbol{S}\hat{\boldsymbol{z}} > \chi^2_{1-\alpha}(n-m) \equiv TD, \tag{3.35}$$

where, $\chi^2_{1-\alpha}(n-m)$ is the $1-\alpha$ quantile of the central Chi-squared distribution with $n-m$ degrees of freedom (DoF) [10]. n is the number of measurements, m is the number of state variables, α is the *probability of false alarm* (P_{fa}) and is set to 0.05 in this study. TD is a threshold that can be obtained by checking the χ^2 distribution table according to the DoF and quantile $1-\alpha$. If

$$\hat{\boldsymbol{z}}^\top \boldsymbol{S}\hat{\boldsymbol{z}} > TD, \tag{3.36}$$

the probability of outliers existing in the measurements is $1-\alpha$ (95%).

Because of the properties of the χ^2 distribution and the error model (3.30), outliers in the measurements can be rejected by (3.35). If the bias $\boldsymbol{b}$ in (3.30) is not equal to 0, the WSSE will not fit a central χ^2 distribution but a non-central distribution. The non-centrality parameter is $\lambda \equiv \boldsymbol{b}^\top \boldsymbol{S}\boldsymbol{b}$ [10], which is shown as

$$E(\hat{\boldsymbol{z}}^\top \boldsymbol{S}\hat{\boldsymbol{z}}) - (n-m) = \boldsymbol{b}^\top \boldsymbol{S}\boldsymbol{b}. \tag{3.37}$$

Therefore, the central Chi-squared distribution in (3.35) cannot be satisfied only if all outliers are theoretically removed from the observations (i.e. the bias in the error model is equal to 0). In this way, a greedy algorithm [118] can be applied to iteratively exclude the observations corresponding to the largest residuals (considered to be outliers) until the (3.35) is satisfied.

3.3.3 The Proof of Central χ^2 Distribution of WSSE withtout Outliers

Under the assumption of being without outliers, the detailed proof of the central Chi-squared (χ^2) distribution of WSSE is shown below. As mentioned in (3.34), the matrix $\boldsymbol{P}$ and $\boldsymbol{I}-\boldsymbol{P}$ are idempotent matrices. According to the properties of idempotent matrix, we have

$$\begin{gathered} trace(\boldsymbol{P}) = rank(\boldsymbol{P}) = rank(\boldsymbol{J}) = n \\ rank(\boldsymbol{I}-\boldsymbol{P}) = track(\boldsymbol{I}-\boldsymbol{P}) = trace(\boldsymbol{I}) - trace(\boldsymbol{P}) = m-n \end{gathered}, \tag{3.38}$$

with m is the number of measurements, n is the DoF of states. Meanwhile, based on the properties of matrix trace, the WSSE in (3.33) can be expressed by

$$\boldsymbol{\varepsilon}^\top \boldsymbol{W} \boldsymbol{\varepsilon} \Rightarrow \hat{\boldsymbol{z}}^\top \boldsymbol{W}(\boldsymbol{I}-\boldsymbol{P})^2 \hat{\boldsymbol{z}} = trace(\boldsymbol{V}\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^\top \boldsymbol{V}^\top) = \boldsymbol{u}^\top \boldsymbol{u}, \tag{3.39}$$

where $\boldsymbol{V} = \boldsymbol{L}(\boldsymbol{I}-\boldsymbol{P})$. $\boldsymbol{W} = \boldsymbol{L}^\top \boldsymbol{L}$ is the Cholesky decomposition of $\boldsymbol{W}$, because the information matrix is symmetric positive definite. Among (3.39), we have

$$\boldsymbol{u} = \boldsymbol{L}(\boldsymbol{I}-\boldsymbol{P})\hat{\boldsymbol{z}}, \tag{3.40}$$

since $\hat{\boldsymbol{z}} \sim \mathcal{N}(0, \boldsymbol{Q})$, $\boldsymbol{u} \sim \mathcal{N}(0, \boldsymbol{U})$ comes from the linear transformation of Gaussian distribution with $\boldsymbol{U} = \boldsymbol{L}(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{Q}\boldsymbol{L}(\boldsymbol{I}-\boldsymbol{P})^\top$. Combining (3.38), it can be seen that WSSE is the sum of squares of zero-mean Gaussian distribution with $(m-n)$ DoFs, which is consistent with the definition of the central χ^2 distribution [131].

3.3.4 Protection Level Calculation

Even if the Chi-squared test is passed, one or more outliers may still remain in the measurements and the existence of the potential outliers is denoted by the P_{fa}. Therefore, the error of the localization results based on the surviving measurements is mainly caused by two factors: (1) the potential missed outliers described by the probability P_{fa}. (2) the Gaussian noise associated with the surviving measurements.

The general expression of PL can be expressed as

$$\mathcal{E}_{bound} = \mathcal{E}_{out} + \mathcal{E}_{n}. \tag{3.41}$$

In particular, the first term is used to predict the bias-induced error in localization, and the second term is used to bound the noise-induced error.

Noise-induced Error

When only the error caused by the random noise is considered, the impact of the measurement noise on the state estimation can be found from (3.31). Due to $\hat{\boldsymbol{z}} \sim \mathcal{N}(0, \boldsymbol{Q})$, $\Delta\hat{\boldsymbol{x}}$ is also described by a standard Gaussian distribution according to its linear propagation theory, we have

$$\Delta\hat{\boldsymbol{x}} \sim \mathcal{N}(0, \boldsymbol{\Sigma}), \tag{3.42}$$

substituting (3.31), the covariance matrix $\boldsymbol{\Sigma}$ of $\Delta\hat{\boldsymbol{x}}$ can be computed by

$$\begin{aligned} cov(\Delta\hat{\boldsymbol{x}}) &= \mathbf{E}(\Delta\hat{\boldsymbol{x}}\Delta\hat{\boldsymbol{x}}^\top) \\ &= \mathbf{E}[(\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^\top\boldsymbol{W}\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^\top\boldsymbol{W}^\top\boldsymbol{J}(\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-\top}] \\ &= (\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^\top\boldsymbol{W}\mathbf{E}(\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^\top)\boldsymbol{W}^\top\boldsymbol{J}(\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-\top} \\ &= (\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-1} \end{aligned}, \tag{3.43}$$

with $\mathbf{E}(\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}^\top) = \boldsymbol{Q} = \boldsymbol{W}^{-1}$. Then, the state estimation error induced by measurement noises can be calculated by

$$\mathcal{E}_{ni} = k\sqrt{[(\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-1}]_{j,j}}, \tag{3.44}$$

where j is the index of the state variable, such as $j = 1,2,3$ mean the x-direction, y-direction, and z-direction, respectively. When $k = 3$, (3.44) is the typical 3σ

method, the corresponding probability of the values that lie within the 3σ interval is 99.73% [132]. This method is utilized to calculate the state error bound (i.e. PL) induced by measurement noise.

Outlier-induced Error

This study aims at considering the case where multiple outliers are contained in observations even after the Chi-squared test. Referring to RAIM and [118], we divide the translation into the x, y, z and rotation into the $roll$, $pitch$, yaw when calculating the PL in the proposed method, and then use an index vector formed by 0 and 1 to represent a state variable, for example

$$\begin{aligned} x &: \boldsymbol{H}_1 = [1,0,0,0,0,0]; roll : \boldsymbol{H}_4 = [0,0,0,1,0,0] \\ y &: \boldsymbol{H}_2 = [0,1,0,0,0,0]; pitch : \boldsymbol{H}_5 = [0,0,0,0,1,0]. \\ z &: \boldsymbol{H}_3 = [0,0,1,0,0,0]; yaw : \boldsymbol{H}_6 = [0,0,0,0,0,1] \end{aligned} \tag{3.45}$$

In terms of the first term in PL in (3.41), the bias vector for the i^{th} state can be given based on error propagation and (3.31) [10]:

$$\boldsymbol{b}_j^* = \boldsymbol{H}_j(\boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{J})^{-1} \boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{b}, \tag{3.46}$$

so the norm of the bias vector is

$$\begin{aligned} \|\boldsymbol{b}_j^*\| &= \sqrt{\boldsymbol{b}^\top \boldsymbol{D}_j \boldsymbol{b}} \\ \boldsymbol{D}_j &= \boldsymbol{W} \boldsymbol{J}(\boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{J})^{-1} \boldsymbol{H}_j^\top \boldsymbol{H}_j (\boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{J})^{-1} \boldsymbol{J}^\top \boldsymbol{W} \end{aligned}. \tag{3.47}$$

For the multiple-outliers case, assuming there are n measurements in total, r of them have biases, and $1 \leqslant r \leqslant (n-m)$, for example $n = 7$, $r = 2$. Since we do not know which two observations are true outliers, An index matrix cluster $\boldsymbol{A}_k$ composed of 0 and 1 with size $n \times r$ is defined to choose the two outliers from the seven observations, where the observation referred by number 1 is an outlier. For instance,

$$\boldsymbol{A}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top, \tag{3.48}$$

which means the first and second measurements are outliers. In this case, matrix cluster $\boldsymbol{A}_k$ need to iterate through all combinations of "two of seven", which is denoted by $C_7^2 = \frac{7\times 6}{2!} = 21$, so $j = 21$ this cluster has 21 different matrices. Then, the biases in observations can be selected iteratively by

$$\boldsymbol{b}_{jk} = \boldsymbol{A}_k \boldsymbol{b}_j, \tag{3.49}$$

To make sure the PL is to bound the error caused by biases, the maximum bias-induced error results from the example $n = 7$, $r = 2$ needs to be calculated. However, since the Chi-squared test has been passed, the condition $\boldsymbol{b}^\top \boldsymbol{S}\boldsymbol{b} \leqslant TD$ also needs to be satisfied [10]. Based on (3.47) and (3.49) , a constrained optimization model can be constructed as

$$\begin{aligned} \mathcal{E}_{out_j} &= \max \boldsymbol{b}_{jk}^* = \sqrt{\boldsymbol{b}_{jk}^\top \boldsymbol{D}_j \boldsymbol{b}_{jk}} \\ &= \max_{\boldsymbol{A}_k \in \boldsymbol{A}_r, \boldsymbol{b}_j} \sqrt{\boldsymbol{b}_j^\top \boldsymbol{A}_k^\top \boldsymbol{D}_j \boldsymbol{A}_k \boldsymbol{b}_j}, \\ & s.t. \boldsymbol{b}_j^\top \boldsymbol{A}_k^\top \boldsymbol{S} \boldsymbol{A}_k \boldsymbol{b}_j \leqslant TD \end{aligned} \tag{3.50}$$

where, $\boldsymbol{S}$ and $\boldsymbol{D}_i$ can be found in (3.34), (3.47). According to [10], this optimization question can be equally simplified by linear algebra theory as follows,

$$\max \boldsymbol{b}_{jk}^* = \max_{\boldsymbol{A}_k \in \boldsymbol{A}_r} \sqrt{\Lambda_{\max}(\boldsymbol{A}_k, \boldsymbol{D}_j, \boldsymbol{S})\Gamma}, \tag{3.51}$$

where, $\Gamma = TD$, $\Lambda_{\max}(\boldsymbol{A}_k, \boldsymbol{D}_j, \boldsymbol{S})$ is the largest eigenvalue of

$$\boldsymbol{A}_k^\top \boldsymbol{D}_j \boldsymbol{A}_k (\boldsymbol{A}_k^\top \boldsymbol{S} \boldsymbol{A}_k)^{-1}. \tag{3.52}$$

By iterating through all $\boldsymbol{A}_k$ and comparing the corresponding largest eigenvalue of (3.52), we can obtain the maximum bias-induced error in this direction.

On the other hand, the noise-induced error can be computed by 3σ method in (3.44), Substituting (3.51),(3.44) into (3.41), we got

$$\mathcal{E}_{bound\,j} = \max_{\boldsymbol{A}_k \in \boldsymbol{A}_r} \sqrt{\Lambda_{\max}(\boldsymbol{A}_k, \boldsymbol{D}_j, \boldsymbol{S})\Gamma} + k\sqrt{[(\boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{J})^{-1}]_{j,j}}. \tag{3.53}$$

The integral calculation of PL about j^{th} state variable can be obtained from the above equation.

3.4 Experiments & Discussions

The proposed framework is evaluated in real-world environments using both public indoor datasets for the UAVs and outdoor datasets collected by UGVs. We compare the performance of the proposed method with other state-of-the-art methods to evaluate the proposed optimization model and the safety-quantification method. The root mean squared error (RMSE) of absolute trajectory error (ATE) [2] is the evaluation criterion of algorithm accuracy. All experiments are implemented on a desktop with Intel-Core i9-12900K and Ubuntu 20.04. The proposed framework is developed by the robot operating system (ROS) [133] and C++ language. In terms of the evaluation of the localization performance, the algorithms compared include:

1. ***VINS***: the VINS framework developed in [1].
2. ***Benchmark***: the line-to-line visual localization method developed in [36].
3. ***PPL***: the proposed visual localization method without outlier rejection.
4. ***PPL-OR***: the proposed visual localization method with outlier rejection.

To evaluate the proposed safety-quantification function, the estimated PL is compared with the exact localization error. The localization system is considered safe when the estimated PL effectively bounds the actual localization error. The comparison methods include:

1. ***3σ***: a typical method used to represent the uncertainty of the state [118], the computational model can be seen in (3.44).

2. ***PL***: the proposed safety-quantification model.

3.4.1 UAV Indoor Experiment

Dataset & Comparison Methods

The EuRoc MAV (Micro Aerial Vehicle) dataset [134] is selected for evaluation, which is collected by global shutter stereo cameras (Aptina MT9V034, WVGA burri2016euroc, 20 FPS), synchronized IMU unit (ADIS16448, 200 Hz). The 3D point cloud prior maps are constructed by a Leica MS50 with an accuracy of about 1 mm, and the ground-truth states are obtained by a Vicon (100 Hz) and Leica MS50. The dataset includes six trajectories in two different rooms (V1 and V2), and there are three separate tracks of varying difficulty in each room. The MAV can be seen in Fig. 3.1. In particular, our current algorithm applies the direct output of VINS without loop-closure for state initialization. However, in the proposed framework, the 3D point cloud prior map is introduced to compensate for VINS drift while obtaining absolute localization results.

Verification of Optimization Model & Outlier Rejection

The comparison results of the pose estimation accuracy of VINS [1], benchmark [36], and ours are presented through the RMSE of ATE in Table 3.1. Specifically, the average accuracy improvements of ours over benchmark are 27.1% and 22.8% with and without outlier rejection (OR). Among that, both the PPL (our method without OR module) and the benchmark utilize line features for state estimation. The key distinction is that PPL employs a newly proposed line-based optimization model, which differs from the model used in the benchmark. This experimental

Table 3.1: RMSE [2] of ATE (m)

Sequence	VINS [1]	Benchmark [36]	PPL [ours]	PPL-OR [ours]
V1_01_esay	**0.098**	0.164	0.153	0.152
V1_02_medium	0.110	0.152	**0.092**	0.099
V1_03_difficult	0.189	0.217	0.186	**0.161**
V2_01_esay	**0.096**	0.192	0.185	0.166
V2_02_medium	0.167	0.281	0.155	**0.136**
V2_03_difficult	0.327	0.441	0.319	**0.312**

RMSE of ATE: The root mean squared error of absolute trajectory error.

result validates the proposed optimization model has better convergence performance than benchmark. The medium and difficult datasets in Table 3.1 refer to the fast flying speed and trajectory maneuverability of UAVs, which is challenging for visual-based localization methods. As shown in Table 3.1, in medium and difficult scenarios, the accuracy of our method with OR (PPL-OR) is 12.0% and 35.4% better than VINS and benchmark, respectively. However, in easy sequences, the initial alignment error introduced by the transformation between VIO and prior map outweighs its marginal benefit in drift correction. This means that our method based on structured lines is more resistant to disturbances in complex motion scenarios.

In addition, the proposed method with outlier rejection (PPL-OR) has generally a better performance than PPL, which reflects the effectiveness of the Chi-squared test-based outlier rejection algorithm. Details about the OR effectiveness are shown in Fig. 3.6, which displays the variation of WSSE of four keyframes as the outliers are detected and removed by a greedy algorithm sequentially. The x-axis is the number of outliers removed, and the y-axis represents the residual of the observation model. By sequentially rejecting the visual outliers, the WSSE decreases.

Verification of PL

This subsection presents a protection level-based safety quantitative analysis by

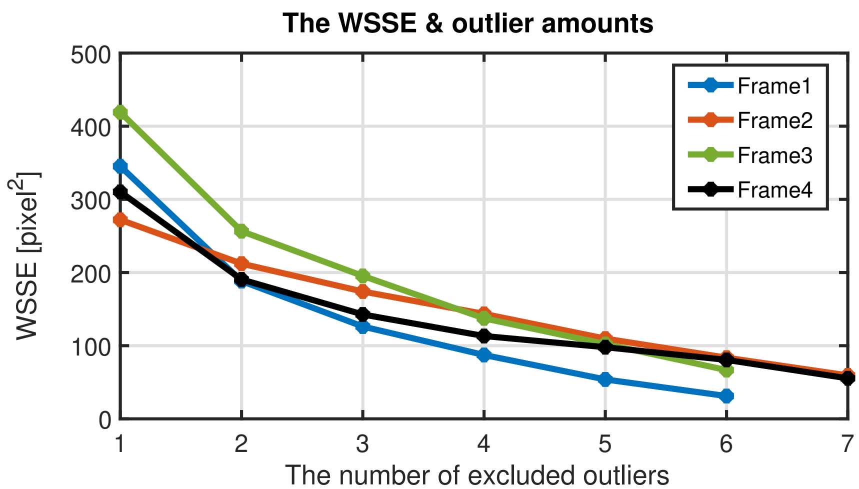


Figure 3.6: The plot of the relevance WSSE and the number of outliers. The 4 frames are selected from the V1_02_medium dataset. As the outliers are detected and excluded in turn, the residual gradually decreases.

taking the classic V1_02_medium sequence as an example. In this experiment, we compare the bound rate of PL and 3σ (3.44) [118] on state estimation errors, the bound rate b_r can be computed as

$$b_r = \frac{n}{m} \tag{3.54}$$

where, n is the number of $PL \geqslant error$ or $3\sigma \geqslant error$, separately. m is the number of all frames. If the estimated PL could bound the actual error of localization solution, it indicates that the proposed safety quantification is effective.

The results of translation and rotation are shown in Fig. 3.7 and Fig. 3.8, respectively, expressed by XPL, YPL, ZPL, RollPL, PitchPL, and YawPL. In Fig. 3.7, it is clear PL performs much better than 3σ for error bounding, with PL having more than 70% bound rate in all three directions, while 3σ is only about 20% (see Table 3.3). And it can be seen that PL has a similar trend to the error in most cases. In Fig. 3.8, PL quantifies the error significantly better than 3σ, but its performance is uneven in three directions, especially in yaw rotation. The main reasons for this

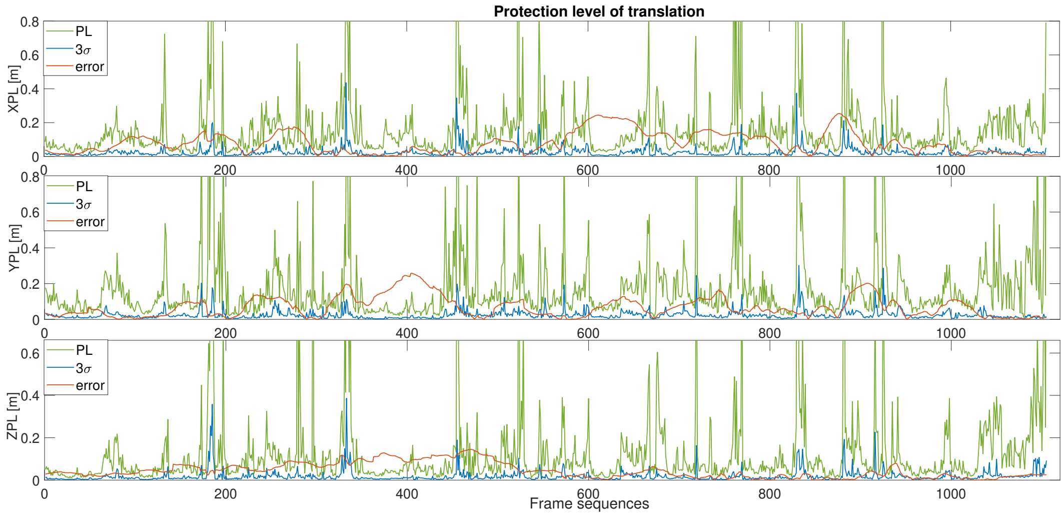


Figure 3.7: The comparison results of protection level, 3σ, and estimation error in translation under V1_02_medium sequence. The measurement covariance is 7-pixel^2 and the number of outliers is set to two.

phenomenon are the limitations of the line features and the IMU sensor. In terms of line features, since there are many horizontal lines in space, when the object is rotated around z-axis to produce *yaw* rotation, the motion will be parallel to the lines making the motion useless for feature matching and state optimization, which is one of the most common degradation symptoms of line features [33].

The Discussion on PL

Ideally, the PL should bound the actual localization error; however, in practice, uncertainties such as unknown outliers and imperfectly modeled observation noise may limit its ability to fully bound the error. Based on the derivation of PL in Chapter 3.3.4, the calculation of PL is affected by bias-induced error resulting from outliers and noise-induced error because of observation noises. Firstly, the number of outliers existing in the current observations after passing the Chi-squared test is unknown. Secondly, the noise covariance matrix of the visual measurements cannot be obtained exactly, which is given empirically in this study. All these uncertainties are factors that can affect the performance of PL estimation.

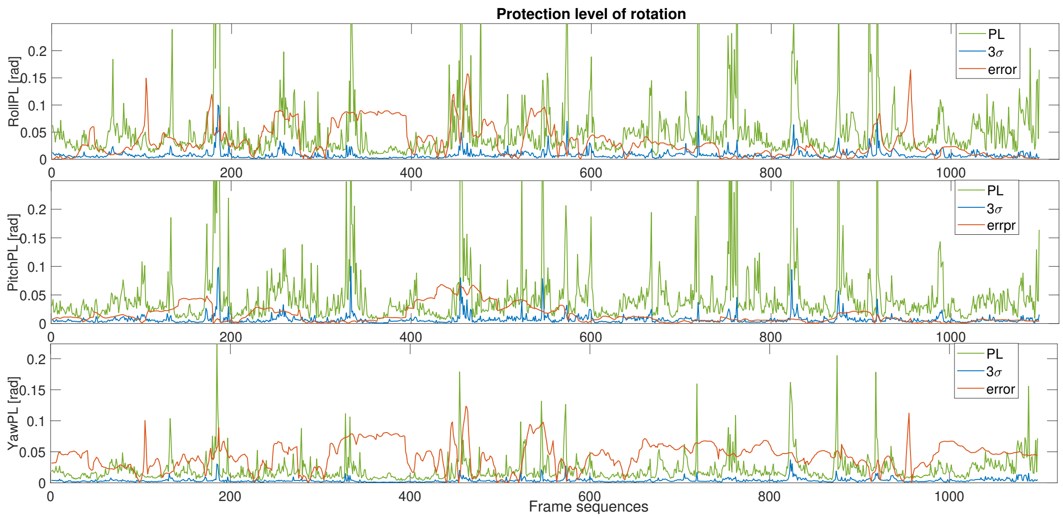


Figure 3.8: The comparison results of protection level, 3σ, and estimation error in rotation under V1_02_medium sequence. The measurement covariance is 7-pixel^2 and the number of outliers is set to two.

Table 3.2: The b_r (%) of PL / 3σ with error using different number of outliers in translation and rotation. The covariance is 7-pixel^2.

Num.	1	2	3
x	**71.6%**/ 19.6%	**74.2%**/ 19.6%	**74.2%**/ 19.6%
y	**75.7%**/ 24.8%	**76.8%**/ 24.8%	**76.8%**/ 24.8%
z	**71.3%**/ 22.9%	**73.5%**/ 22.9%	**73.5%**/ 22.9%
$roll$	**43.9%**/ 20.8%	**65.2%**/ 20.8%	**65.2%**/ 20.8%
$pitch$	**66.6%**/ 31.0%	**85.3%**/ 31.0%	**85.3%**/ 31.0%
yaw	**8.23%**/ 1.99%	**18.8%**/ 1.99%	**18.8%**/ 1.99%

To gain more insight into PL, we evaluate the bound rate of PL and 3σ under different assumptions with multiple outliers and noise covariances in Table 3.2 and Table 3.3, respectively. From Table 3.2, it can be seen that increasing the number of outliers can improve the bound rate of PL, but when the number of outliers is set to 3 or more, it will not change, indicating that the largest number of outliers in all sequences is 2. In Table 3.3, the measurement noise of the line feature is set to 3, 5, and 7 pixels, respectively. In particular, the measurement noise of a single pixel point feature is usually set between 1-3 pixels [135]. Since this work uses line

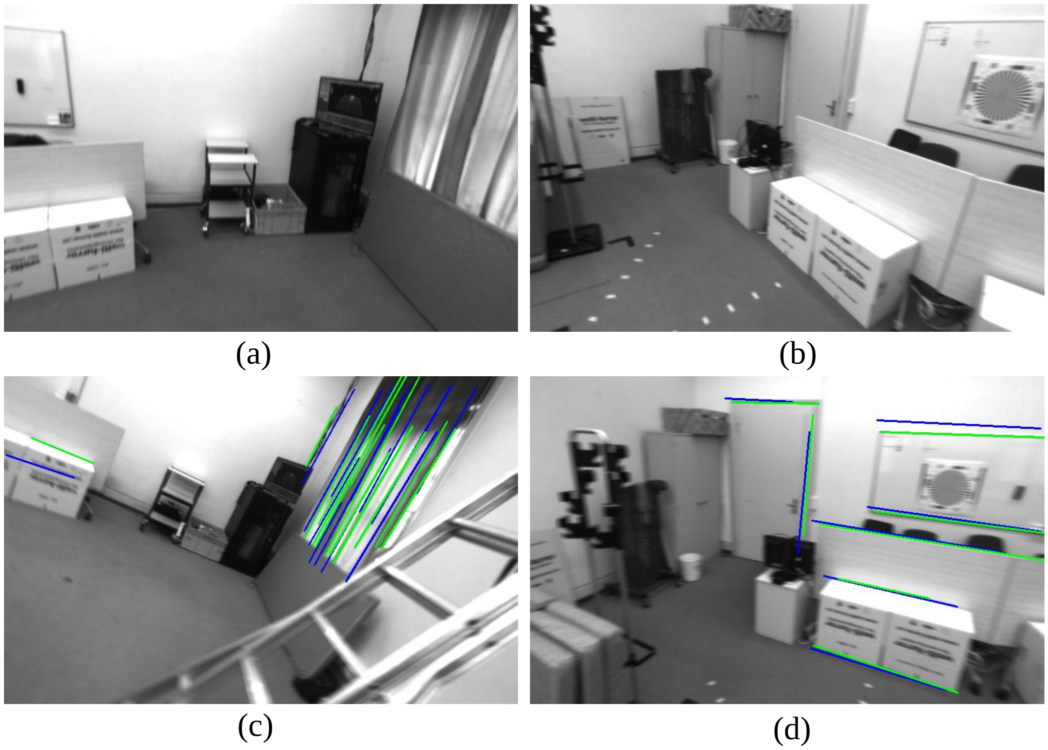
Table 3.3: The b_r (%) of PL / 3σ with error using different covariance assumptions in translation and rotation. The number of outliers is two.

Cov.	3-pixel^2	5-pixel^2	7-pixel^2
x	**42.5%**/ 7.51%	**59.8%**/ 13.5%	**74.2%**/ 19.6%
y	**47.1%**/ 10.0%	**64.2%**/ 18.3%	**76.8%**/ 24.8%
z	**48.0%**/ 11.8%	**61.3%**/ 16.0%	**73.5%**/ 22.9%
roll	**36.5%**/ 11.0%	**51.8%**/ 15.3%	**65.2%**/ 20.8%
pitch	**60.3%**/ 14.8%	**74.8%**/ 22.5%	**85.3%**/ 31.0%
yaw	**6.03%**/ 0.81%	**10.9%**/ 1.53%	**18.8%**/ 1.99%

features, the detection of lines will introduce more errors than point features. Also, the 3D line features extracted from the point cloud map could impose observation error, so the covariance assumption is amplified here. As shown in Table 3.3 , the b_r for both the PL and 3σ increase with the growing covariance of the line features, which is consistent with the theoretical expectation that their computational formulas both include a covariance term. Moreover, PL demonstrates a sharper response to this change than 3σ, reflecting its advantage in characterizing how uncertainty evolves with observation quality. In summary, more outliers and larger covariance increase PL, which is consistent with the definition of PL for quantifying safety.

The PL & Feature Degeneration

It is well known that when discussing localization safety, it is inevitable to consider the impact of the feature quality on it [136]. Here, the quality of features can be considered in terms of the feature detection error and the feature distribution. In this study, PL is used as a metric for safety quantification, so we will discuss the impact of features on the reliability of safety quantification around anomalous PL values. The following analysis is done for two special cases. First, near frame 400 in Fig. 3.7, the PL is too conservative to bound the localization error. Two images in the area around frame 400 are shown in Fig. 3.9 (a) and (b). It is clear to see



(a) (b) (c) (d)

Figure 3.9: (a), (b) two examples of the motion blur. (c), (d) two examples of the feature degeneration. The blue and green lines are the feature matching pairs.

that images suffer from severe motion blur, which could potentially trigger additional detection errors in feature measurements [50]. Since the feature covariance that we assume during PL computation is not dynamically changing, PL is unable to accurately estimate localization errors in scenarios where feature detection errors increase, which means that the safety quantification is inaccurate when the feature detection error is suddenly large.

In the second case, the anomalous PL values were significantly larger than the localization error and were not informative, such as the frame 200. In addition to the feature detection errors described above, the spatial distribution of the line features utilized for the localization is the other source affecting the PL estimation. It is expected that the features are evenly distributed within the field of view, which could introduce ideal constraints. However, the features can flock together where the degeneration of the state estimation will occur [137]. To assess the impacts of the feature distribution on the localization performance, the condition number (CN) is introduced in [138]. In particular, CN is determined by the ratio between the maximum and minimum eigenvalues of $\boldsymbol{J}^\top \boldsymbol{J}$, which can be shown as

$$\kappa = \frac{\sigma_{max}}{\sigma_{min}} \tag{3.55}$$

where the $\boldsymbol{J}$ denotes the Jacobian matrix of the visual line constraint. A low condition number means to be well-conditioned, while a high condition number indicates ill-conditioning.

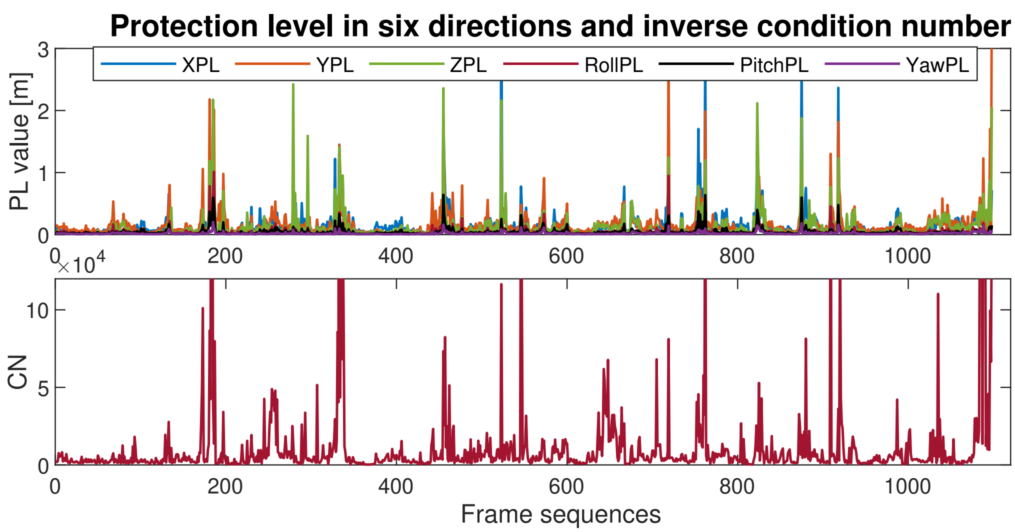


Figure 3.10: The comparison of condition number (CN) and protection level in six transformation directions. The assumption of the measurement covariance is 7-pixel and the number of outliers is set to two.

Fig. 3.10 displays the CN and PL in six directions. As seen from the figure, PL has a highly consistent trend with CN at the anomalous mutation, which indicates these frames have ill-conditional constraints and feature degeneration. Based on this discovery, we can conclude that feature degeneration largely causes exceptional maxima peaks in PL. On the other hand, we could utilize PL to remind the feature degeneration phenomenon during state estimation to avoid serious optimization failures. Fig. 3.9 (c) and (d) show two typical feature degeneration scenarios. It is clear that most of the features in images are parallel, which makes the constraint with its vertical direction very weak, resulting in an ill-conditioned case.

3.4.2 UGV Outdoor Experiment

Experiment Setup

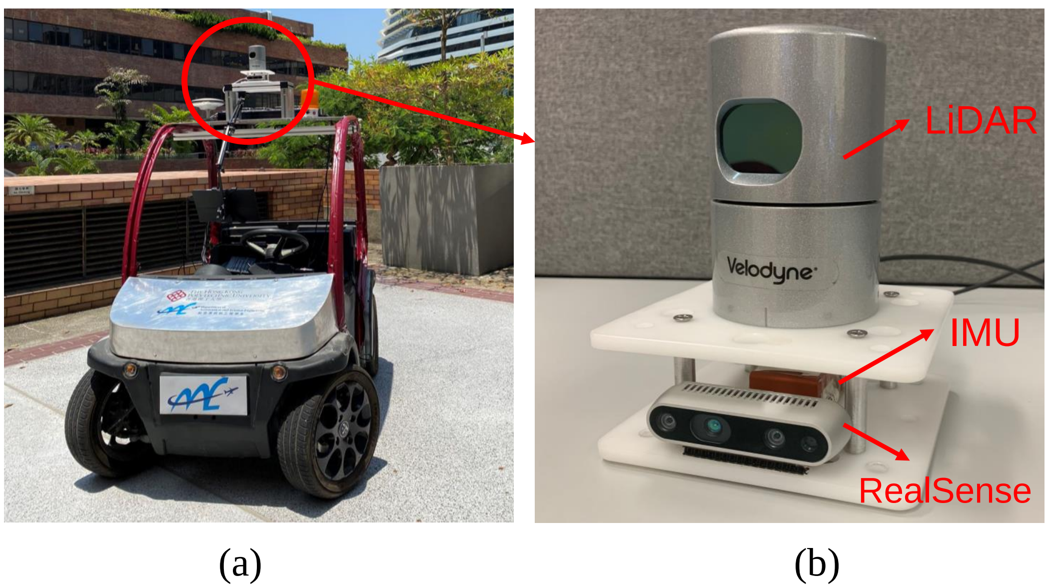
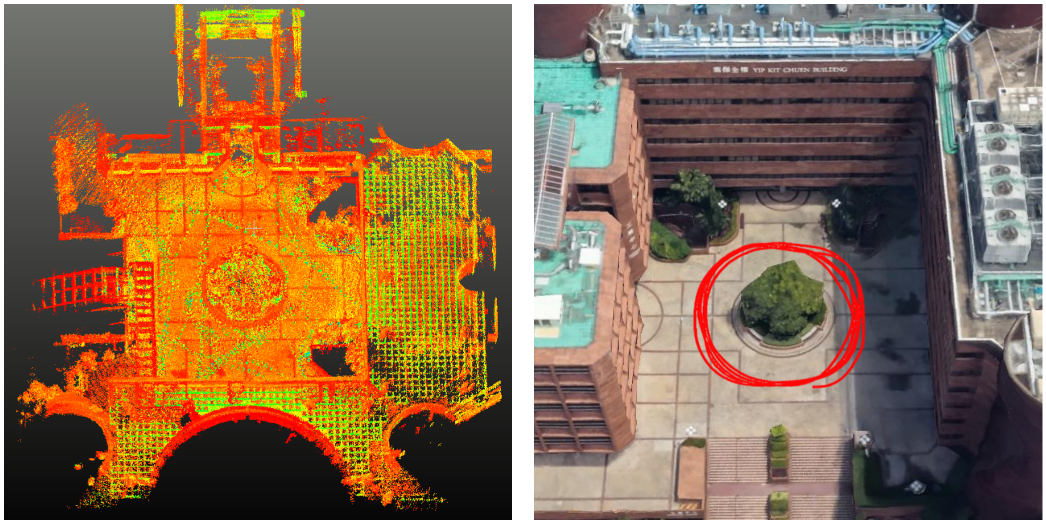


Figure 3.11: (a) The UGV platform loaded with customized sensor unit. (b) The detailed Sensor unit including LiDAR, IMU, and RealSense camera.

The UGV experiment is conducted on the CARLA simulator [139] and real-world environment in the Hong Kong Polytechnic University (PolyU) campus. The real-world UGV platform used in this experiment is shown in Fig. 3.11 (a). The detailed sensor module is displayed in Fig. 3.11 (b), which contains a LiDAR (HDL 32E Velodyne), an IMU (Xsens Mti 10), and a RealSense camera (D435i, 640$\times$480, 30 Hz). The dataset is collected by driving around a square on the Hong Kong Polytechnic University (PolyU) campus (Fig. 3.12). The total path length is 295.5



(a) (b)

Figure 3.12: (a) The regional point cloud of the Hong Kong Polytechnic University. (b) The corresponding Google 3D map of the same area. The red line indicates the trajectory of the UGV during the testing.

m. The IMU data and images captured by RealSense are the input of VINS. The data from LiDAR and IMU are processed by LIO_SAM implementation [16], and used to build a prior point cloud map shown in Fig. 3.12 (a), the origin of this 3D map is defined at the starting position of the IMU. The trajectory estimated by LIO_SAM is regarded as the ground truth of localization. With the help of the LiDAR loop closure, centimeter-level accuracy can be obtained from the LIO_SAM in the evaluated scene.

In CARLA simulator, a vehicle equipped with a 64-channels LiDAR (rotation frequency 100, range 200m), IMU (100Hz, measurement noise 10^{-2}, bias random walk 10^{-4}), and a monocular camera (resolution 900×600, FoV 90 degree) derived around a town to construct prior point cloud map and dataset. The view of the town and the trajectory of vehicles are shown in Fig. 3.13. In this town, we run three different routes with lengths of 1km (red trajectory labeled with CARLA_01), 2km (green trajectory labeled with CARLA_02), and 3km (blue trajectory labeled with

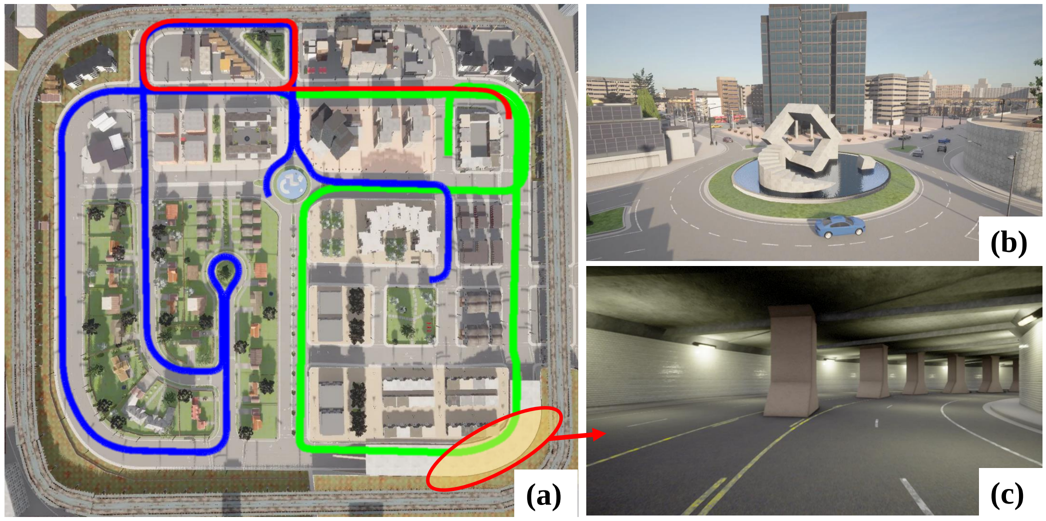


Figure 3.13: (a) The Carla simulation town and three driving trajectories with different lengths. (b) A scene in the town. (c) The tunnel scene marked by the red circle in the figure (a).

CARLA_03). The speed of the car is 30km/h. These three paths pass through a variety of scenarios, such as urban canyons, tunnels, and shrubs, which are used to verify the effectiveness of our algorithm compared to the VINS and benchmark. Accordingly, similar evaluation criteria and comparisons with UAV experiment are adopted in this section.

System Evaluation

Table 3.4: RMSE [2] of ATE (m)

Dataset	VINS [1]	Benchmark [36]	PPL [ours]	PPL-OR [ours]
PolyU	1.688	2.191	1.607	**1.602**
CARLA_01	2.380	3.626	2.137	**1.931**
CARLA_02	4.613	4.312	3.677	**3.321**
CARLA_03	7.894	7.986	7.733	**7.311**

RMSE of ATE: The root mean squared error of absolute trajectory error.

Table 3.4 lists the RMSE of ATE comparison results of the proposed method with VINS and benchmark. It can be seen that our method outperforms others in

CARLA simulated dataset and real-world environment. Specifically, as the running trajectory grows gradually in CARLA dataset, the proposed methods have higher accuracy than VINS by 18.91%, 27.98%, 5.10%, and than benchmark by 46.83%, 22.97%, 8.46% in 1-3km. Meanwhile, our method (PPL-OR) improves the localization accuracy by 5% and 26.9% over VINS and benchmark in the real-world environment. Fig. 3.14 presents the estimation errors of translation and rotation, it is clear our algorithm is able to reduce the drift problem of VINS to some extent, which is evident in the z and *yaw* motions, while the errors in other directions can be offset periodically because the UGV trajectory is a constantly repeating circular motion. However, as the path keeps getting longer and the VINS drift increases, the effectiveness of our algorithm for removing the cumulative error gradually diminishes (see Fig. 3.14) due to the fact that the proposed method uses the output of VINS as the initial guess.

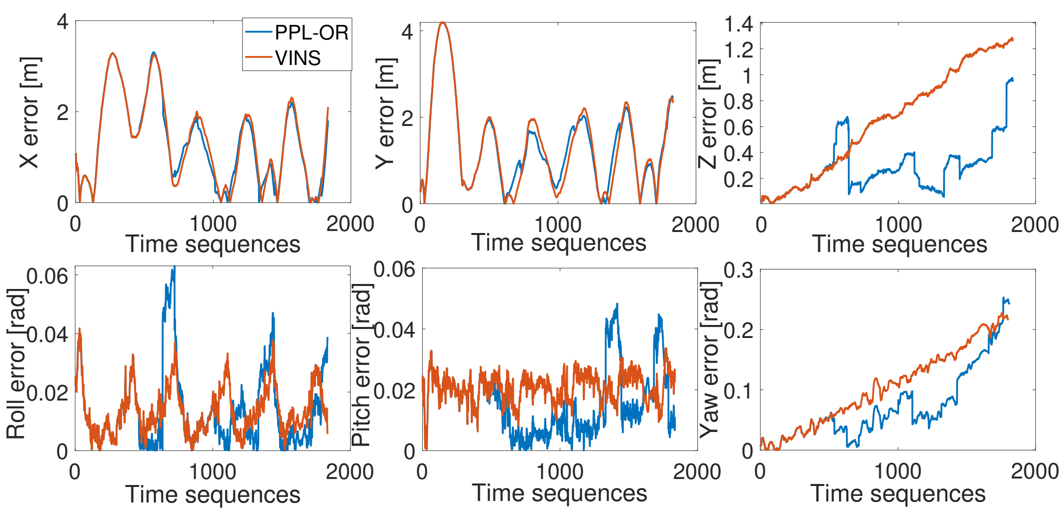


Figure 3.14: The position and orientation error of PPL-OR compared with VINS.

Additionally, the time statistics of VINS, benchmark, and our method on three datasets are displayed in Table 4.3, where Feature Processing includes the feature detection and matching modules. Optimization consists of pose estimation and

Table 3.5: Time statistics of Feature Processing / Optimization Treads (ms)

Datasets	VINS [1]	Benchmark [36]	PPL-OR [ours]
EuRoC	5.8 / 15.4	11.3 / 7.5	9.9 / 1.5
PolyU	2.9 / 23.3	5.6 / 4.8	4.8 / 1.3
CARLA_01	8.2 / 20.1	48.9 / 52.1	46.5 / 1.8

marginalization modules. The optimization computation of our method is fast due to the fact that we use fewer line constraints than point constraints during pose estimation as well as the proposed outlier rejection module refines the constraints to accelerate the convergence of the estimator. However, on the CARLA dataset, the feature processing module of the prior-map based methods has significantly increased in time consumption. This is because the prior line maps are large in size, which requires the algorithm to spend more time completing cross-modal matching. Overall, our algorithm satisfies the real-time requirement while loading a reasonable prior line map. Therefore, in practice, maintaining the size of the loaded prior map within a rational range is vital to the real-time performance of the method.

Table 3.6: The b_r (%) of PL with error using different numbers of outliers in translation and rotation with covariance is 7-pixel

Num.	x	y	z	$roll$	$pitch$	yaw
1	83.1%	66.2%	50.8%	64.6%	53.9%	5.13%
2	85.1%	69.2%	54.9%	66.7%	56.4%	5.64%
3	85.1%	69.2%	54.9%	66.7%	56.4%	5.64%

PL Result

Based on the discussion of PL in the previous subsection, the safety quantification of UGV in the real-world environment is verified by adjusting the number of outliers and the covariance of measurements. By fixing the covariance to 7 pixel^2, the relationship between the bound rate of PL and the number of outliers is shown in

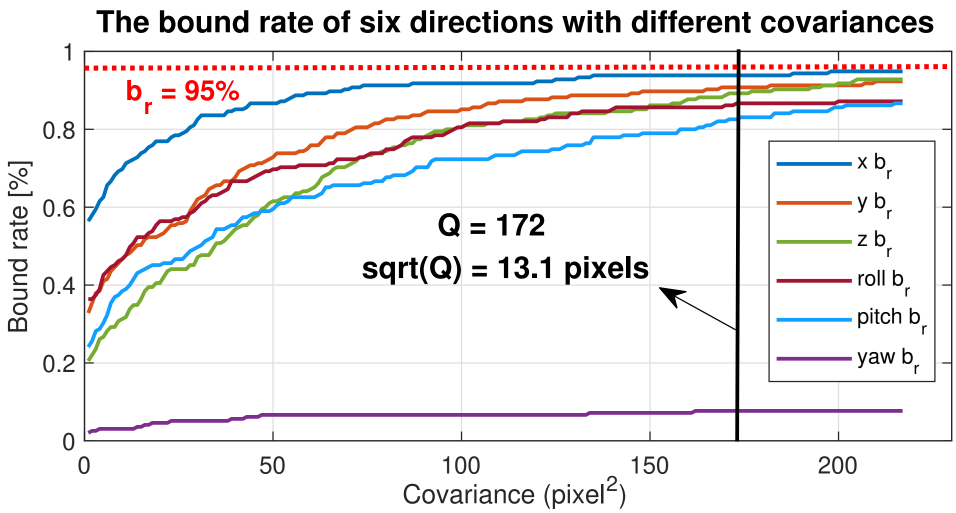


Figure 3.15: The bound rate (b_r) of six directions with different measurement noises from 3-pixel to 15-pixel. The red dash line means the $b_r = 95\%$. The covariance corresponding to the black line is 13.1-pixel^2.

Table 3.7: The b_r (%) of PL / 3σ under six transformation directions with covariance is 13.1-pixel^2 and outliers are 2.

	x	*y*	*z*	*roll*	*pitch*	*yaw*
PL	**93.9%**	**89.7%**	**88.2%**	**85.6%**	**80.5%**	**7.69%**
3σ	40.0%	16.4%	6.67%	20.5%	12.8%	0.51%

Table 3.6. It is clear that the number of residual outliers in the UGV dataset is two, which is the same as the UAV experiment. Next, by fixing the number of outliers to 2 and changing the covariance from 3-pixel^2 to 15-pixel^2, the trends of the bound rate of the PL in six directions are shown in Fig. 3.15. As seen from the figure, the bound rates of the PL in six directions gradually level off when covariance is equal to 13.1-pixel^2 (the black line). In particular, the bound rate in x-direction is infinitely close to 95% (the red dashed line), which is consistent with the probability $(1-\alpha)$ set in the outlier rejection section. Based on this figure, the final covariance in the UGV experiment is set to 13.1-pixel^2, and the corresponding bound rates of PL and 3σ are shown in Table 3.7. Fig. 3.16 and Fig. 3.17 are the trends of PL, 3σ,

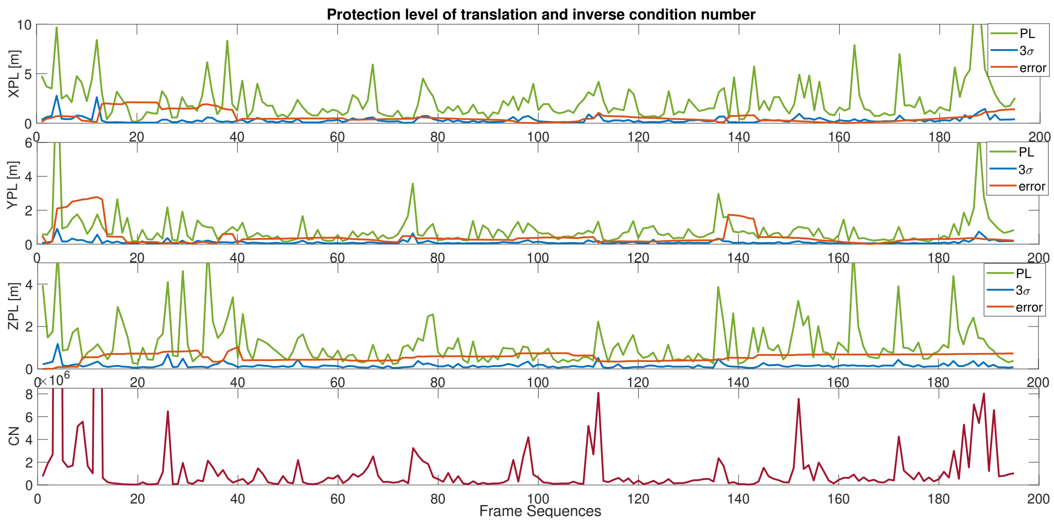


Figure 3.16: The CN and comparison of protection level, 3σ, and error in three translation directions. The assumption of the measurement covariance is 13.1-pixel^2 and the number of outliers is set to two.

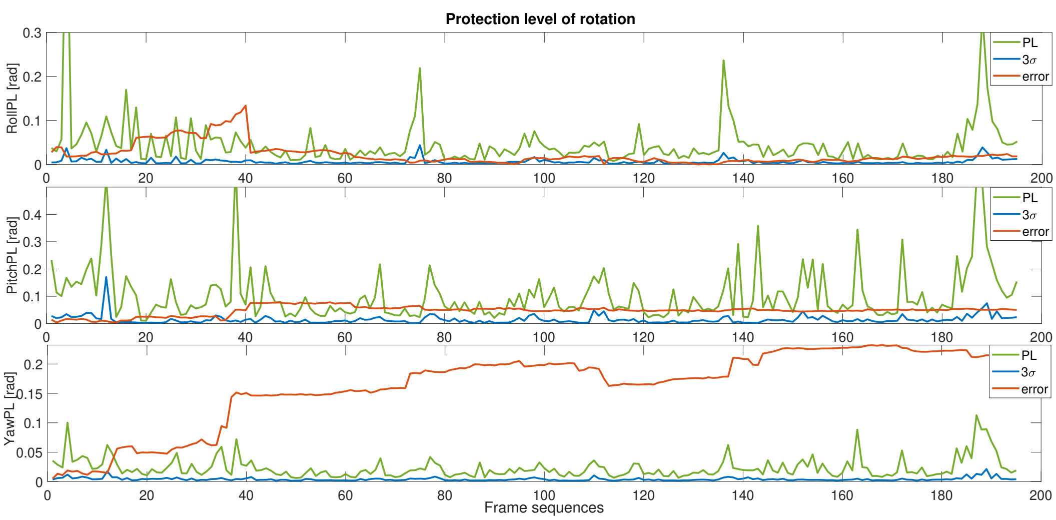


Figure 3.17: The comparison results of protection level, 3σ, and error in three rotation directions. The assumption of the measurement covariance is 13.1-pixel^2 and the number of outliers is set to two.

error, and the ICN in translation and rotation, respectively.

Discussion

In terms of accuracy improvement, the proposed method can eliminate the cumulative drift of VINS to a certain extent by using prior map-based localization. However, because of the loosely coupled form with VINS, the proposed method is greatly influenced by the initial guess of the state from VINS, and when the drift is severe, our method will have difficulty converging (Fig. 3.14). Furthermore, the measurement noise of the line feature in UGV dataset is about 13 pixels, which is even greater than the noise in UAV experiment. Excluding the noise of 2D line feature detection in images, the 3D line detection in point cloud map could induce more errors than UAV experiment because of the low-precision point cloud. It follows that the denseness and precision of the point cloud map are alse important factors affecting the performance of the algorithm and the PL calculation, as it impacts the line detection and the error model.

Meanwhile, the same to the UAV experiment, the performance of PL is particularly poor at the *yaw* rotation. Comparing the localization error and PL of the *yaw* axis with the *roll* and *pitch* axes in Fig. 3.8 and Fig. 3.17, it is obvious that the overall error of the yaw axis is larger than the others, and the PL is indeed smaller. Apart from the degeneration caused by the poor geometry of the visual line features, the error involved in the IMU is another source leading to the conservative estimation of the PL. On the one hand, the *yaw* angle of the IMU is not directly observable by the system [140]. On the other hand, the circular trajectory of the UGV experiment leads to additional challenges for the *yaw* estimation, which will excite more errors on the *yaw*. As a result, the accuracy of initial guess of the state from the VINS is not guaranteed, which can lead to poor estimation of the *yaw*. This phenomenon is shown in the *yaw* axis of Fig. 3.14, which presents a severe divergence trend. In summary, the above-mentioned reasons are leading to the poor performance of PL on the *yaw* axis.

3.5 Summary

This study presents a safety-quantifiable line feature-based monocular visual localization framework with point cloud maps. A new point-to-point line constrained optimization model is built to achieve accurate and robust localization in point cloud maps to eliminate cumulative drift occurring in relative positioning. Furthermore, a safety quantification strategy motivated by RAIM system is introduced into the current autonomous systems to evaluate the maximum value of the state estimation errors, including three translation and three rotation directions with multiple faults assumption. The experimental results on UAV and UGV platforms show the proposed method outperforms the comparison algorithms in accuracy and robustness, and the error quantification of PL for state estimation is efficient in the multiple outlier assumption as well as in the translational and rotational motions. We also provide the open-source implementation to benefit the community.

There is still potential for improving our algorithm. For instance, the current system's loosely coupled model with VINS has limitations in enhancing localization accuracy. Future work could explore a tightly coupled approach to improve precision and make fuller use of prior 3D map information, such as surface structures, to further refine the algorithm's performance. Moreover, the impact of prior map size on computational efficiency also needs to be considered. When the prior map has a large coverage area, the calculation can be accelerated by adjusting the map storage data structure. The performance of PL predicting localization errors can be improved by incorporating dynamic error models and feature quality. In this study, the computed PL is used solely for predicting state estimation errors. A promising direction for future research is to explore its broader applicability within robotic systems. Inspired by the GNSS domain, we could predefine a localization accuracy threshold and utilize PL-predicted errors to evaluate whether the current localization results meet operational requirements, enabling safer real-world robotic deployments. Additionally, PL's potential applications extend beyond localization; its error predictions could also be leveraged in mapping and planning processes, opening new avenues for exploration.

Chapter 4

Tightly-coupled Visual/Inertial/Map Localization and Observability Analysis

The loosely coupled framework presented in the previous chapter, while effective in integrating visual localization with prior point cloud maps, still exhibits several inherent limitations. First, the optimization relies heavily on the initial guess provided by the VIO subsystem, meaning that drift or biases accumulated in VIO directly influence the final pose estimation. Second, because the visual–inertial estimation and the map-based refinement are executed in two separate optimization stages, the system cannot fully exploit the joint constraints among visual, inertial, and prior-map measurements, and the repeated optimization also increases computational cost. In addition, the loosely coupled structure shows relatively weak estimation performance in the yaw direction, and the underlying cause requires further investigation to better understand the system's rotational behavior.

To address these limitations, this chapter introduces a tightly coupled framework that integrates all sensing modalities within a unified optimization process. Such a design enables the simultaneous use of all measurements, reduces reliance on VIO initial guesses, and allows the system to more effectively exploit the struc-

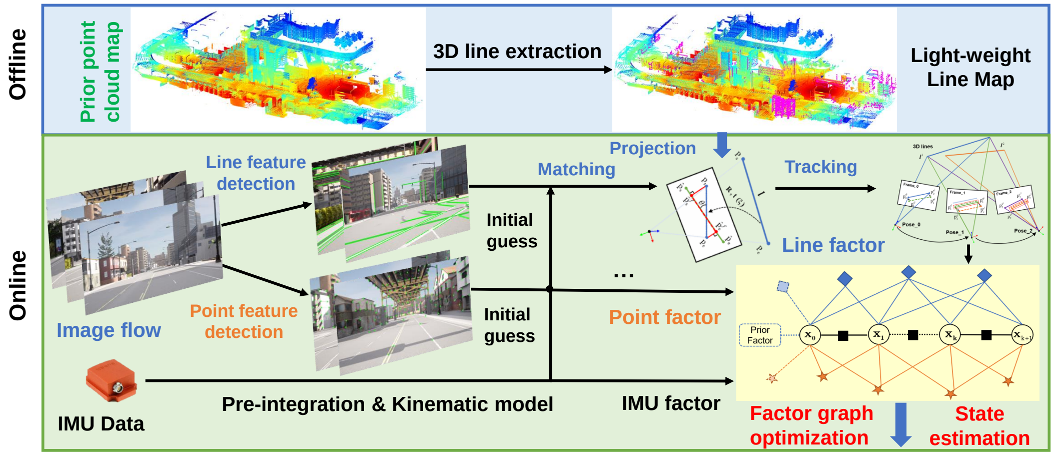


Figure 4.1: The flowchart of the proposed framework. The 3D lines in the prior map are detected offline, which are matched with the online detected 2D lines. After feature matching and tracking, the line correspondences, IMU measurements, and point feature pairs are integrated as constraints for factor graph optimization in a tightly coupled form.

tural information contained in the prior map. Furthermore, by incorporating prior-map constraints directly into the estimation pipeline, the proposed system provides a clearer foundation for analyzing the observability properties introduced by map-based measurements, thereby enabling a deeper understanding of the intrinsic relationships between these measurements and the system states.

4.1 System Overview

Building on the limitations identified above, this chapter adopts a tightly-coupled scheme that simultaneously optimizes visual, IMU, and prior-map measurements within a unified framework. Specifically, this framework associated the classic VIO optimization factors with the cross-modal line features-based constraints for pose estimation, which is shown in Fig. 5.1. In this framework, the 3D lines are extracted from the prior map offline, and then the original point cloud map can be condensed to a lightweight line map containing building edges and road segment markings.

The cross-modal line correspondences are obtained by line matching and a proposed line tracking strategy for outlier rejection. The novel line feature-based cost model proposed in Section 3 is tightly coupled with the IMU pre-integration factor and point feature factor for state estimation. Meanwhile, we justify the rationality of this novel model in this work. Besides, we analyze the observability of the proposed system to better understand the internal relationships between state variables and system measurements. To the best of our knowledge, we are the first group that analyze the observability of the prior map-aided VIO system.

This study utilizes IMU measurements, visual point features, and line features to construct constraints for nonlinear optimization-based state estimation. To improve stability and accuracy, the proposed method applies the sliding window structure, which can compose multi-frame constraints simultaneously and is the same as the classic VIO system [1]. Based on the sliding window pipeline, the states are defined as

$$\begin{aligned}
\mathcal{X} &= \left[\boldsymbol{x}_k, \ldots, \boldsymbol{x}_{k+N}, \boldsymbol{x}_c^b, \lambda_m, \ldots, \lambda_{m+M}\right] \\
\boldsymbol{x}_i &= \left[\boldsymbol{p}_{b_i}^w, \boldsymbol{v}_{b_i}^w, \boldsymbol{q}_{b_i}^w, \boldsymbol{b}_a, \boldsymbol{b}_g\right], i \in [k, k+N], \\
\boldsymbol{x}_c^b &= \left[\boldsymbol{p}_c^b, \boldsymbol{q}_c^b\right]
\end{aligned} \tag{4.1}$$

For clarity, the notations used in (4.1) are summarized as follows. The coordinate systems involved include the global world frame $(\cdot)^w$, the body (IMU) frame $(\cdot)^b$, and the camera frame $(\cdot)^c$. The rotation matrix $\boldsymbol{R}_b^a$ denotes the rotation from frame b to frame a. The estimated states in the localization problem include the body pose $\boldsymbol{T}_b^w = (\boldsymbol{p}_b^w, \boldsymbol{q}_b^w)$, velocity $\boldsymbol{v}_b^w$, IMU accelerometer bias $\boldsymbol{b}_a$, and gyroscope bias $\boldsymbol{b}_g$. The extrinsic parameters from the camera frame to the body frame are $\boldsymbol{p}_c^b$ and $\boldsymbol{q}_c^b$, while λ denotes the inverse depth of each point feature. Here, $N+1$ denotes the number of frames in the sliding window, and $M+1$ represents the number of point features observed within it.

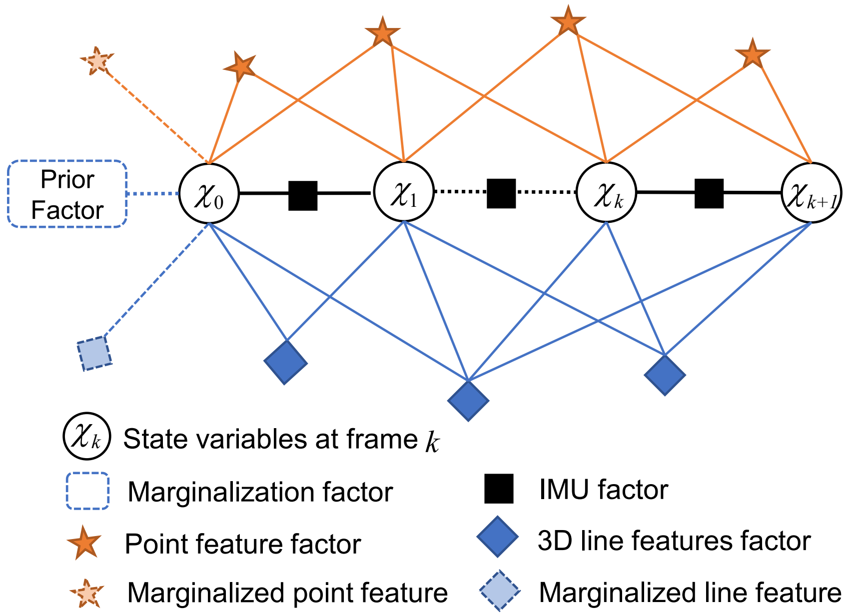


Figure 4.2: Factor graph structure in the proposed system. The different symbols indicate the prior, IMU, point, and line constraints used in the optimization problem.

The factor graph structure of the proposed optimization model is displayed in Fig. 4.2. The model can be expressed by

$$\begin{aligned}\mathcal{X}^* = \min_{\mathcal{X}} \Bigg\{ & \|\boldsymbol{r}_p - \boldsymbol{H}_p\mathcal{X}\|^2 + \sum_{k\in\mathcal{B}} \|\boldsymbol{r}_b(\boldsymbol{z}_b,\mathcal{X})\|^2_{\boldsymbol{Q}_b} \\ & + \sum_{k\in\mathcal{C}} \|\boldsymbol{r}_c(\boldsymbol{z}_c,\mathcal{X})\|^2_{\boldsymbol{Q}_c} + \sum_{k\in\mathcal{L}} \|\boldsymbol{r}_l(\boldsymbol{z}_l,\mathcal{X})\|^2_{\boldsymbol{Q}_l} \Bigg\},\end{aligned} \tag{4.2}$$

where, the four cost terms $\boldsymbol{r}_x$ are the prior factor, IMU measurement factor, point feature-based factor, and line feature-based factor. $\boldsymbol{z}_{(\cdot)}$ means the different measurements. It should be mentioned that the prior factor comes from the marginalized constraints shown in Fig. 4.2 and $\boldsymbol{H}_p$ is the Jacobian matrix of the prior factor. The construction of the other factors is derived in the next section.

4.2 State Estimation with Multiple Factors

4.2.1 IMU Pre-integration

For monocular visual localization systems, the addition of IMU can provide scale information for vision, and the gyroscope and accelerometer equipped with IMU can deliver high-frequency angular velocity and acceleration data for state estimation. This has profound significance in improving the positioning accuracy of low-cost automatic robot systems. However, the measurement noise and random walk of IMU lead to gradually increasing drift errors in inertial measurements. Meanwhile, as the optimization progresses, the measurement bias of IMU will change, so that the position, velocity, and angle increments need to be recalculated. To avoid repeated integration computations, pre-integration provides an approximate correction for IMU measurements [141]. In VIO system, the low-frequency image keyframes are used as time nodes for state estimation. The pre-integration of IMU combines high-frequency inertial measurements between two keyframes into relative motion constraints in factor graph optimization in this study [142].

IMU-driven System Kinematic Model

The raw measurements of IMU gyroscope $\widetilde{\boldsymbol{\omega}}$ and accelerometer $\widetilde{\boldsymbol{a}}$ are modeled as

$$
\begin{aligned}
\widetilde{\boldsymbol{\omega}}(t) &= \boldsymbol{\omega}(t) + \boldsymbol{b}_g(t) + \boldsymbol{n}_g(t) \\
\widetilde{\boldsymbol{a}}(t) &= \boldsymbol{R}_w^b(\boldsymbol{a}(t) - \boldsymbol{g}^w) + \boldsymbol{b}_a(t) + \boldsymbol{n}_a(t)
\end{aligned}, \tag{4.3}
$$

where $\boldsymbol{\omega}$ and $\boldsymbol{a}$ are the acceleration and angular velocity of the body frame with respect to the inertial frame. $\boldsymbol{g}^w = [0,0,g]^T$ is the Gravity acceleration in the world frame. The $\boldsymbol{n}_g$ and $\boldsymbol{n}_a$ are measurement Gaussian white noise with zero-mean, as $\boldsymbol{n}_g \sim \mathcal{N}(0,\boldsymbol{\sigma}_g^2)$, $\boldsymbol{n}_a \sim \mathcal{N}(0,\boldsymbol{\sigma}_a^2)$ [1]. $\boldsymbol{b}_g(t)$ and $\boldsymbol{b}_a(t)$ are modeled as random-walk processes and their derivatives are zero-mean, white Gaussian noise, $\boldsymbol{n}_{b_g} \sim$

$\mathcal{N}(0, \boldsymbol{\sigma}_{b_g}^2)$, $\boldsymbol{n}_{b_a} \sim \mathcal{N}(0, \boldsymbol{\sigma}_{b_a}^2)$, we have

$$\dot{\boldsymbol{b}}_g(t) = \boldsymbol{n}_{b_g}, \quad \dot{\boldsymbol{b}}_a(t) = \boldsymbol{n}_{b_a}. \tag{4.4}$$

The kinematic model can be described as [112]

$$\begin{aligned}
\dot{\boldsymbol{p}}_b^w(t) &= \boldsymbol{v}_b^w(t) \\
\dot{\boldsymbol{v}}_b^w(t) &= \boldsymbol{R}_b^w(t)\boldsymbol{a}(t) \\
\dot{\boldsymbol{q}}_b^w(t) &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}(t))\boldsymbol{q}_b^w(t), \\
\dot{\boldsymbol{b}}_g(t) &= \boldsymbol{n}_{b_g}(t) \\
\dot{\boldsymbol{b}}_a(t) &= \boldsymbol{n}_{b_a}(t)
\end{aligned} \tag{4.5}$$

where,

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\rfloor_\times & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^\top & 0 \end{bmatrix}, \lfloor\boldsymbol{\omega}\rfloor_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{4.6}$$

IMU pre-intergration

If the IMU body states in frame k is known (position $\boldsymbol{p}_{b_k}^w$, velocity $\boldsymbol{v}_{b_k}^w$, and rotation $\boldsymbol{q}_{b_k}^w$), according to the (4.3) and (4.5), these states in frame $k+1$ can be calculated by the following continuous integral formula,

$$\begin{aligned}
\boldsymbol{p}_{b_{k+1}}^w &= \boldsymbol{p}_{b_k}^w + \boldsymbol{v}_{b_k}^w \Delta t + \iint_{t\in[t_k,t_{k+1}]} \boldsymbol{R}_b^w(t)(\tilde{\boldsymbol{a}}(t) - \boldsymbol{b}_a(t) - g^w)dt^2 \\
\boldsymbol{v}_{b_{k+1}}^w &= \boldsymbol{v}_{b_k}^w + \int_{t\in[t_k,t_{k+1}]} \boldsymbol{R}_b^w(t)(\tilde{\boldsymbol{a}}(t) - \boldsymbol{b}_a(t) - g^w)dt \qquad , \\
\boldsymbol{q}_{b_{k+1}}^w &= \boldsymbol{q}_{b_k}^w \otimes \int_{t\in[t_k,t_{k+1}]} \frac{1}{2}\boldsymbol{\Omega}(\tilde{\boldsymbol{\omega}}(t) - \boldsymbol{b}_g(t))\boldsymbol{q}_b^{b_k}(t)dt
\end{aligned} \tag{4.7}$$

Where Δt is the time different between two frames, $\boldsymbol{R}_b^w(t)$ means the rotation from body to world frame at time t. It can be seen from (4.7), when IMU body states at frame k changes because of optimization updates, the states at frame $k+1$ require reintegration. To avoid this computational consumption, (4.7) is decomposed and simplified to

$$\begin{aligned} \boldsymbol{p}_{b_{k+1}}^{w} &= \boldsymbol{p}_{b_k}^{w} + \boldsymbol{v}_{b_k}^{w}\Delta t - \frac{1}{2}\boldsymbol{g}^{w}\Delta t^2 + \boldsymbol{p}_{b_k}^{w}\alpha_{b_{k+1}}^{b_k} \\ \boldsymbol{v}_{b_{k+1}}^{w} &= \boldsymbol{v}_{b_k}^{w} - \boldsymbol{g}^{w}\Delta t + \boldsymbol{p}_{b_k}^{w}\beta_{b_{k+1}}^{b_k} \\ \boldsymbol{q}_{b_{k+1}}^{w} &= \boldsymbol{q}_{b_k}^{w} \otimes \gamma_{b_{k+1}}^{b_k} \end{aligned} , \tag{4.8}$$

among that,

$$\begin{aligned} \alpha_{b_{k+1}}^{b_k} &= \iint\limits_{t\in[t_k,t_{k+1}]} \boldsymbol{R}_{b}^{b_k}(t)(\tilde{\boldsymbol{a}}(t) - \boldsymbol{b}_a(t))dt^2 \\ \beta_{b_{k+1}}^{b_k} &= \int_{t\in[t_k,t_{k+1}]} \boldsymbol{R}_{b}^{b_k}(t)(\tilde{\boldsymbol{a}}(t) - \boldsymbol{b}_a(t))dt \\ \gamma_{b_{k+1}}^{b_k} &= \int_{t\in[t_k,t_{k+1}]} \frac{1}{2}\boldsymbol{\Omega}(\tilde{\boldsymbol{\omega}}(t) - \boldsymbol{b}_g(t))\gamma_t^{b_k}dt \end{aligned} \cdot \tag{4.9}$$

Where, $\alpha_{b_{k+1}}^{b_k}, \beta_{b_{k+1}}^{b_k}, \gamma_{b_{k+1}}^{b_k}$ are called as IMU pre-integration measurements and can be solved without knowing the IMU body states in frame k [34]. The three new measurements will be used for IMU constraints during state optimization. However, (4.9) include the state variables $\boldsymbol{b}_a$ and $\boldsymbol{b}_g$. To avoid repeat integration operation when $\boldsymbol{b}_a$ and $\boldsymbol{b}_g$ are changed during optimization, the solution of (4.9) is approximated by a first-order expansion [1],

$$\begin{aligned} \alpha_{b_{k+1}}^{b_k} &= \hat{\alpha}_{b_{k+1}}^{b_k} + \boldsymbol{J}_{b_a}^{\alpha}\delta\boldsymbol{b}_{a_k} + \boldsymbol{J}_{b_g}^{\alpha}\delta\boldsymbol{b}_{g_k} \\ \beta_{b_{k+1}}^{b_k} &= \hat{\beta}_{b_{k+1}}^{b_k} + \boldsymbol{J}_{b_a}^{\beta}\delta\boldsymbol{b}_{a_k} + \boldsymbol{J}_{b_g}^{\beta}\delta\boldsymbol{b}_{g_k} \\ \gamma_{b_{k+1}}^{b_k} &= \hat{\gamma}_{b_{k+1}}^{b_k} + \otimes\begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{J}_{b_g}^{\gamma}\delta\boldsymbol{b}_{g_k} \end{bmatrix} \end{aligned} , \tag{4.10}$$

where $\boldsymbol{J}_{b}^{a}$ is the Jacobian matrix of pre-integrated measurements with respect to bias at time k [34]. So far, the IMU pre-integration has been converted to a linear operation, which greatly reduces the complexity of the computation.

Combining the (4.8) and (4.10), the measurement residual of IMU preintegration between two consecutive frames b_k and b_{k+1} can be defined as

$$\begin{aligned} \boldsymbol{r}_b(\boldsymbol{z}_{b_{k+1}}^{b_k}, \mathcal{X}) &= \begin{bmatrix} \delta \alpha_{b_{k+1}}^{b_k} \\ \delta \beta_{b_{k+1}}^{b_k} \\ \delta \gamma_{b_{k+1}}^{b_k} \\ \delta b_a \\ \delta b_g \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{R}_w^{b_k}(\boldsymbol{p}_{b_{k+1}}^w - \boldsymbol{p}_{b_k}^w - \boldsymbol{v}_{b_k}^w \Delta t + \frac{1}{2}\boldsymbol{g}^w \Delta t^2) - \hat{\alpha}_{b_{k+1}}^{b_k} \\ \boldsymbol{R}_w^{b_k}(\boldsymbol{v}_{b_{k+1}}^w - \boldsymbol{v}_{b_k}^w + \boldsymbol{g}^w \Delta t) - \hat{\beta}_{b_{k+1}}^{b_k} \\ 2[(\boldsymbol{q}_{b_k}^w)^{-1} \otimes \boldsymbol{q}_{b_{k+1}}^w \otimes (\hat{\gamma}_{b_{k+1}}^{b_k})^{-1}] \\ b_{ab_{k+1}} - b_{ab_k} \\ b_{gb_{k+1}} - b_{gb_k} \end{bmatrix} \end{aligned} \tag{4.11}$$

4.2.2 Point Feature Measurements Model

This work retains the point feature as an optimization factor in the VIO system. The factor graph optimization is illustrated in Fig. 4.2. In terms of point features, the measurement residual is built by the re-projection error. This error means the pixel distance in an image between the projected feature p and the observed feature $\hat{p}$. Assuming the k-th point feature is co-measured in the i-th and j-th frame as $\hat{p}_k^{c_i}(\mu_k^{c_i}, \nu_k^{c_i})$ and $\hat{p}_k^{c_j}(\mu_k^{c_j}, \nu_k^{c_j})$, respectively. Its measurement residual model can be written as

$$\begin{aligned} \boldsymbol{r}_c(\boldsymbol{z}_c, \mathcal{X}) &= p_k^{c_j} - \hat{p}_k^{c_j} \\ p_k^{c_j} &= \pi\Big(\boldsymbol{R}_b^c\Big(\boldsymbol{R}_w^{b_j}\Big(\boldsymbol{P}_k^w - \boldsymbol{p}_{b_j}^w\Big) - \boldsymbol{p}_c^b\Big)\Big) \end{aligned}, \tag{4.12}$$

$$\boldsymbol{P}_k^w = \begin{bmatrix} \boldsymbol{X}_k^w \\ \boldsymbol{Y}_k^w \\ \boldsymbol{Z}_k^w \end{bmatrix} = \boldsymbol{R}_{b_i}^w\left(\boldsymbol{R}_c^b \frac{1}{\lambda_i} \pi^{-1}\left(\begin{bmatrix} \hat{\mu}_k^{c_i} \\ \hat{\nu}_k^{c_i} \end{bmatrix}\right) + \boldsymbol{p}_c^b\right) + \boldsymbol{p}_{b_i}^w, \tag{4.13}$$

where, $\boldsymbol{P}_k^w$ is the position of the k-th point feature in the world frame, and $\pi(\cdot)$ means the projection function and $\pi^{-1}(\cdot)$ is the back-projection.

4.2.3 Line Feature Measurements Model

Consistent with the representation of line feature in Chapter 3, this work still uses the two explicit endpoints $\boldsymbol{P}_s(x_s, y_s, z_s), \boldsymbol{P}_e(x_e, y_e, z_e)$ to represent the spatial line and performs subsequent operations based on these two endpoints.

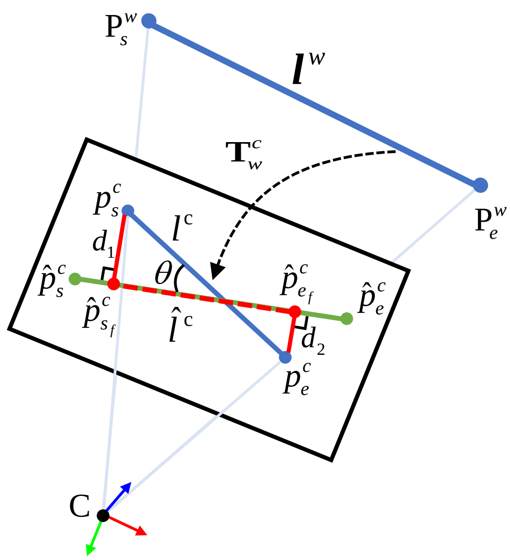


Figure 4.3: The line matching criteria and reprojection error.

Line Matching & Tracking

Before constructing optimization constraints using line segment features, it is necessary to detect and match lines in 2D images and 3D prior maps. First, the line segments in prior map are extracted by a segment-based 3D line detection method [122] and the lines in 2D images are detected by fast line detection (FLD) method [125]. For line matching, 3D lines in prior map are projected onto image plane based on the field of view (FoV) of camera first. Then these 3D lines falling into FoV will be matched with the line features detected in 2D images. Fig. 4.3 shows the matching criteria. A 3D line $\boldsymbol{l}^w$ is projected into an image plane by the initial pose $\boldsymbol{T}_w^c$, which is the line l^c containing two endpoints p_s^c, p_e^c. The line l^c is matched with a detected line $\hat{l}^c$ by the line distance: $D = d_1 + d_2$, the angle θ between the two lines, and the overlap shown by the red dash line $(\hat{p}_{sf}^c, \hat{p}_{ef}^c)$. As mentioned above, a sliding window structure is used to improve the robustness and accuracy of state estimation. The visual features within the field of view (FoV) of the sliding window are tracked and filtered by the Lucas-Kanade (LK) optical flow method [143] to improve feature reliability.

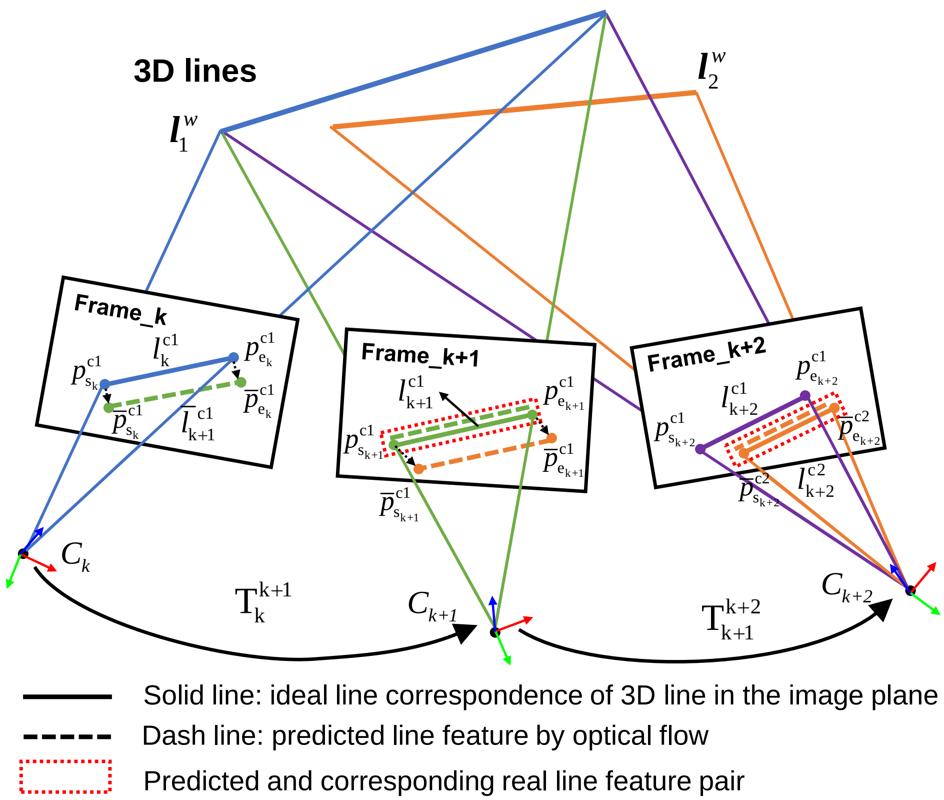


Figure 4.4: The line tracking strategy.

For the line feature tracking strategy in this work, we still focus on the two endpoints of lines. By drawing on the point tracking method, the line features introduced in this study adopts a comparable optical flow-based tracking approach. As shown in Fig. 4.4, when a detected 2D line l_k^{c1} on frame k is matched with a 3D line $\boldsymbol{l}_1^w$, the two endpoints on l_k^{c1} are tracked by LK optical flow method to predict its position $\bar{l}_{k+1}^{c1}$ on the next keyframe $k+1$. The detected line l_{k+1}^{c1}, which is closest to the predicted line $\bar{l}_{k+1}^{c1}$ on the next frame $k+1$, is taken as the corresponding tracked line feature of l_k^{c1}. Next, if the line l_{k+1}^{c1} has the same matching 3D line $\boldsymbol{l}_1^w$ as l_k^{c1}, then this set of matching pairs is considered reliable. Otherwise, this set of features is removed from observations. Fig. 4.4 demonstrates the two cases: frame k to $k+1$ shows the case of successful tracking, and frame $k+1$ to $k+2$ is the failed case.

Line Feature-based Optimization Model

As shown in Fig. 4.3, a 3D line $\boldsymbol{l}^w$ with two endpoints $\boldsymbol{P}_s^w, \boldsymbol{P}_e^w$ is projected into l^c (called as the predicted line) on the image plane, its endpoints are p_s^c, p_e^c. This projected line l^c is matched with the detected line $\hat{l}^c$ on this image and with endpoints p_s^c, p_e^c have two corresponding foot points $\hat{p}_{s_f}^c, \hat{p}_{e_f}^c$ on $\hat{l}^c$. For the line pair, the error function can be built by the distance between the endpoint and its foot point (taking the point pair (p_s^c, $\hat{p}_{s_f}^c$) as example),

$$\begin{aligned}
\boldsymbol{r}_l(\boldsymbol{z}_l, \mathcal{X}) &= p_s^c - \hat{p}_{s_f}^c \\
p_l^c &= \pi\Big(\boldsymbol{R}_b^c\Big(\boldsymbol{R}_w^b(\boldsymbol{P}_s^w - \boldsymbol{p}_b^w) - \boldsymbol{p}_c^b\Big)\Big). \\
\hat{p}_{s_f}^c &= f(p_s^c)
\end{aligned} \tag{4.14}$$

where, $f(\cdot)$ is to solve the foot point $\hat{p}_{s_f}^c$ on the matched line $\hat{l}^c$, which is related to the states $\boldsymbol{R}_b^c$, $\boldsymbol{p}_b^c$, $\boldsymbol{R}_w^b$, $\boldsymbol{p}_w^b$. Firstly, the general equation of line $\hat{l}^c$ on the image is

$$\boldsymbol{A}\hat{\mu} + \boldsymbol{B}\hat{\nu} + \boldsymbol{C} = 0. \tag{4.15}$$

$(\hat{\mu}, \hat{\nu})$ is the pixel coordinate on the image plane. If the pixel location of the endpoint p_s^c on image is (μ, ν), its foot point $\hat{p}_{s_f}^c$ on $\hat{l}^c$ can be computed by [124]

$$\hat{\mu}_f = \frac{\boldsymbol{B}^2\mu - \boldsymbol{A}\boldsymbol{B}\nu - \boldsymbol{A}\boldsymbol{C}}{\boldsymbol{A}^2 + \boldsymbol{B}^2}, \hat{\nu}_f = \frac{\boldsymbol{A}^2\nu - \boldsymbol{A}\boldsymbol{B}\mu - \boldsymbol{B}\boldsymbol{C}}{\boldsymbol{A}^2 + \boldsymbol{B}^2}. \tag{4.16}$$

So, the function $f(\cdot)$ can be written by

$$\begin{aligned} \hat{p}_{s_f}^c &= \begin{bmatrix} \hat{\mu}_f \\ \hat{\nu}_f \\ 1 \end{bmatrix} = f(p_s^c) = \boldsymbol{F} \begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix} \\ \boldsymbol{F} &= \frac{1}{\boldsymbol{A}^2 + \boldsymbol{B}^2} \begin{bmatrix} \boldsymbol{B}^2 & -\boldsymbol{A}\boldsymbol{B} & -\boldsymbol{A}\boldsymbol{C} \\ -\boldsymbol{A}\boldsymbol{B} & \boldsymbol{A}^2 & -\boldsymbol{B}\boldsymbol{C} \\ 0 & 0 & \boldsymbol{A}^2 + \boldsymbol{B}^2 \end{bmatrix} \end{aligned} \tag{4.17}$$

Rationality Analysis of the New Cost Function

By combining formulas (4.14) and (4.17), the line feature-based residual can be expressed as

$$\begin{aligned} \boldsymbol{r}_l(\boldsymbol{z}_l, \mathcal{X}) &= p_s^c - f(p_s^c) = (\boldsymbol{I} - \boldsymbol{F}) p_s^c \\ &= \frac{1}{\boldsymbol{A}^2 + \boldsymbol{B}^2} \begin{bmatrix} \boldsymbol{A}^2 & \boldsymbol{A}\boldsymbol{B} & \boldsymbol{A}\boldsymbol{C} \\ \boldsymbol{A}\boldsymbol{B} & \boldsymbol{B}^2 & \boldsymbol{B}\boldsymbol{C} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix}. \end{aligned} \tag{4.18}$$

In an ideal situation, when the model converges to the global optimal, the residual $\boldsymbol{r}_l = 0$. Based on the (4.18), we have

$$\boldsymbol{A}\mu + \boldsymbol{B}\nu + \boldsymbol{C} = 0, \tag{4.19}$$

which means that the two endpoints on the predicted line l^c satisfy the line equation of the matched line $\hat{l}^c$. This implies that the two lines coincide exactly, proving that our proposed optimization model has optimal convergence under ideal conditions. This proof indicates the rationality and feasibility of our proposed model.

4.3 Observability Analysis

During the state estimation process, observability analysis is regularly used to describe whether a system is observable over different state spaces [112, 144, 145].

Unobservability may lead to the non-convergence of the direction of the state to be estimated. Therefore, the properties of the system state estimation can be profoundly understood through observability analysis.

First, the proposed system can be modeled as a nonlinear affine expression as follows:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}_0(\boldsymbol{x}) + \sum_{i=1}^{n} \boldsymbol{f}_i(\boldsymbol{x}) u_i \\ \boldsymbol{z}_p = \boldsymbol{h}_1(\boldsymbol{x}) \\ \boldsymbol{z}_l = \boldsymbol{h}_2(\boldsymbol{x}) \end{cases} \tag{4.20}$$

where, $\boldsymbol{x}$ is the states, $\boldsymbol{u} = [\widetilde{\boldsymbol{\omega}}, \widetilde{\boldsymbol{a}}]$ is the IMU inputs. $\dot{\boldsymbol{x}}$ is from IMU system kinematic model in (4.5), where the gyroscope and accelerometer biases $\boldsymbol{b}_g$ and $\boldsymbol{b}_a$ are assumed as piecewise constant between consecutive keyframes for discrete-time state estimation. which is equivalent to adopting the model $\dot{\boldsymbol{b}}_g = 0, \dot{\boldsymbol{b}}_a = 0$, the derivative of the extrinsic parameters $\dot{\boldsymbol{q}}_c^b = \mathbf{0}_{3\times1}, \dot{\boldsymbol{p}}_c^b = \mathbf{0}_{3\times1}$, and the derivative of the point feature 3D position $\dot{\boldsymbol{P}}_k^w = \mathbf{0}_{3\times1}$. $\boldsymbol{z}_p$ is the point feature-based measurement model, which can be seen in our work. $\boldsymbol{z}_l$ is the line feature-based measurement model.

Therefore, the complete model in (4.20) can be expressed by

$$
\begin{aligned}
&\dot{\boldsymbol{x}} = \boldsymbol{f}_0(\boldsymbol{x}) + \boldsymbol{f}_1(\boldsymbol{x})\widetilde{\boldsymbol{\omega}} + \boldsymbol{f}_2(\boldsymbol{x})\widetilde{\boldsymbol{a}} \\
\Longrightarrow &\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{s}}_1 \\ \dot{\boldsymbol{b}}_g \\ \dot{\boldsymbol{b}}_a \\ \dot{\boldsymbol{p}}_c^b \\ \dot{\boldsymbol{s}}_2 \\ \dot{\boldsymbol{P}}_k^w \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{g}^w - \boldsymbol{R}_b^w \boldsymbol{b}_a \\ -\boldsymbol{D}_1 \boldsymbol{b}_g \\ 0_{3\times1} \\ 0_{3\times1} \\ 0_{3\times1} \\ 0_{3\times1} \\ 0_{3\times1} \end{bmatrix} + \begin{bmatrix} 0_3 \\ 0_3 \\ \boldsymbol{D}_1 \\ 0_3 \\ 0_3 \\ 0_3 \\ 0_3 \\ 0_3 \end{bmatrix} \widetilde{\boldsymbol{\omega}} + \begin{bmatrix} 0_3 \\ \boldsymbol{R}_b^w \\ 0_3 \\ 0_3 \\ 0_3 \\ 0_3 \\ 0_3 \\ 0_3 \end{bmatrix} \widetilde{\boldsymbol{a}} \\
&\boldsymbol{z}_p = \boldsymbol{\pi}\left(\boldsymbol{R}_b^c\left(\boldsymbol{R}_w^b(\boldsymbol{P}_k^w - \boldsymbol{p}_b^w) - \boldsymbol{p}_c^b\right)\right) \\
&\boldsymbol{z}_l = \boldsymbol{\pi}\left(\boldsymbol{R}_b^c\left(\boldsymbol{R}_w^b(\boldsymbol{P}_l^w - \boldsymbol{p}_b^w) - \boldsymbol{p}_c^b\right)\right)
\end{aligned}
\tag{4.21}
$$

where, $\boldsymbol{P}_l^w$ means the endpoint of the line segment, $\boldsymbol{s}_1$ corresponds to the $\boldsymbol{q}_b^w$, $\boldsymbol{s}_2$ means the $\boldsymbol{q}_c^b$. $\boldsymbol{s}$ is the Cayley-Gibbs-Rodrigues (CGR) parameterization representation of the orientation to facilitate the derivation calculation, and $\boldsymbol{s} = \boldsymbol{n}\tan\frac{\theta}{2}$ with $\boldsymbol{n}$ is the rotation axis and θ is the axial angle [146][147]. The partial derivative of $\boldsymbol{q}_b^w(t)$ respect to time is

$$
\begin{aligned}
\dot{\boldsymbol{s}}_b^w(t) &= \boldsymbol{D}(\widetilde{\boldsymbol{\omega}} - \boldsymbol{b}_g) \\
\boldsymbol{D} \triangleq \frac{\partial \boldsymbol{s}}{\partial \boldsymbol{\theta}} &= \frac{1}{2}(\boldsymbol{I} + \lfloor \boldsymbol{s} \rfloor_\times + \boldsymbol{s}\boldsymbol{s}^\top)
\end{aligned}.
\tag{4.22}
$$

Typically, the observability analysis of a nonlinear system can be performed by taking the Lie derivatives of the measurement function $\boldsymbol{h}(\boldsymbol{x})$ to construct the observability matrix $\mathcal{O}$ [112, 144] , i.e.,

$$
\mathcal{O} = \begin{bmatrix} \nabla_{\boldsymbol{x}} \mathcal{L}^0 \boldsymbol{h}_k \\ \nabla_{\boldsymbol{x}} \mathcal{L}^1_{\boldsymbol{f}_i} \boldsymbol{h}_k \\ \nabla_{\boldsymbol{x}} \mathcal{L}^2_{\boldsymbol{f}_i \boldsymbol{f}_j} \boldsymbol{h}_k \\ \vdots \end{bmatrix}, \begin{cases} k = 1,2, \\ i,j = 1,2,3, \end{cases}
\tag{4.23}
$$

$$
\begin{aligned}
&\mathcal{L}^0 \boldsymbol{h}_k = \boldsymbol{h}_k(x), \\
&\mathcal{L}^{i+1}_{\boldsymbol{f}_j} \boldsymbol{h}_k = \nabla_{\boldsymbol{x}} \mathcal{L}^i \boldsymbol{h}_k \cdot \boldsymbol{f}_j, \\
&\nabla_{\boldsymbol{x}} \mathcal{L}^i \boldsymbol{h}_k = \begin{bmatrix} \frac{\partial \mathcal{L}^i \boldsymbol{h}_k}{\partial x_1} & \frac{\partial \mathcal{L}^i \boldsymbol{h}_k}{\partial x_2} & \cdots & \frac{\partial \mathcal{L}^i \boldsymbol{h}_k}{\partial x_m} \end{bmatrix}.
\end{aligned}
\tag{4.24}
$$

The right null space of $\mathcal{O}$ identifies the unobservable directions of the system. However, the full expression of $\mathcal{O}$ can be cumbersome. To simplify the computation, we adopt the approach from [112] by introducing a set of basis elements of the state variables, denoted as $\boldsymbol{\beta}(\boldsymbol{x}) = [\beta_1(\boldsymbol{x})^\top, \beta_2(\boldsymbol{x})^\top, \cdots, \beta_t(\boldsymbol{x})^\top]$. This allows the observability matrix to be factorized as

$$
\mathcal{O} = \begin{bmatrix} \mathcal{L}^0 \boldsymbol{h}_k \\ \mathcal{L}^1_{\boldsymbol{f}_i} \boldsymbol{h}_k \\ \mathcal{L}^2_{\boldsymbol{f}_i \boldsymbol{f}_j} \boldsymbol{h}_k \\ \vdots \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{L}^0 \boldsymbol{h}_k}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathcal{L}^1_{\boldsymbol{f}_i} \boldsymbol{h}_k}{\partial \boldsymbol{\beta}} \\ \frac{\partial \mathcal{L}^2_{\boldsymbol{f}_i \boldsymbol{f}_j} \boldsymbol{h}_k}{\partial \boldsymbol{\beta}} \\ \vdots \end{bmatrix} \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{x}} = \Xi \cdot \boldsymbol{B}. \tag{4.25}
$$

Here, Ξ is a full-rank matrix, which ensures that $\boldsymbol{B}$ shares the same null space as $\mathcal{O}$, i.e.,

$$
null(\mathcal{O}) = null(\boldsymbol{B}). \tag{4.26}
$$

The matrix $\boldsymbol{B}$ can be expressed as

$$
\begin{aligned}
\boldsymbol{B} &= \begin{bmatrix} \frac{\partial \beta_1}{\partial \boldsymbol{x}} & \frac{\partial \beta_2}{\partial \boldsymbol{x}} & \cdots & \frac{\partial \beta_8}{\partial \boldsymbol{x}} \end{bmatrix}^\top, \\
\frac{\partial \beta_i}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{\partial \beta_i}{\partial \boldsymbol{x}_1} & \frac{\partial \beta_i}{\partial \boldsymbol{x}_2} & \cdots & \frac{\partial \beta_i}{\partial \boldsymbol{x}_m} \end{bmatrix},
\end{aligned}
\tag{4.27}
$$

where m denotes the number of states. Therefore, the unobservable directions of the system can be efficiently derived by computing the right null space of $\boldsymbol{B}$.

Specifically, the derivation can be formalized by the following theorem [112].

Theorem 1: Assume a nonlinear transformation $\boldsymbol{\beta}(\boldsymbol{x}) = [\beta_1(\boldsymbol{x})^\top, \ldots, \beta_t(\boldsymbol{x})^\top]$ exists, where each β_i is a function of the state $\boldsymbol{x}$ and t is the number of basis elements, satisfying

$$
\begin{aligned}
&(C1) \quad \beta_1(\boldsymbol{x}) = \boldsymbol{z}(\boldsymbol{x}), \\
&(C2) \quad \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{x}} \cdot \boldsymbol{f}_i = 0, \quad i = 0, \ldots, l, \\
&(C3) \quad \begin{cases} \dot{\boldsymbol{\beta}} = \boldsymbol{g}_0(\boldsymbol{\beta}) + \sum_{i=1}^{l} \boldsymbol{g}_i(\boldsymbol{\beta})\boldsymbol{u}_i, \\ \boldsymbol{z} = \boldsymbol{h} = \beta_1(\boldsymbol{x}). \end{cases}
\end{aligned}
\tag{4.28}
$$

Then, the observability matrix of the original system can be factorized as $\mathcal{O} = \Xi \cdot \boldsymbol{B}$ and $null(\mathcal{O}) = null(\boldsymbol{B})$.

Applying Theorem 1 to our system, we first determine the basis functions according to (C1) and (C2). From (C1), the first basis function is

$$
\beta_1 = \boldsymbol{z}_p = \frac{1}{p_z} \begin{bmatrix} p_x \\ p_y \end{bmatrix}, \tag{4.29}
$$

where $\boldsymbol{z}_p$ denotes the normalized pixel coordinates of a feature point. Next, condition (C2) is used to determine the remaining basis functions.

We start by computing the span of $\partial \beta_1$ with respect to $\boldsymbol{x}$, i.e.

$$
\begin{aligned}
\frac{\partial \beta_1}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{\partial \beta_1}{\partial \boldsymbol{p}} & \frac{\partial \beta_1}{\partial \boldsymbol{v}} & \frac{\partial \beta_1}{\partial \theta_1}\frac{\partial \theta_1}{\partial \boldsymbol{s}_1} & \frac{\partial \beta_1}{\partial \boldsymbol{b}_g} & \frac{\partial \beta_1}{\partial \boldsymbol{b}_a} & \frac{\partial \beta_1}{\partial \boldsymbol{p}_c^b} & \frac{\partial \beta_1}{\partial \theta_2}\frac{\partial \theta_2}{\partial \boldsymbol{s}_2} & \frac{\partial \beta_1}{\partial \boldsymbol{p}_k^w} \end{bmatrix} \\
&= \frac{\partial \boldsymbol{z}_p}{\partial \boldsymbol{P}_k^c} \cdot \frac{\partial \boldsymbol{P}_k^c}{\partial \boldsymbol{x}} \\
&= \begin{bmatrix} \frac{1}{p_z} & 0 & -\frac{p_x}{p_z^2} \\ 0 & \frac{1}{p_z} & -\frac{p_y}{p_z^2} \end{bmatrix} \begin{bmatrix} \boldsymbol{C}_2\boldsymbol{C}_1 & \boldsymbol{0}_3 & \boldsymbol{C}_2\lfloor \boldsymbol{P}_k^b \rfloor_\times \frac{\partial \theta_1}{\partial \boldsymbol{s}_1} & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{C}_2 & \lfloor \boldsymbol{P}_k^c \rfloor_\times \frac{\partial \theta_2}{\partial \boldsymbol{s}_2} & -\boldsymbol{C}_2\boldsymbol{C}_1 \end{bmatrix}
\end{aligned}
\tag{4.30}
$$

Where, $\boldsymbol{C}_1 = \boldsymbol{R}_w^b$, $\boldsymbol{C}_2 = \boldsymbol{R}_b^c$, $\frac{\partial \boldsymbol{z}_p}{\partial \boldsymbol{P}_k^c}$ comes from the normalized camera intrinsic projection model. Regarding as $\frac{\partial \beta}{\partial \theta}$, it can be solved by a small-perturbation model about the derivation of $\boldsymbol{Rp}$ respect to the rotation $\boldsymbol{R}$ [73], it is

$$
\frac{\partial \boldsymbol{Rp}}{\partial \varphi} = -\lfloor \boldsymbol{Rp} \rfloor_\times, \tag{4.31}
$$

among that, $\boldsymbol{p}$ is a space point, $\boldsymbol{R}$ is a rotation matrix, $\varphi = \Delta\boldsymbol{R}$ is the small perturbation and equal to $\boldsymbol{\theta}$, $\lfloor\cdot\rfloor_\times$ means the notation of the skew symmetric matrix, same with (4.6).

Then, the $\frac{\partial\beta_1}{\partial\boldsymbol{x}}\boldsymbol{f}_0$ is

$$\begin{aligned}\frac{\partial\beta_1}{\partial\boldsymbol{x}}\boldsymbol{f}_0 &= \begin{bmatrix}\frac{1}{p_z} & 0 & -\frac{p_x}{p_z^2}\\ 0 & \frac{1}{p_z} & -\frac{p_y}{p_z^2}\end{bmatrix}(\boldsymbol{C}_2\boldsymbol{C}_1\boldsymbol{v} - \boldsymbol{C}_2\lfloor\boldsymbol{P}_k^b\rfloor_\times\boldsymbol{b}_g)\\ &= \begin{bmatrix}\boldsymbol{I}_2 & -\beta_1\end{bmatrix}\left(\frac{1}{p_z}\boldsymbol{C}_2\boldsymbol{C}_1\boldsymbol{v} - \left\lfloor\begin{bmatrix}\beta_1\\1\end{bmatrix} + \frac{1}{p_z}\boldsymbol{C}_2\boldsymbol{p}_c^b\right\rfloor_\times\boldsymbol{C}_2\boldsymbol{b}_g\right),\end{aligned} \tag{4.32}$$

since

$$\boldsymbol{C}_2\lfloor\boldsymbol{P}_k^b\rfloor_\times = \boldsymbol{C}_2\lfloor\boldsymbol{R}_c^b\boldsymbol{P}_k^c + \boldsymbol{P}_c^b\rfloor_\times = \lfloor\boldsymbol{C}_2\boldsymbol{C}_2^\top\boldsymbol{P}_k^c + \boldsymbol{C}_2\boldsymbol{P}_c^b\rfloor_\times\boldsymbol{C}_2 = \lfloor\boldsymbol{P}_k^c + \boldsymbol{C}_2\boldsymbol{P}_c^b\rfloor_\times\boldsymbol{C}_2 \tag{4.33}$$

based on the skew symmetric matrix operation rule:

$$\lfloor\boldsymbol{U}\boldsymbol{a}\rfloor_\times = \boldsymbol{U}\lfloor\boldsymbol{a}\rfloor_\times\boldsymbol{U}^\top \Longrightarrow \lfloor\boldsymbol{U}\boldsymbol{a}\rfloor_\times\boldsymbol{U} = \boldsymbol{U}\lfloor\boldsymbol{a}\rfloor_\times, \tag{4.34}$$

with $\boldsymbol{U}$ is a rotation matrix, $\boldsymbol{a}$ is a vector.

It is obvious that $\frac{\partial\beta_1}{\partial\boldsymbol{x}}\boldsymbol{f}_0$ cannot be expressed by the basis function β_1, so we add other basis functions as:

$$\begin{aligned}\beta_2 &\triangleq \frac{1}{p_z}\\ \beta_3 &\triangleq \boldsymbol{C}_1\boldsymbol{v}\\ \beta_4 &\triangleq \boldsymbol{C}_2\\ \beta_5 &\triangleq \boldsymbol{P}_c^b\\ \beta_6 &\triangleq \boldsymbol{b}_g\end{aligned}, \tag{4.35}$$

then, we have

$$\frac{\partial\beta_1}{\partial\boldsymbol{x}}\boldsymbol{f}_0 = \begin{bmatrix}\boldsymbol{I}_2 & -\beta_1\end{bmatrix}(\beta_2\beta_4\beta_3 - \left\lfloor\begin{bmatrix}\beta_1\\1\end{bmatrix} + \beta_2\beta_4\beta_5\right\rfloor_\times\beta_4\beta_6) \tag{4.36}$$

The $\frac{\partial \beta_1}{\partial \boldsymbol{x}} \boldsymbol{f}_1$ is

$$\frac{\partial \beta_1}{\partial \boldsymbol{x}} \boldsymbol{f}_1 = \begin{bmatrix} \frac{1}{p_z} & 0 & -\frac{p_x}{p_z^2} \\ 0 & \frac{1}{p_z} & -\frac{p_y}{p_z^2} \end{bmatrix} \boldsymbol{C}_2 \lfloor \boldsymbol{P}_k^b \rfloor_\times = \begin{bmatrix} \boldsymbol{I}_2 & -\beta_1 \end{bmatrix} \beta_4 \left\lfloor \begin{bmatrix} \beta_1 \\ 1 \end{bmatrix} + \beta_2 \beta_4 \beta_5 \right\rfloor_\times \beta_4 \tag{4.37}$$

We do not need to add any other functions.

For $\frac{\partial \beta_1}{\partial \boldsymbol{x}} \boldsymbol{f}_2$, we have

$$\frac{\partial \beta_1}{\partial \boldsymbol{x}} \boldsymbol{f}_2 = \mathbf{0}. \tag{4.38}$$

So far, we have completed the calculation of the first basis function β_1. Next, we will apply this calculation process to other basis functions.

For basis function $\beta_2 = \frac{1}{p_z}$, the span space is

$$\begin{aligned} \frac{\partial \beta_2}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{\partial \beta_2}{\partial \boldsymbol{p}} & \frac{\partial \beta_2}{\partial \boldsymbol{v}} & \frac{\partial \beta_2}{\partial \boldsymbol{\theta}_1} \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \frac{\partial \beta_2}{\partial \boldsymbol{b}_g} & \frac{\partial \beta_2}{\partial \boldsymbol{b}_a} & \frac{\partial \beta_2}{\partial \boldsymbol{p}_c^b} & \frac{\partial \beta_2}{\partial \boldsymbol{\theta}_2} \frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \frac{\partial \beta_2}{\partial \boldsymbol{p}_k^w} \end{bmatrix} \\ &= \frac{\partial \beta_2}{\partial \boldsymbol{P}_k^c} \cdot \frac{\partial \boldsymbol{P}_k^c}{\partial \boldsymbol{x}} \\ &= -\frac{1}{p_z^2} \boldsymbol{e}_3^\top \begin{bmatrix} \boldsymbol{C}_2 \boldsymbol{C}_1 & \mathbf{0}_3 & \boldsymbol{C}_2 \lfloor \boldsymbol{P}_k^b \rfloor_\times \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{C}_2 & \lfloor \boldsymbol{P}_k^c \rfloor_\times \frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & -\boldsymbol{C}_2 \boldsymbol{C}_1 \end{bmatrix} \end{aligned}, \tag{4.39}$$

where $\boldsymbol{e}_3$ is the third column of a 3×3 identity matrix. $[\boldsymbol{e}_1\ \boldsymbol{e}_2\ \boldsymbol{e}_3] = \boldsymbol{I}_3$.

Similar to (4.57), (4.37), (4.38), we have

$$\begin{aligned} \frac{\partial \beta_2}{\partial \boldsymbol{x}} \boldsymbol{f}_0 &= -\beta_2 \boldsymbol{e}_3^\top (\beta_2 \beta_4 \beta_3 - \left\lfloor \begin{bmatrix} \beta_1 \\ 1 \end{bmatrix} + \beta_2 \beta_4 \beta_5 \right\rfloor_\times \beta_4 \beta_6) \\ \frac{\partial \beta_2}{\partial \boldsymbol{x}} \boldsymbol{f}_1 &= -\beta_2 \boldsymbol{e}_3^\top \beta_4 \left\lfloor \begin{bmatrix} \beta_1 \\ 1 \end{bmatrix} + \beta_2 \beta_4 \beta_5 \right\rfloor_\times \beta_4 \\ \frac{\partial \beta_2}{\partial \boldsymbol{x}} \boldsymbol{f}_2 &= \mathbf{0}. \end{aligned} \tag{4.40}$$

For basis function $\beta_3 = \boldsymbol{C}_1 \boldsymbol{v}$, we have

$$\begin{aligned} \frac{\partial \beta_3}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{\partial \beta_3}{\partial \boldsymbol{p}} & \frac{\partial \beta_3}{\partial \boldsymbol{v}} & \frac{\partial \beta_3}{\partial \boldsymbol{\theta}_1} \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \frac{\partial \beta_3}{\partial \boldsymbol{b}_g} & \frac{\partial \beta_3}{\partial \boldsymbol{b}_a} & \frac{\partial \beta_3}{\partial \boldsymbol{p}_c^b} & \frac{\partial \beta_3}{\partial \boldsymbol{\theta}_2} \frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \frac{\partial \beta_3}{\partial \boldsymbol{p}_k^w} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0}_3 & \boldsymbol{C}_1 & \lfloor \boldsymbol{C}_1 \boldsymbol{v} \rfloor_\times \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \end{aligned}, \tag{4.41}$$

then,

$$\frac{\partial \beta_3}{\partial \boldsymbol{x}} \boldsymbol{f}_0 = \boldsymbol{C}_1 \boldsymbol{g}^w - \boldsymbol{b}_a - \lfloor \boldsymbol{C}_1 \boldsymbol{v} \rfloor_\times \boldsymbol{b}_g, \tag{4.42}$$

here, we need to define two new basis functions as

$$\begin{aligned} \beta_7 &\triangleq \boldsymbol{C}_1 \boldsymbol{g}^w \\ \beta_8 &\triangleq \boldsymbol{b}_a \end{aligned}, \tag{4.43}$$

so, we have

$$\begin{aligned} \frac{\partial \beta_3}{\partial \boldsymbol{x}} \boldsymbol{f}_0 &= \beta_7 - \beta_8 - \lfloor \beta_3 \rfloor_\times \beta_6 \\ \frac{\partial \beta_3}{\partial \boldsymbol{x}} \boldsymbol{f}_1 &= \lfloor \boldsymbol{C}_1 \boldsymbol{v} \rfloor_\times = \lfloor \beta_3 \rfloor_\times \\ \frac{\partial \beta_3}{\partial \boldsymbol{x}} \boldsymbol{f}_2 &= \boldsymbol{I}_3 \end{aligned} \tag{4.44}$$

For basis function $\beta_4 = \boldsymbol{C}_2$, we have

$$\begin{aligned} \frac{\partial \beta_4}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{\partial \beta_4}{\partial \boldsymbol{p}} & \frac{\partial \beta_4}{\partial \boldsymbol{v}} & \frac{\partial \beta_4}{\partial \boldsymbol{\theta}_1}\frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \frac{\partial \beta_4}{\partial \boldsymbol{b}_g} & \frac{\partial \beta_4}{\partial \boldsymbol{b}_a} & \frac{\partial \beta_4}{\partial \boldsymbol{p}_c^b} & \frac{\partial \beta_4}{\partial \boldsymbol{\theta}_2}\frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \frac{\partial \beta_4}{\partial \boldsymbol{p}_k^w} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 \end{bmatrix} \end{aligned}, \tag{4.45}$$

then,

$$\begin{aligned} \frac{\partial \beta_4}{\partial \boldsymbol{x}} \boldsymbol{f}_0 &= \mathbf{0} \\ \frac{\partial \beta_4}{\partial \boldsymbol{x}} \boldsymbol{f}_1 &= \mathbf{0} \\ \frac{\partial \beta_4}{\partial \boldsymbol{x}} \boldsymbol{f}_2 &= \mathbf{0} \end{aligned} \tag{4.46}$$

For basis function $\beta_5 = \boldsymbol{P}_c^b$, we have

$$\begin{aligned} \frac{\partial \beta_5}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{\partial \beta_5}{\partial \boldsymbol{p}} & \frac{\partial \beta_5}{\partial \boldsymbol{v}} & \frac{\partial \beta_5}{\partial \boldsymbol{\theta}_1}\frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \frac{\partial \beta_5}{\partial \boldsymbol{b}_g} & \frac{\partial \beta_5}{\partial \boldsymbol{b}_a} & \frac{\partial \beta_5}{\partial \boldsymbol{p}_c^b} & \frac{\partial \beta_5}{\partial \boldsymbol{\theta}_2}\frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \frac{\partial \beta_2}{\partial \boldsymbol{p}_k^w} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \end{aligned}, \tag{4.47}$$

with,

$$\begin{aligned}\frac{\partial \beta_5}{\partial \boldsymbol{x}}\boldsymbol{f}_0 &= \mathbf{0}\\ \frac{\partial \beta_5}{\partial \boldsymbol{x}}\boldsymbol{f}_1 &= \mathbf{0}\cdot\\ \frac{\partial \beta_5}{\partial \boldsymbol{x}}\boldsymbol{f}_2 &= \mathbf{0}\end{aligned} \tag{4.48}$$

For basis function $\beta_6 = \boldsymbol{b}_g$, we have

$$\begin{aligned}\frac{\partial \beta_6}{\partial \boldsymbol{x}} &= \begin{bmatrix}\frac{\partial \beta_6}{\partial \boldsymbol{p}} & \frac{\partial \beta_6}{\partial \boldsymbol{v}} & \frac{\partial \beta_6}{\partial \boldsymbol{\theta}_1}\frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \frac{\partial \beta_6}{\partial \boldsymbol{b}_g} & \frac{\partial \beta_6}{\partial \boldsymbol{b}_a} & \frac{\partial \beta_6}{\partial \boldsymbol{p}_c^b} & \frac{\partial \beta_6}{\partial \boldsymbol{\theta}_2}\frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \frac{\partial \beta_6}{\partial \boldsymbol{p}_k^w}\end{bmatrix},\\ &= \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix}\end{aligned} \tag{4.49}$$

with,

$$\begin{aligned}\frac{\partial \beta_6}{\partial \boldsymbol{x}}\boldsymbol{f}_0 &= \mathbf{0}\\ \frac{\partial \beta_6}{\partial \boldsymbol{x}}\boldsymbol{f}_1 &= \mathbf{0}\cdot\\ \frac{\partial \beta_6}{\partial \boldsymbol{x}}\boldsymbol{f}_2 &= \mathbf{0}\end{aligned} \tag{4.50}$$

For basis function $\beta_7 = \boldsymbol{C}_1\boldsymbol{g}^w$, we have

$$\begin{aligned}\frac{\partial \beta_7}{\partial \boldsymbol{x}} &= \begin{bmatrix}\frac{\partial \beta_7}{\partial \boldsymbol{p}} & \frac{\partial \beta_7}{\partial \boldsymbol{v}} & \frac{\partial \beta_7}{\partial \boldsymbol{\theta}_1}\frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \frac{\partial \beta_7}{\partial \boldsymbol{b}_g} & \frac{\partial \beta_7}{\partial \boldsymbol{b}_a} & \frac{\partial \beta_7}{\partial \boldsymbol{p}_c^b} & \frac{\partial \beta_7}{\partial \boldsymbol{\theta}_2}\frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \frac{\partial \beta_7}{\partial \boldsymbol{p}_k^w}\end{bmatrix},\\ &= \begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3 & \lfloor \boldsymbol{C}_1\boldsymbol{g}^w\rfloor_\times \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix}\end{aligned} \tag{4.51}$$

with,

$$\begin{aligned}\frac{\partial \beta_6}{\partial \boldsymbol{x}}\boldsymbol{f}_0 &= -\lfloor \boldsymbol{C}_1\boldsymbol{g}^w\rfloor_\times \boldsymbol{b}_g = \lfloor \beta_7\rfloor_\times \beta_6\\ \frac{\partial \beta_6}{\partial \boldsymbol{x}}\boldsymbol{f}_1 &= \lfloor \boldsymbol{C}_1\boldsymbol{g}^w\rfloor_\times = \lfloor \beta_7\rfloor_\times\\ \frac{\partial \beta_6}{\partial \boldsymbol{x}}\boldsymbol{f}_2 &= \mathbf{0}\end{aligned} \tag{4.52}$$

For basis function $\beta_8 = \boldsymbol{b}_a$, we have

$$\begin{aligned}\frac{\partial \beta_8}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{\partial \beta_8}{\partial \boldsymbol{p}} & \frac{\partial \beta_8}{\partial \boldsymbol{v}} & \frac{\partial \beta_8}{\partial \boldsymbol{\theta}_1}\frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \frac{\partial \beta_8}{\partial \boldsymbol{b}_g} & \frac{\partial \beta_8}{\partial \boldsymbol{b}_a} & \frac{\partial \beta_8}{\partial \boldsymbol{p}_c^b} & \frac{\partial \beta_8}{\partial \boldsymbol{\theta}_2}\frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \frac{\partial \beta_8}{\partial \boldsymbol{p}_k^w} \end{bmatrix}, \\ &= \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}\end{aligned} \tag{4.53}$$

with,

$$\begin{aligned}\frac{\partial \beta_6}{\partial \boldsymbol{x}}\boldsymbol{f}_0 &= \mathbf{0} \\ \frac{\partial \beta_6}{\partial \boldsymbol{x}}\boldsymbol{f}_1 &= \mathbf{0}\cdot \\ \frac{\partial \beta_6}{\partial \boldsymbol{x}}\boldsymbol{f}_2 &= \mathbf{0}\end{aligned} \tag{4.54}$$

Besides, there is another measurement model $\boldsymbol{z}_l$ in the proposed method, which is similar to the basis function β_1. We define it as

$$\beta_9 = \boldsymbol{z}_l = \frac{1}{p_{z_l}}\begin{bmatrix} p_{x_l} \\ p_{y_l} \end{bmatrix}, \tag{4.55}$$

where $\boldsymbol{z}_l$ is the pixel coordinate of an endpoint of a line feature under the normalized camera model.

Therefore, referring to the calculation of the basis function β_1, we have

$$\begin{aligned}\frac{\partial \beta_9}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{\partial \beta_9}{\partial \boldsymbol{p}} & \frac{\partial \beta_9}{\partial \boldsymbol{v}} & \frac{\partial \beta_9}{\partial \boldsymbol{\theta}_1}\frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \frac{\partial \beta_9}{\partial \boldsymbol{b}_g} & \frac{\partial \beta_9}{\partial \boldsymbol{b}_a} & \frac{\partial \beta_9}{\partial \boldsymbol{p}_c^b} & \frac{\partial \beta_9}{\partial \boldsymbol{\theta}_2}\frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \frac{\partial \beta_9}{\partial \boldsymbol{p}_k^w} \end{bmatrix} \\ &= \frac{\partial \boldsymbol{z}_l}{\partial \boldsymbol{P}_l^c}\cdot\frac{\partial \boldsymbol{P}_l^c}{\partial \boldsymbol{x}} \\ &= \begin{bmatrix} \frac{1}{p_{z_l}} & 0 & -\frac{p_{x_l}}{p_{z_l}^2} \\ 0 & \frac{1}{p_{z_l}} & -\frac{p_{y_l}}{p_{z_l}^2} \end{bmatrix}\begin{bmatrix} \boldsymbol{C}_2\boldsymbol{C}_1 & \mathbf{0}_3 & \boldsymbol{C}_2\lfloor \boldsymbol{P}_l^b \rfloor_\times \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{C}_2 & \lfloor \boldsymbol{P}_l^c \rfloor_\times \frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \mathbf{0}_3 \end{bmatrix}\end{aligned}. \tag{4.56}$$

with

$$\begin{aligned}\frac{\partial \beta_9}{\partial \boldsymbol{x}}\boldsymbol{f}_0 &= \begin{bmatrix} \frac{1}{p_{z_l}} & 0 & -\frac{p_{x_l}}{p_{z_l}^2} \\ 0 & \frac{1}{p_{z_l}} & -\frac{p_{y_l}}{p_{z_l}^2} \end{bmatrix}(\boldsymbol{C}_2\boldsymbol{C}_1\boldsymbol{v} - \boldsymbol{C}_2\lfloor \boldsymbol{P}_l^b \rfloor_\times \boldsymbol{b}_g) \\ &= \begin{bmatrix} \boldsymbol{I}_2 & -\beta_9 \end{bmatrix}(\frac{1}{p_{z_l}}\boldsymbol{C}_2\boldsymbol{C}_1\boldsymbol{v} - \lfloor \begin{bmatrix} \beta_9 \\ 1 \end{bmatrix} + \frac{1}{p_{z_l}}\boldsymbol{C}_2\boldsymbol{p}_c^b \rfloor_\times \boldsymbol{C}_2\boldsymbol{b}_g).\end{aligned} \tag{4.57}$$

Here, we need to define a new basis function

$$\beta_{10} \triangleq \frac{1}{p_{z_l}} \tag{4.58}$$

then,

$$\begin{aligned}
&\frac{\partial \beta_9}{\partial \boldsymbol{x}} \boldsymbol{f}_0 = \begin{bmatrix} \boldsymbol{I}_2 & -\beta_9 \end{bmatrix} (\beta_{10}\beta_4\beta_3 - \lfloor \begin{bmatrix} \beta_9 \\ 1 \end{bmatrix} + \beta_{10}\beta_4\beta_5 \rfloor_\times \beta_4\beta_6) \\
&\frac{\partial \beta_9}{\partial \boldsymbol{x}} \boldsymbol{f}_1 = \begin{bmatrix} \frac{1}{p_{z_l}} & 0 & -\frac{p_{x_l}}{p_{z_l}^2} \\ 0 & \frac{1}{p_{z_l}} & -\frac{p_{y_l}}{p_{z_l}^2} \end{bmatrix} \boldsymbol{C}_2 \lfloor \boldsymbol{P}_k^b \rfloor_\times = \begin{bmatrix} \boldsymbol{I}_2 & -\beta_9 \end{bmatrix} \beta_4 \lfloor \begin{bmatrix} \beta_9 \\ 1 \end{bmatrix} + \beta_{10}\beta_4\beta_5 \rfloor_\times \beta_4 \\
&\frac{\partial \beta_9}{\partial \boldsymbol{x}} \boldsymbol{f}_1 = \boldsymbol{0}
\end{aligned} \tag{4.59}$$

For basis function $\beta_{10} = \frac{1}{p_{z_l}}$, the span space is

$$\begin{aligned}
\frac{\partial \beta_{10}}{\partial \boldsymbol{x}} &= \begin{bmatrix} \frac{\partial \beta_{10}}{\partial \boldsymbol{p}} & \frac{\partial \beta_{10}}{\partial \boldsymbol{v}} & \frac{\partial \beta_{10}}{\partial \theta_1}\frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \frac{\partial \beta_{10}}{\partial \boldsymbol{b}_g} & \frac{\partial \beta_{10}}{\partial \boldsymbol{b}_a} & \frac{\partial \beta_{10}}{\partial \boldsymbol{p}_c^b} & \frac{\partial \beta_{10}}{\partial \theta_2}\frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \frac{\partial \beta_{10}}{\partial \boldsymbol{p}_k^w} \end{bmatrix} \\
&= \frac{\partial \beta_{10}}{\partial \boldsymbol{P}_l^c} \cdot \frac{\partial \boldsymbol{P}_l^c}{\partial \boldsymbol{x}} \qquad , \\
&= -\frac{1}{p_{z_l}^2} \boldsymbol{e}_3^\top \begin{bmatrix} \boldsymbol{C}_2\boldsymbol{C}_1 & \boldsymbol{0}_3 & \boldsymbol{C}_2 \lfloor \boldsymbol{P}_l^b \rfloor_\times \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{C}_2 & \lfloor \boldsymbol{P}_l^c \rfloor_\times \frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \boldsymbol{0}_3 \end{bmatrix}
\end{aligned} \tag{4.60}$$

with,

$$\begin{aligned}
&\frac{\partial \beta_{10}}{\partial \boldsymbol{x}} \boldsymbol{f}_0 = -\beta_{10} \boldsymbol{e}_3^\top (\beta_{10}\beta_4\beta_3 - \lfloor \begin{bmatrix} \beta_9 \\ 1 \end{bmatrix} + \beta_{10}\beta_4\beta_5 \rfloor_\times \beta_4\beta_6) \\
&\frac{\partial \beta_{10}}{\partial \boldsymbol{x}} \boldsymbol{f}_1 = -\beta_{10} \boldsymbol{e}_3^\top \beta_4 \lfloor \begin{bmatrix} \beta_9 \\ 1 \end{bmatrix} + \beta_{10}\beta_4\beta_5 \rfloor_\times \beta_4 \\
&\frac{\partial \beta_{10}}{\partial \boldsymbol{x}} \boldsymbol{f}_2 = \boldsymbol{0}.
\end{aligned} \tag{4.61}$$

Until now, we have obtained the new nonlinear system constructed with all basis functions $\boldsymbol{\beta}$ based on *Theorem 1 C(3)*

$$\begin{aligned}\dot{\boldsymbol{\beta}} = \frac{\partial\boldsymbol{\beta}}{\partial\boldsymbol{x}}\frac{\partial\boldsymbol{x}}{\partial t} = \frac{\partial\boldsymbol{\beta}}{\partial\boldsymbol{x}}\dot{\boldsymbol{x}} = \frac{\partial\boldsymbol{\beta}}{\partial\boldsymbol{x}}(\boldsymbol{f}_0+\boldsymbol{f}_1\boldsymbol{\omega}+\boldsymbol{f}_2\boldsymbol{a}) = \boldsymbol{g}_0+\boldsymbol{g}_1\boldsymbol{\omega}+\boldsymbol{g}_2\boldsymbol{a}\\ \boldsymbol{z}_1 = \beta_1\\ \boldsymbol{z}_2 = \beta_9\end{aligned} \tag{4.62}$$

According to the *Theorem 1 (ii)*, $null(\mathcal{O}) = null(\boldsymbol{B})$ with $\boldsymbol{B} \triangleq \frac{\partial\boldsymbol{\beta}}{\partial\boldsymbol{x}}$. Combining (4.30), (4.39), (4.41), (4.45), (4.47), (4.49), (4.51), (4.53), (4.56), (4.60), we get $\boldsymbol{B}$

$$\begin{aligned}\boldsymbol{B} &\triangleq \boldsymbol{B}_1\cdot\boldsymbol{B}_2\\ &= \begin{bmatrix}\zeta & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \eta & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3\end{bmatrix}\cdot\\ &\begin{bmatrix}\boldsymbol{C}_2\boldsymbol{C}_1 & \mathbf{0}_3 & \boldsymbol{C}_2\lfloor\boldsymbol{P}_k^b\rfloor_\times\frac{\partial\theta_1}{\partial\boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{C}_2 & \lfloor\boldsymbol{P}_k^c\rfloor_\times\frac{\partial\theta_2}{\partial\boldsymbol{s}_2} & -\boldsymbol{C}_2\boldsymbol{C}_1\\ \boldsymbol{C}_2\boldsymbol{C}_1 & \mathbf{0}_3 & \boldsymbol{C}_2\lfloor\boldsymbol{P}_l^b\rfloor_\times\frac{\partial\theta_1}{\partial\boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{C}_2 & \lfloor\boldsymbol{P}_l^c\rfloor_\times\frac{\partial\theta_2}{\partial\boldsymbol{s}_2} & \mathbf{0}_3\\ \mathbf{0}_3 & \boldsymbol{C}_1 & \lfloor\boldsymbol{C}_1\boldsymbol{v}\rfloor_\times\frac{\partial\theta_1}{\partial\boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \lfloor\boldsymbol{C}_1\boldsymbol{g}^w\rfloor_\times\frac{\partial\theta_1}{\partial\boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3\end{bmatrix}\end{aligned} \tag{4.63}$$

where,

$$\zeta = \begin{bmatrix}\frac{1}{p_z} & 0 & -\frac{p_x}{p_z^2}\\ 0 & \frac{1}{p_z} & -\frac{p_y}{p_z^2}\\ 0 & 0 & -\frac{1}{p_z^2}\end{bmatrix}, \quad \eta = \begin{bmatrix}\frac{1}{p_{z_l}} & 0 & -\frac{p_{x_l}}{p_{z_l}^2}\\ 0 & \frac{1}{p_{z_l}} & -\frac{p_{y_l}}{p_{z_l}^2}\\ 0 & 0 & -\frac{1}{p_{z_l}^2}\end{bmatrix}. \tag{4.64}$$

where the first row of the matrix $\boldsymbol{B}_2$ comes from the combination of $\frac{\partial\beta_1}{\partial\boldsymbol{x}}$ and $\frac{\partial\beta_2}{\partial\boldsymbol{x}}$. Similarly, the second row of the matrix $\boldsymbol{B}_2$ is the combination of $\frac{\partial\beta_9}{\partial\boldsymbol{x}}$ and $\frac{\partial\beta_{10}}{\partial\boldsymbol{x}}$. For ease of operation, we adjust it to the second row of $\boldsymbol{B}_2$.

It is obvious that $\boldsymbol{B}_1$ is full rank. The null space of $\boldsymbol{B}_2$ is the same as that of $\boldsymbol{B}$. Assuming the left null space of $\boldsymbol{B}_2$ is $\boldsymbol{N}_L = [\boldsymbol{N}_1, \boldsymbol{N}_2, \boldsymbol{N}_3, \boldsymbol{N}_4, \boldsymbol{N}_5, \boldsymbol{N}_6, \boldsymbol{N}_7, \boldsymbol{N}_8]$, we have

$$\mathbf{0} = \begin{bmatrix} \boldsymbol{N}_1 \\ \boldsymbol{N}_2 \\ \boldsymbol{N}_3 \\ \boldsymbol{N}_4 \\ \boldsymbol{N}_5 \\ \boldsymbol{N}_6 \\ \boldsymbol{N}_7 \\ \boldsymbol{N}_8 \end{bmatrix}^\top \cdot \begin{bmatrix} \boldsymbol{C}_2\boldsymbol{C}_1 & \mathbf{0}_3 & \boldsymbol{C}_2\lfloor \boldsymbol{P}_k^b \rfloor_\times \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{C}_2 & \lfloor \boldsymbol{P}_k^c \rfloor_\times \frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & -\boldsymbol{C}_2\boldsymbol{C}_1 \\ \boldsymbol{C}_2\boldsymbol{C}_1 & \mathbf{0}_3 & \boldsymbol{C}_2\lfloor \boldsymbol{P}_l^b \rfloor_\times \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{C}_2 & \lfloor \boldsymbol{P}_l^c \rfloor_\times \frac{\partial \boldsymbol{\theta}_2}{\partial \boldsymbol{s}_2} & \mathbf{0}_3 \\ \mathbf{0}_3 & \boldsymbol{C}_1 & \lfloor \boldsymbol{C}_1 \boldsymbol{v} \rfloor_\times \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \lfloor \boldsymbol{C}_1 \boldsymbol{g}^w \rfloor_\times \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}. \tag{4.65}$$

It can be seen from (4.65) that $\boldsymbol{N}_3\boldsymbol{C}_1 = \mathbf{0}$, $\boldsymbol{N}_6\boldsymbol{I}_1 = \mathbf{0}$, $\boldsymbol{N}_8\boldsymbol{I}_1 = \mathbf{0}$, and $-\boldsymbol{N}_1\boldsymbol{C}_2\boldsymbol{C}_1 = \mathbf{0}$, so $\boldsymbol{N}_1 = 0$, $\boldsymbol{N}_3 = 0$, $\boldsymbol{N}_6 = 0$, and $\boldsymbol{N}_8 = 0$. Then $\boldsymbol{N}_2\boldsymbol{C}_2\boldsymbol{C}_1 = \mathbf{0} \rightarrow \boldsymbol{N}_2 = \mathbf{0}$, so $\boldsymbol{N}_5 = \mathbf{0}$ and $\boldsymbol{N}_4 = \mathbf{0}$ based on the sixth and seventh columns of matrix $\boldsymbol{B}_2$. Therefore, only the $\boldsymbol{N}_7$ is non-zero vector, and it can be seen from the third column of $\boldsymbol{B}_2$,

$$\boldsymbol{N}_7 \lfloor \boldsymbol{C}_1 \boldsymbol{g}^w \rfloor_\times \frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1} = \mathbf{0}. \tag{4.66}$$

Since the matrix $\frac{\partial \boldsymbol{\theta}_1}{\partial \boldsymbol{s}_1}$ is full rank, we have

$$\boldsymbol{N}_7 = \pm(\boldsymbol{C}_1 \boldsymbol{g}^w)^\top, \tag{4.67}$$

because of the another skew symmetric matrix operation rule,

$$\lfloor \boldsymbol{a} \rfloor_\times \boldsymbol{a} = \mathbf{0}, \quad \boldsymbol{a}^\top \lfloor \boldsymbol{a} \rfloor_\times = \mathbf{0} \tag{4.68}$$

In summary, the left null space of $\boldsymbol{B}_2$ is

$$\boldsymbol{N}_L = [\mathbf{0}_{1\times3}, \mathbf{0}_{1\times3}, \mathbf{0}_{1\times3}, \mathbf{0}_{1\times3}, \mathbf{0}_{1\times3}, \mathbf{0}_{1\times3}, (\boldsymbol{C}_1 \boldsymbol{g}^w)^\top, \mathbf{0}_{1\times3}]_{1\times24} \tag{4.69}$$

Its dimension is 1. Therefore, the dimension of the right null space of 24×24 matrix $\boldsymbol{B}_2$ is also 1. The right null space of $\boldsymbol{B}_2$ is,

$$\boldsymbol{N}_R = \begin{bmatrix} \lfloor \boldsymbol{C}_1^\top \boldsymbol{P}_c^b \rfloor_\times \boldsymbol{g}^w \\ -\lfloor \boldsymbol{v} \rfloor_\times \boldsymbol{g}^w \\ \frac{\partial \boldsymbol{s}_1}{\partial \theta_1} \boldsymbol{C}_1 \boldsymbol{g}^w \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_3 \\ \boldsymbol{0}_3 \\ -\frac{\partial \boldsymbol{s}_2}{\partial \theta_2} \boldsymbol{C}_2 \boldsymbol{C}_1 \boldsymbol{g}^w \\ \boldsymbol{0}_3 \end{bmatrix}_{24 \times 1}. \tag{4.70}$$

The definition of unobservable state variables is as follows: a small perturbation is applied to state variables utilizing the span of right null space, state variables affected by the disturbance are unobservable [112]. It can be seen from $\boldsymbol{N}_R$ that the only 1 dimension right null space is related to the gravity $\boldsymbol{g}^w$. Here, we use $\boldsymbol{N}_R$ to exert a small perturbation on states, which is a small rotation disturbance around the gravity $\boldsymbol{R}_w^{w'}$, and we have

$$\boldsymbol{R}_w^{w'} \approx \boldsymbol{I}_3 - c\lfloor \bar{\boldsymbol{g}}^w \rfloor_\times, \tag{4.71}$$

where, c is a constant, $\bar{\boldsymbol{g}}^w$ is the unit vector along the direction of gravity.

Taking the rotation matrix $\boldsymbol{C}_1 = \boldsymbol{R}_w^b$ as an example, the impact of $\boldsymbol{R}_w^{w'}$ on the rotation matrix is,

$$\begin{aligned} \boldsymbol{R}_{w'}^b &\approx (\boldsymbol{I}_3 - \lfloor \delta\boldsymbol{\theta} \rfloor_\times)\boldsymbol{R}_w^b \\ &= (\boldsymbol{I}_3 - \lfloor \delta\boldsymbol{\theta} \rfloor_\times)\boldsymbol{R}_{w'}^b \boldsymbol{R}_w^{w'} = (\boldsymbol{I}_3 - \lfloor \delta\boldsymbol{\theta} \rfloor_\times)\boldsymbol{R}_{w'}^b (\boldsymbol{I}_3 - c\lfloor \bar{\boldsymbol{g}}^w \rfloor_\times) \\ &= \boldsymbol{R}_{w'}^b - \lfloor \delta\boldsymbol{\theta} \rfloor_\times \boldsymbol{R}_{w'}^b - c\boldsymbol{R}_{w'}^b \lfloor \bar{\boldsymbol{g}}^w \rfloor_\times + c\lfloor \delta\boldsymbol{\theta} \rfloor_\times \boldsymbol{R}_{w'}^b \lfloor \bar{\boldsymbol{g}}^w \rfloor_\times \\ &\approx \boldsymbol{R}_{w'}^b - \lfloor \delta\boldsymbol{\theta} \rfloor_\times \boldsymbol{R}_{w'}^b - c\boldsymbol{R}_{w'}^b \lfloor \bar{\boldsymbol{g}}^w \rfloor_\times \end{aligned} \tag{4.72}$$

$\Longrightarrow$

$$\begin{aligned} \lfloor \delta\boldsymbol{\theta} \rfloor_\times \boldsymbol{R}_{w'}^b &= -c\boldsymbol{R}_{w'}^b \lfloor \bar{\boldsymbol{g}}^w \rfloor_\times \\ \lfloor \delta\boldsymbol{\theta} \rfloor_\times &= -c\boldsymbol{R}_{w'}^b \lfloor \bar{\boldsymbol{g}}^w \rfloor_\times {\boldsymbol{R}_{w'}^b}^\top = -c\lfloor \boldsymbol{R}_{w'}^b \bar{\boldsymbol{g}}^w \rfloor_\times \end{aligned} \tag{4.73}$$

$\Longrightarrow$

$$\begin{aligned}\delta\theta &= -c\boldsymbol{R}^b_{w'}\bar{\boldsymbol{g}}^w \\ &= -c\boldsymbol{R}^b_w\boldsymbol{R}^{w'}_w\bar{\boldsymbol{g}}^w = -c\boldsymbol{R}^b_w\bar{\boldsymbol{g}}^w \\ &= -\frac{c}{\parallel \boldsymbol{g}^w \parallel_2}\boldsymbol{R}^b_w\boldsymbol{g}^w\end{aligned} \tag{4.74}$$

since $\boldsymbol{R}^{w'}_w$ express a rotation around the direction of gravity, then $\boldsymbol{R}^{w'}_w\bar{\boldsymbol{g}}^w = \bar{\boldsymbol{g}}^w$. Based on (4.72) (4.73) (4.74), we have the perturbation of matrix $\boldsymbol{R}^b_w$ is

$$\delta\boldsymbol{s}_1 = \frac{\partial\boldsymbol{s}_1}{\partial\theta_1}\delta\theta = -\frac{c}{\parallel \boldsymbol{g}^w \parallel_2}\frac{\partial\boldsymbol{s}_1}{\partial\theta_1}\boldsymbol{R}^b_w\boldsymbol{g}^w = -\frac{c}{\parallel \boldsymbol{g}^w \parallel_2}\frac{\partial\boldsymbol{s}}{\partial\theta}\boldsymbol{C}_1\boldsymbol{g}^w. \tag{4.75}$$

For rotation matrix $\boldsymbol{C}_2 = \boldsymbol{R}^b_c$, we have a similar derivation. For states other than rotation matrix, we take state $\boldsymbol{v}$ as an example,

$$\begin{aligned}\boldsymbol{v}' &= \boldsymbol{R}^{w'}_w\boldsymbol{x} \approx (\boldsymbol{I}_3 - c\lfloor\bar{\boldsymbol{g}}^w\rfloor_\times)\boldsymbol{v} \\ &= \boldsymbol{x} - c\lfloor\bar{\boldsymbol{g}}^w\rfloor_\times\boldsymbol{v} = \boldsymbol{x} - \frac{c}{\parallel \boldsymbol{g}^w \parallel_2}(-\lfloor\boldsymbol{v}\rfloor_\times\boldsymbol{g}^w)\end{aligned} \tag{4.76}$$

therefore, the perturbation of state $\boldsymbol{v}$ is

$$\delta\boldsymbol{v} = -\frac{c}{\parallel \boldsymbol{g}^w \parallel_2}(-\lfloor\boldsymbol{v}\rfloor_\times\boldsymbol{g}^w) \tag{4.77}$$

Combining (4.75) and (4.77), it can be seen that

$$\delta\boldsymbol{x} = -\frac{c}{\parallel \boldsymbol{g}^w \parallel_2}\boldsymbol{N}_R \tag{4.78}$$

The impact of states caused by the small perturbation induced by the rotation around the vector of gravity is shown in (4.78), which means that the rotation around the gravity (yaw angle) is unobservable. Among that, the first row of (4.78) is not strictly equal to $-\frac{c}{\|\boldsymbol{g}^w\|_2}(-\lfloor\boldsymbol{p}\rfloor_\times\boldsymbol{g}^w)$. However, $\boldsymbol{C}_1^\top\boldsymbol{P}^b_c = \boldsymbol{R}^w_b\boldsymbol{P}^b_c$ can be regarded as the mapping of the extrinsic translation in the global coordinate system.

In summary, the observability analysis indicates that the global x, y, z translation directions of the proposed system after fusing prior line map are observable. The only unobservable direction corresponds to global rotation about the gravity vector (yaw angle).

4.4 Experiments & Discussions

In this section, we evaluate the performance of the proposed method through experiments in both simulation and real-world environments. The simulation environment is conducted in the CARLA simulator [139] and the real-world experiments are based on the public EuRoC micro aerial vehicle (MAV) dataset [134]. The comparison methods we executed include

1. **VINS**: the VINS-Mono method without loop-closure module that is developed by Qin [1].

2. **Benchmark**: a loosely coupled structure between prior line feature factor and VIO proposed by Yu [36].

3. **LCL**: our previous proposed **l**oosely **c**oupled **l**ine feature-aided framework with outlier rejection module [3].

4. **TCL**: the proposed **t**ightly **c**oupled **l**ine-aided framework in this work.

The criteria of localization accuracy used in this section are the root mean squared error (RMSE), absolute trajectory error (ATE) [148], and Euclidean distance error. All methods are implemented on Ubuntu 20.04 with robot operating system (ROS) and conducted on a PC with Inter-Core i9-12900K and 32 GB memory.

4.4.1 Simulation Experiments

Figure 4.5: Two driving trajectories and scenarios in CARLA simulator. The red and green curves represent the trajectories of 1km and 2km, respectively.

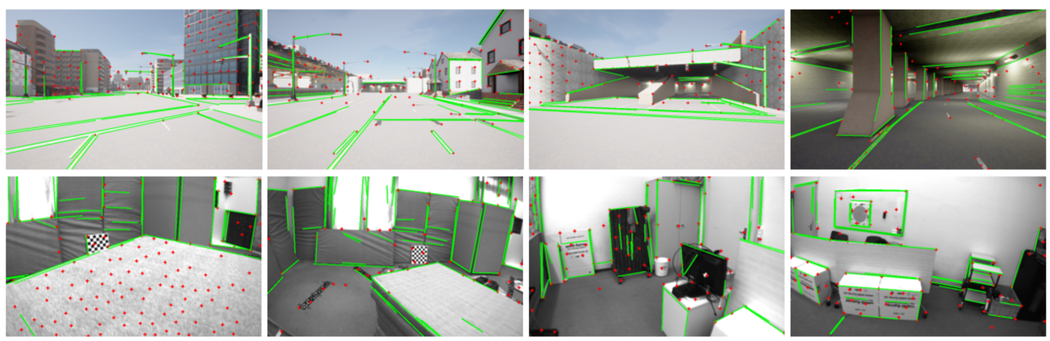


Figure 4.6: Images in the top row show simulated urban environments on CARLA and images in the bottom row display indoor scenarios of the EuRoC dataset. Red marks are the point features and green lines denote the line features.

Setup

CARLA simulator is widely used in autonomous driving and SLAM domains. It supports flexible sensor combinations, such as cameras, LiDAR, IMU, and radar, etc al. We can synthesize diverse scenarios to simulate real-world environments. In this section, we apply an unmanned ground vehicle (UGV) running in urban environments. A LiDAR (64 channels, 100 Hz), IMU (100 Hz), and camera (960x600, 30 Hz) are used to collect datasets. The UGV runs two different trajectories with

Table 4.1: RMSE of ATE (m) on CARLA dataset

Sequence	VINS	Benchmark	LCL	TCL
CARLA_01	2.070	2.726	1.931	**1.712**
CARLA_02	4.613	4.312	3.621	**3.205**

lengths of 1km and 2km in a town, which are shown in Fig. 4.5 and called CARLA_01 and CARLA_02. Some selected city and tunnel scenes in this map and feature detection results are shown in Fig. 4.6. It can be seen that lines can provide more robust features than points in challenging scenarios with low texture and high similarity.

Results

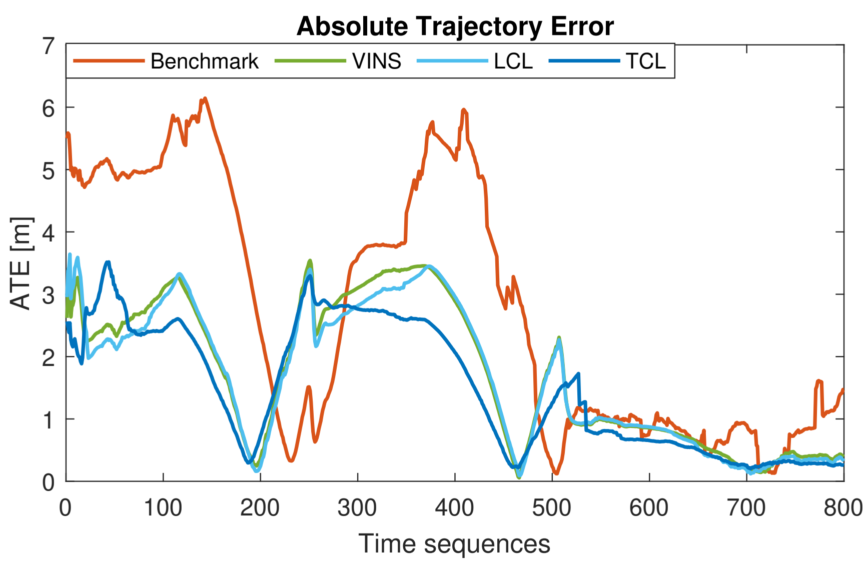


Figure 4.7: The ATE results of the CARLA_01 dataset.

We have named the simulation trajectories CARLA_01 and CARLA_02 according to their lengths. Fig. 4.7 shows the ATE comparison results of VINS-Mono, Benchmark, LCL, and TCL based on dataset CARLA_01. We can see that our proposed TCL method outperforms others. Our methods greatly improve the computational stability compared to Benchmark and have a better absolute positioning

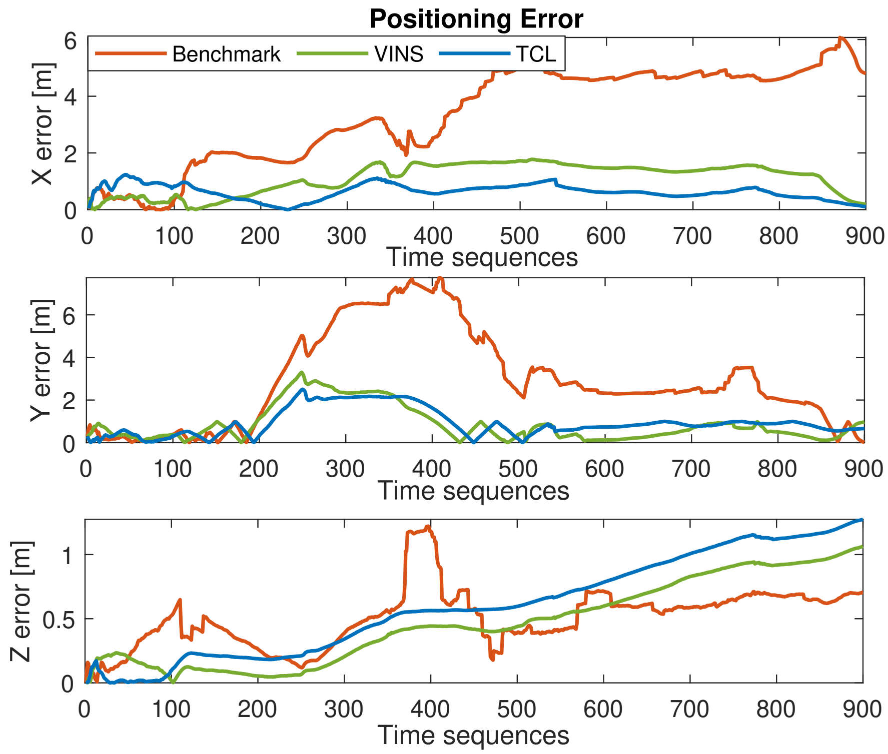


Figure 4.8: The positioning error of CARLA_01 dataset.

accuracy than VINS-Mono. The RMSE of ATE results about two trajectories are displayed in Table 4.1. On three sets of simulation datasets, our proposed TCL method has an average improvement of 11.44% in positioning accuracy compared to LCL, 26.42% compared to VINS-Mono, and 30.15% compared to Benchmark.

As an example, Fig. 4.8 and Fig. 4.9 depicts the positioning error and rotation error of CARLA_01 compared with VINS-Mono and Benchmark, respectively. It can be observed that the proposed method has better accuracy and robustness performances than Benchmark in six directions and its error distribution does not exhibit as many abrupt changes as Benchmark. This is one of the advantages of tightly coupled structures, which have smoother trajectories than loosely coupled structures and are similar to VINS-Mono. Compared to VINS-Mono, TCL has lower errors in x-direction positioning and yaw-angle results, and it has shown some improvement in accuracy in the y-direction, roll, and pitch rotation angles. However, the position-

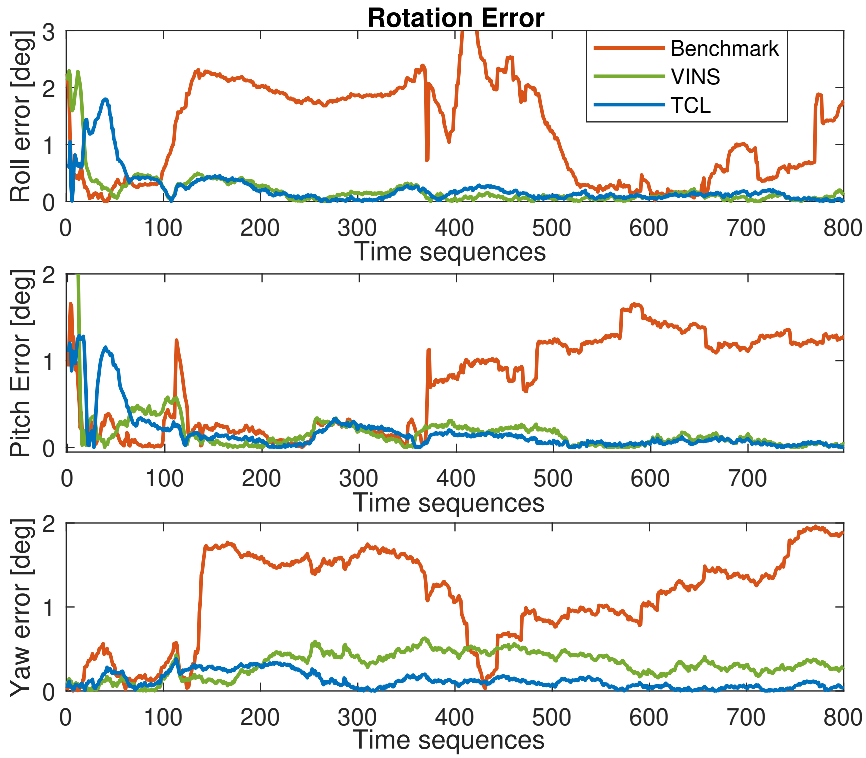


Figure 4.9: The Rotation error of CARLA_01 dataset

ing result in the z-direction is somewhat inferior to that of VINS-Mono. Combined with the observability analysis in the previous section, this result is reasonable. The proposed method can alleviate cumulative drifts, but cannot completely solve this problem.

4.4.2 Real-world Experiments

setup

We have also assessed the performance of the proposed method based on real-world environments. The first dataset we utilized is part of the public EuRoC dataset, which consists of six sequences recorded by an MAV flying through two distinct rooms with three different challenges. These challenges are related to the motion and speed of the MAV, illumination, and texture of environments. Besides,

(a) (b)

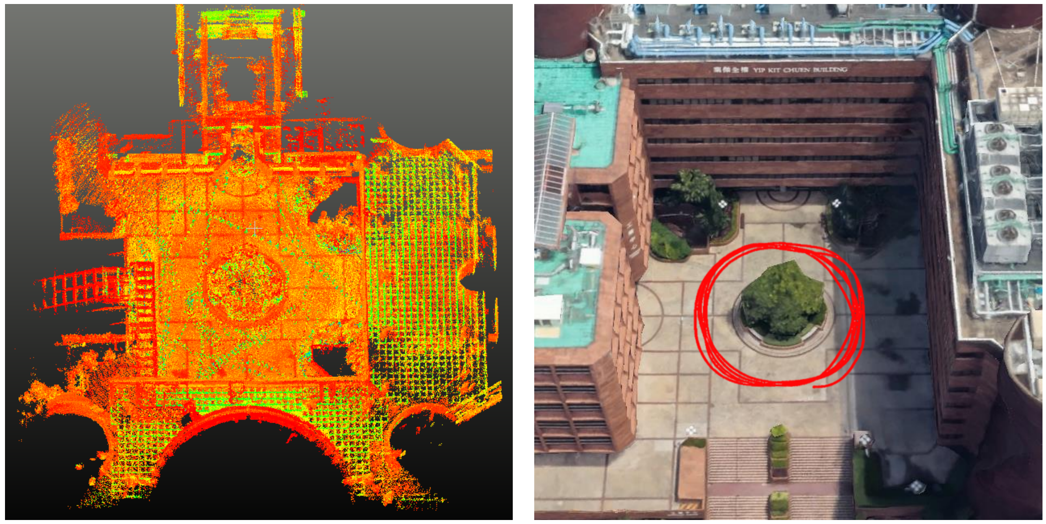
Figure 4.10: (a) The vehicle platform. (b) Sensor unit

the prior point cloud maps of rooms and the ground truth are collected by a Leica MS50 scanner, Vicon motion capture platform, IMU (ADIS16448, 200 Hz), and cameras (640×480, 30 Hz) [134]. The environments of indoor rooms can be seen in Fig. 4.6 and the corresponding line maps are shown in Fig. 4.12.

Besides, we conduct an unmanned ground vehicle (UGV) real-world experiment on the Hong Kong Polytechnic University (PolyU) campus. Fig. 4.10 displays the vehicle platform and the sensor unit, which consists of a LiDAR (HDL 32E Velodyne, 10 Hz), an IMU (Xsens Mti 10, 200 Hz), and a RealSense camera (D435i, 640×480, 30 Hz). Fig. 4.11 shows the collected 3D point cloud of the PolyU square and the motion trajectory of the vehicle on Google Maps. The ground-truth are calculated by LIO_SAM implementation. [16]

Localization Results

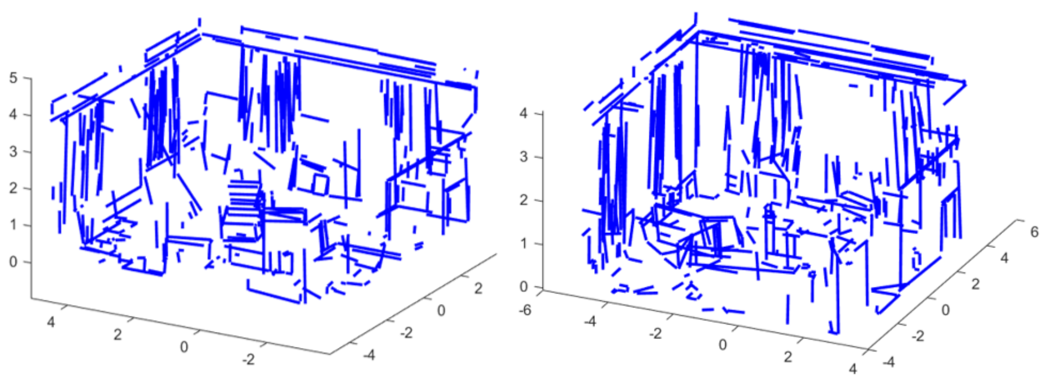
The RMSE of ATE is used to validate the performance of localization accuracy. The results of the proposed method and comparison methods are shown in Table



(a) (b)

Figure 4.11: (a) The 3D point cloud of the PolyU square. (b) Motion trajectory of vehicle on Google 3D map.

4.2. It can be seen that the proposed method is superior to the others on PolyU dataset. Specifically, the accuracy of our algorithm is 17.7% 40.3%, and 13.3% better than VINS, benchmark, and LCL, respectively. The ATE curve results are shown in Fig. 4.13. For EuRoC dataset, the first three and last three datasets in this table correspond to three trajectories in the same room. The same room means the same point cloud prior map and shares a prior lightweight line map. As can be seen



(a) Line map of room 1 (b) Line map of room 2

Figure 4.12: The indoor line maps of two rooms on EuRoC dataset

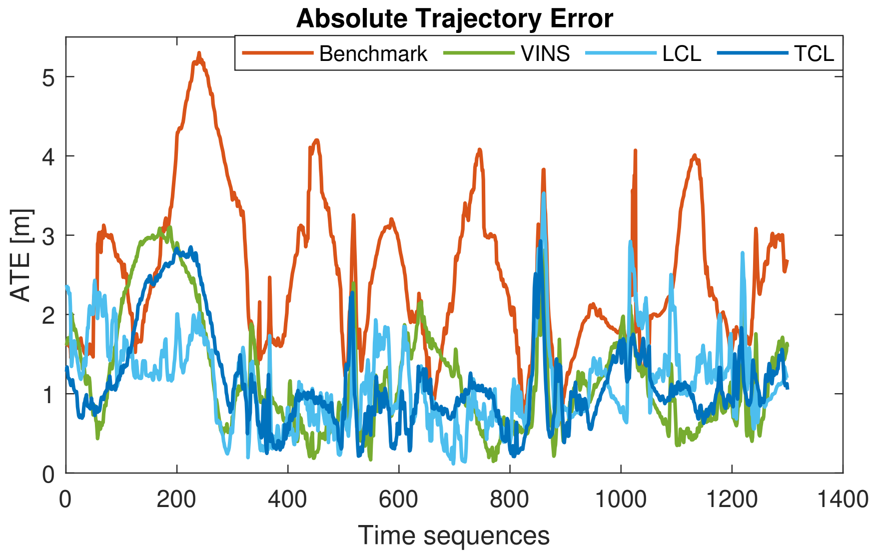


Figure 4.13: The ATE results of the PolyU dataset.

Table 4.2: RMSE of ATE (m) statistics on real-world dataset

Sequence	VINS	Benchmark	LCL	TCL
PolyU	1.588	2.191	1.502	**1.307**
V1_01_esay	0.078	0.164	0.152	**0.068**
V1_02_medium	0.110	0.152	0.092	**0.084**
V1_03_difficult	0.189	0.217	**0.161**	0.182
V2_01_esay	0.096	0.192	0.166	**0.068**
V2_02_medium	0.167	0.281	0.136	**0.108**
V2_03_difficult	0.253	0.341	**0.218**	0.254

in this table, the results in the same room show the effectiveness of the proposed method in performing different trajectories based on the same prior map. It can also be seen that the method proposed in this work outperforms others in the easy and medium scenarios and barely increases the accuracy in the difficult scenarios. The reason is that the performance of the prior map-aided localization method relies on the initial guess of the cross-modality matching. However, in difficult scenarios, it is unstable to provide a good enough initial guess for the tightly coupled method.

As an example, the positioning error and rotation error of the V1_02_medium dataset are shown in Fig. 4.14 and Fig. 4.15, respectively. It can be seen that

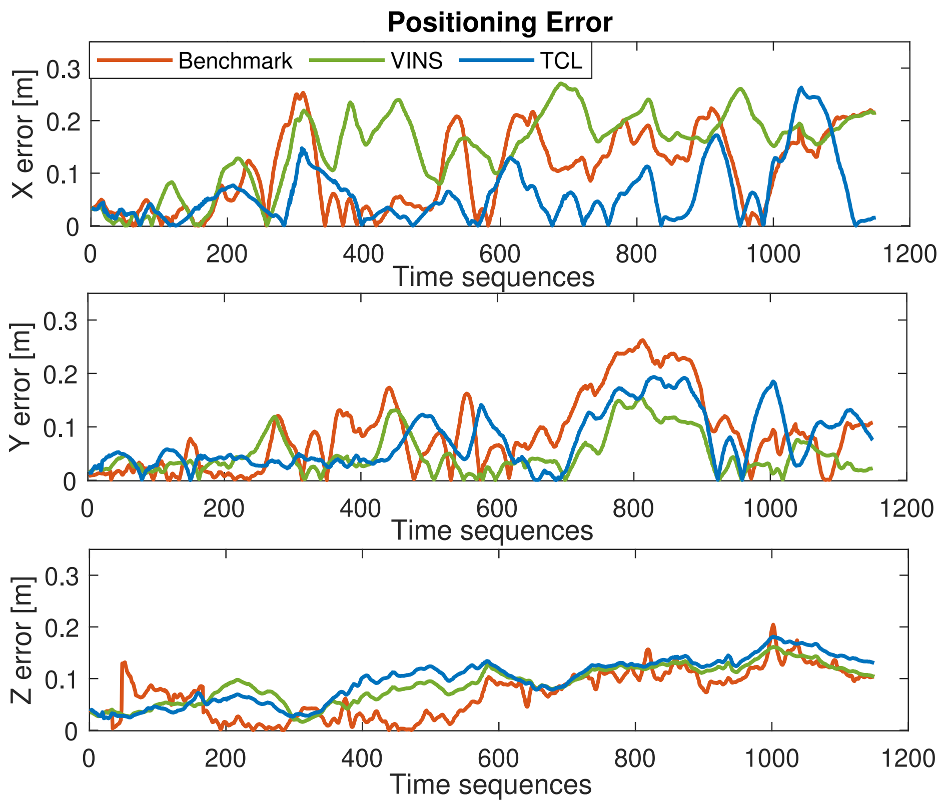


Figure 4.14: The positioning error of V1_02_medium

our method performs better in the x-direction and yaw rotation angle than other methods, which is consistent with the simulation experiments. In addition, the trajectories of ground truth, VINS-Mono, and ours based on the EuRoC dataset V1_02_medium are illustrated in Fig. 4.16. It can be seen that the proposed method has better performance than VINS-Mono.

Time Statistics

The time consumption statistics of VINS, benchmark, LCL, and the proposed TCL method on three datasets are displayed in Table 4.3, where the feature processing includes the image feature detection and tracking module, and the optimization involves sliding window state estimation and marginalization. As seen in the table, after fusing a priori maps in a tightly coupled manner, our algorithm takes more

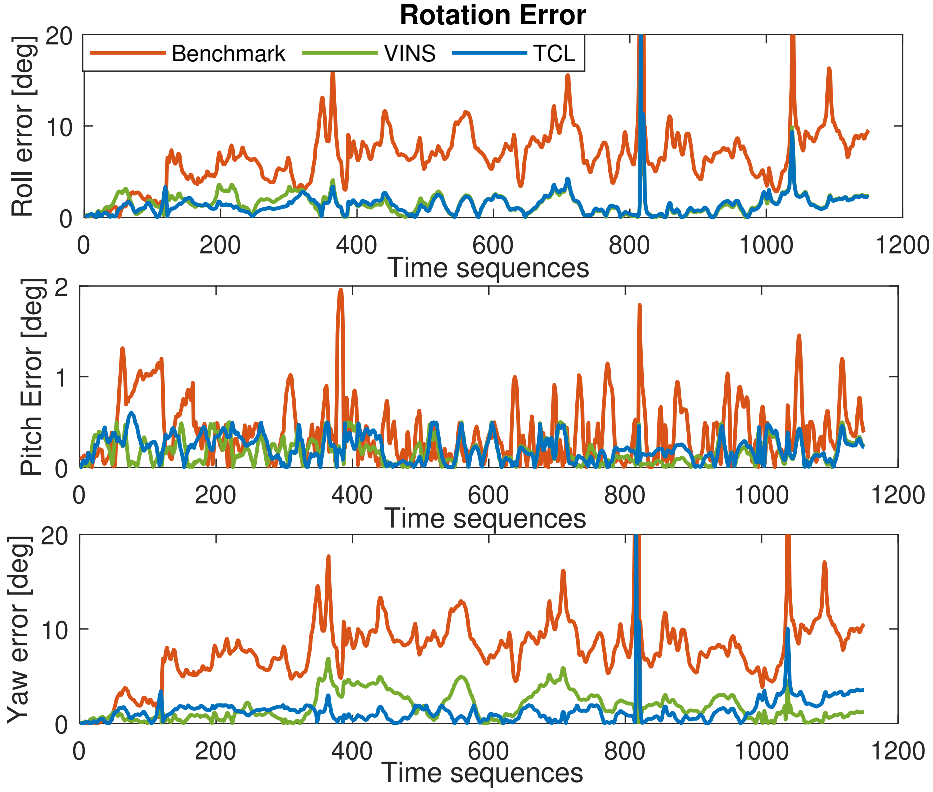


Figure 4.15: The rotation error of V1_02_medium

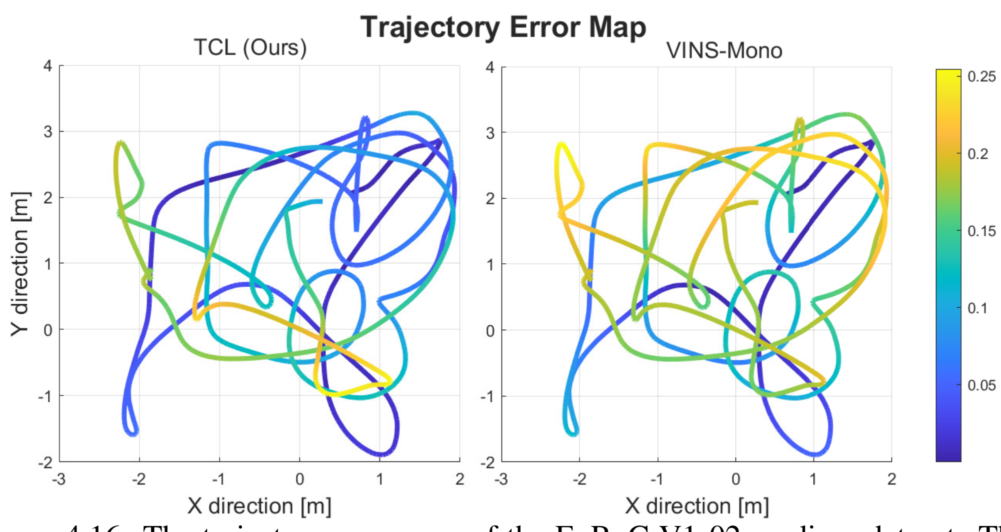


Figure 4.16: The trajectory error map of the EuRoC V1_02_medium dataset. The color of the trajectory corresponds to localization error.

Table 4.3: Time statistics of Feature Processing / Optimization Treads (ms)

Datasets	VINS	Benchmark	LCL	TCL
EuRoC	5.8/17.4	11.3/7.5	9.9/1.5	8.9/25.5
PolyU	2.9/23.3	5.6/4.8	4.8/1.3	4.5/28.1
CARLA_01	8.2/20.1	48.9/52.1	46.5/1.8	24.2/32.2

time than VINS, mainly in feature processing. However, compared to benchmark and LCL in a loosely coupled structure, our method has an advantage in overall time consumption.

4.5 Summary

In this work, we proposed a tightly coupled nonlinear optimization-based state estimation method that integrates the classic VIO and a lightweight prior line map containing IMU pre-integration, point feature, and cross-modality line feature factors. For the line factor, we utilize the simple two endpoints to represent a line feature and present a fast line tracking strategy to monitor and reject outliers for improving the stability of the cross-model line matching. Moreover, a novel line feature-based cost model is proposed and proved in this study, and we also analyze the observability of the proposed framework for the first time. The result shows that global translation directions x, y, and z are observable in this method and only the global yaw rotation is unobservable. Finally, we conducted experiments in both simulated outdoor and real-world indoor environments to verify the performance of our system, and the results show the effectiveness of our method.

Our subsequent work will focus on several identified limitations. First, we will discuss the impact of line feature degeneration on state observability, as our current analysis assumes known endpoint coordinates, unlike the actual optimization which uses segment constraints. Second, the initial map-to-VIO transformation remains critical. To address these issues and further improve robustness, we will explore online estimation of the initial transformation and the integration of additional constraints. For instance, incorporating planar features or a downward-facing laser rangefinder could provide direct observations to mitigate z-axis drift and strengthen overall system stability.

Chapter 5

A Global and Safety-Guaranteed State Estimation Method for Spatial Transformation

The tightly-coupled visual–inertial–prior map framework developed in the previous chapter has further improved localization accuracy by jointly leveraging inertial cues, visual features, and geometric constraints from the prior map. However, several limitations still remain. First, line features may degenerate in certain environments—such as long corridors, planar facades, or repetitive structures—reducing the effectiveness of cross-modal geometric constraints and limiting precision gains. Second, the current outlier detection and rejection mechanisms are neither sufficiently effective nor efficient, potentially compromising robustness. Third, from a safety-quantification perspective, the reliability of predicted state uncertainties fundamentally depends on the convergence quality of the underlying optimization; as reflected in the error-bound formulations, locally optimal solutions may lead to unreliable uncertainty estimations.

To address these limitations, this chapter introduces a two-pronged strategy for safety-aware state estimation. First, we adopt a GNC-TLS–based global optimization framework that provides adjustable, iteration-adaptive outlier rejection

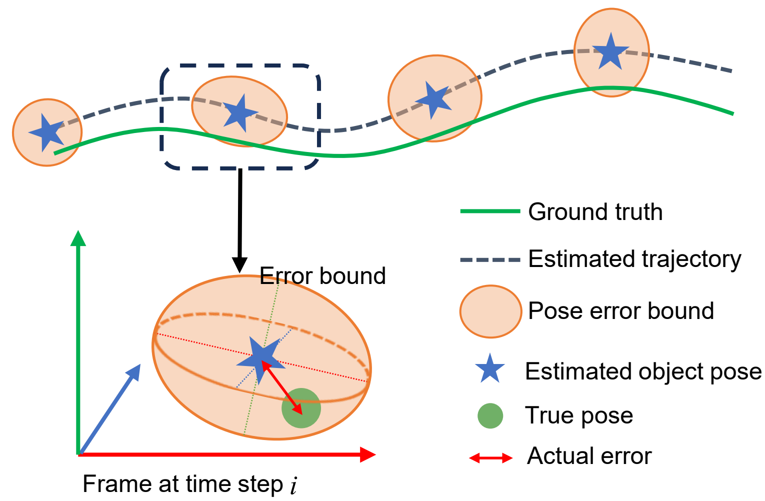


Figure 5.1: Trajectory estimation with pose error bounds. Ground truth (green solid) and estimated trajectory (black dashed) are shown, along with estimated poses (blue stars) and true poses (green dots). Predicted pose error bounds are visualized as orange ellipses. The zoom-in at time step i shows the actual pose error (red arrow) within the predicted bound.

and improves both robustness and estimation accuracy. More importantly, the near-global convergence property of the GNC-TLS formulation helps mitigate the negative impact of local minima on subsequent uncertainty prediction. Second, based on the optimized states obtained under this robust optimization framework, we derive the corresponding error propagation process to predict state uncertainties and compute explicit *error bounds* for surrounding objects, as illustrated in Fig. 5.1. Together, these components enhance the reliability of the estimator and enable safety-guaranteed state uncertainty prediction.

5.1 Preliminaries

Spatial transformations typically consist of state variables such as scale, translation, and rotation, denoted by a scalar s, vector $\boldsymbol{t}$, and matrix $\boldsymbol{R}$ (or quaternion $\boldsymbol{q}$), respectively. Common challenges in robotics and autonomous driving, such as visual localization, point cloud registration, and multi-sensor fusion, can be converted to a spatial transformation issue by solving s, $\boldsymbol{t}$, and $\boldsymbol{R}$. Depending on the type of sensors, the optimization models associated with these problems are generally constructed using either pixel-based reprojection error or spatial-based distance error. Equations (5.1) and (5.2) below present the two classical optimization model:

$$\min_{\boldsymbol{\xi}\in\Xi} \sum_{i\in\mathcal{M}} \left\| \boldsymbol{u}_i - \frac{1}{s_i} \boldsymbol{K} \exp(\boldsymbol{\xi}^\wedge) \boldsymbol{P}_i \right\|^2, \tag{5.1}$$

where $\boldsymbol{u}_i$ is the pixel coordinate of a feature in image, $\boldsymbol{K}$ is the camera intrinsic parameter, $\boldsymbol{\xi}$ is the Lie algebra of object pose $[\boldsymbol{R}|\boldsymbol{t}]$, $\wedge$ denotes the cross product, and $\boldsymbol{P}_i$ is a 3D point in space.

$$\min_{\substack{\boldsymbol{t}\in SO(3)\\ s>0, \boldsymbol{R}\in\mathbb{R}^3}} \sum_{i\in\mathcal{M}} \left\| \boldsymbol{P}_i - (s\boldsymbol{R}\boldsymbol{P}_i' + \boldsymbol{t}) \right\|^2, \tag{5.2}$$

where $\boldsymbol{P}_i$ and $\boldsymbol{P}_i'$ are a pair of 3D correspondence.

The aforementioned spatial transformation problems can be generally cast as state estimation problems and solved via a least-squares optimization framework, which forms the basis for the approaches discussed in this chapter.

5.2 Approximate Global Estimation with Outlier Rejection

To handle outliers in the least-squares optimization problem described above, this section adopts the M-estimation [97] strategy, which introduces a robust loss function $\rho(\cdot)$ into the classical cost function, as shown in,

$$\min_{\boldsymbol{x}\in\mathcal{X}} \sum_{i\in\mathcal{M}} \rho(\|\boldsymbol{r}(\boldsymbol{z}_i, \boldsymbol{x})\|^2). \tag{5.3}$$

However, while the introduction of the loss function enhances robustness against outliers, it also increases the non-convexity of the original optimization problem,

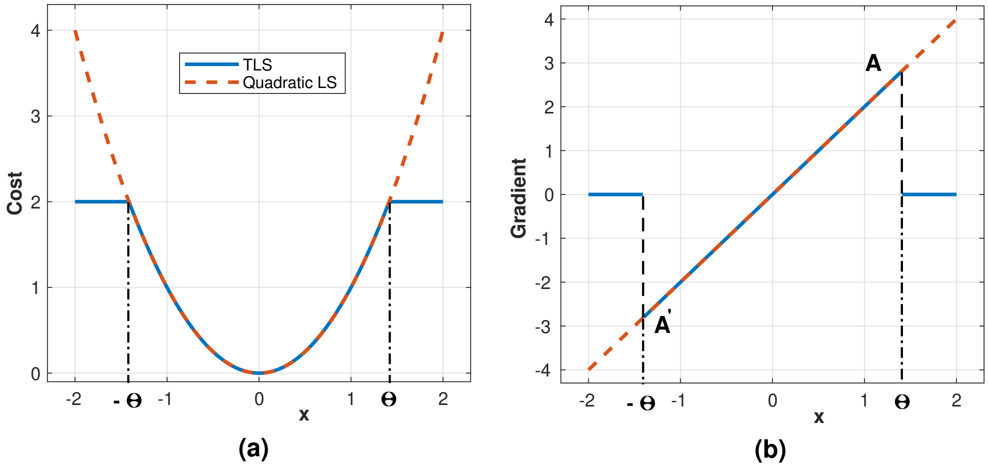


Figure 5.2: (a) the comparison of TLS and quadratic cost with residual. (b) the corresponded gradient with residual.

making convergence to the global optimum more challenging. To address this issue, we employ the GNC method to approximate the original problem as convex. This approach progressively converges towards the global optimum throughout the optimization process.

5.2.1 Truncated Least Squares

The function $\rho(\cdot)$ employed in this work is the truncated least squares (TLS), defined as

$$\rho(r) = \begin{cases} r^2 & |r| \leqslant \Theta \\ \Theta^2 & |r| > \Theta \end{cases} \tag{5.4}$$

where Θ is the threshold used to distinguish between inliers and outliers. The TLS function can be expressed using auxiliary binary variables w_i as follows,

$$\min_{\substack{\boldsymbol{x}\in\mathcal{X} \\ w_i\in\{0,1\}}} \sum_{i\in\mathcal{M}} [w_i \|\boldsymbol{r}(\boldsymbol{z}_i, \boldsymbol{x})\|^2 + (1-w_i)\Theta^2]. \tag{5.5}$$

From (5.5), it is evident that when $w_i = 1$, only the first term contributes to the optimization, indicating that the corresponding measurements are inliers. Conversely, when $w_i = 0$, the constant term Θ^2 remains, signifying that the associated measurement is an outlier, thus ensuring that it does not affect the optimization process and achieving the purpose of outlier rejection.

Taking one-dimension state problem as an example, Fig. 5.2 illustrates the cost curves and gradient changes for TLS and traditional quadratic LS. Compared to quadratic LS, TLS truncates residuals that exceed a specified threshold, resulting in its gradient approaching zero for those values. Meanwhile, the optimization model becomes non-differentiable at points A and A', which means that TLS is strictly non-convex. This is an intractable challenge for optimization algorithms. To address this, we expect to approximate the original problem as a convex one without altering the optimal solution, utilizing methods such as GNC.

5.2.2 Graduated Non-Convexity Optimization

The GNC method begins by introducing a surrogate function, along with a *control parameter*, to approximate the original non-convex problem in a convex manner. The goal is to find the minimum of this new convex. Subsequently, a series of new surrogate functions are generated by progressively adjusting the *control parameter* and solving for the minimum, with each iteration starting from the previous solution [149, 150]. Throughout the process, GNC continuously refines the approximation function, solving for the minimum of the newest surrogate function at each step. In this process, the approximation function gradually converges towards the original non-convex function, and the resulting minimum increasingly approaches the global optimum.

The GNC scheme is first proposed by Blake and Zisserman [149] and they construct a polynomial surrogate function to approximate TLS problem, as illustrated

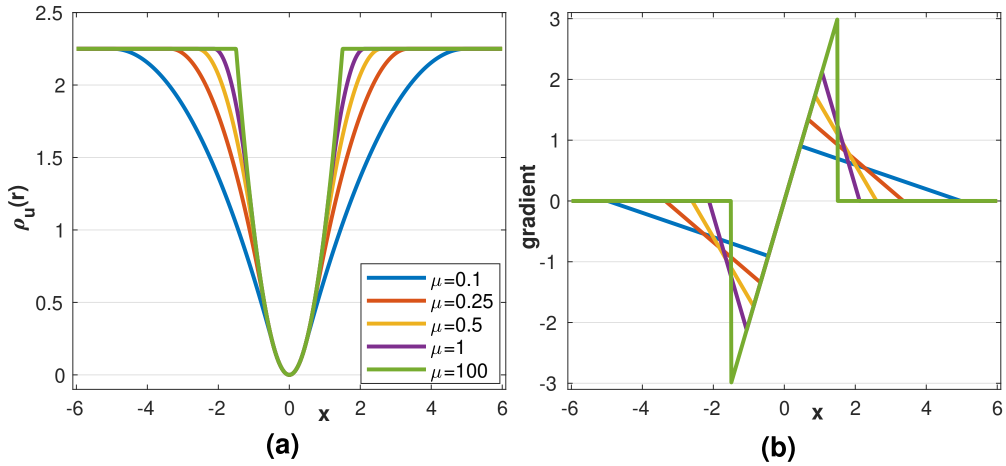


Figure 5.3: (a) The cost curve of ρ_μ with different μ value. (b) The corresponding gradient with residual.

by the following formulation:

$$\rho_\mu(r) = \begin{cases} r^2 & |r| \in \left(0, \sqrt{\frac{\mu}{1+\mu}}\Theta\right] \\ 2\Theta|r|\sqrt{\mu(1+\mu)} - \mu(\Theta^2 + r^2) & |r| \in \left(\sqrt{\frac{\mu}{1+\mu}}\Theta, \sqrt{\frac{1+\mu}{\mu}}\Theta\right] \\ \Theta^2 & |r| \in \left(\sqrt{\frac{1+\mu}{\mu}}\Theta, +\infty\right) \end{cases} \tag{5.6}$$

where μ is the control parameter to adjust the approximation function during the intermediate stages of solving the non-convex problem in (5.6). Fig. 5.3 shows an example of GNC-TLS with different control parameters μ. As depicted, μ modulates both the shape and differentiability of the surrogate function $\rho_\mu(\cdot)$. When $\mu \to 0$, the cost becomes smoother and more convex-like. As $\mu \to \infty$, the surrogate function $\rho_\mu(\cdot)$ gradually approaches the original truncated quadratic. In practice, the initial value of μ is typically set to a constant near 0 for convexity approximation. As the iteration progresses, μ is incrementally scaled up, allowing $\rho_\mu(\cdot)$ to approach the true $\rho(\cdot)$ gradually.

To recover the outlier rejection process from $\rho_\mu(\cdot)$ function and unify it with robust estimation, the GNC-TLS function presented in (5.6) can be equivalently reformulated as a new objective function. This function introduces an additional variable $0 \leqslant w \leqslant 1$ and a penalty function $\Psi(w)$, which is uniformly represented as follows [38],

$$\min_{\substack{\boldsymbol{x}\in\mathcal{X} \\ w_i\in[0,1]}} \sum_{i\in\mathcal{M}} [w_i \|\boldsymbol{r}(\boldsymbol{z}_i, \boldsymbol{x})\|^2 + \Psi(w_i)], \tag{5.7}$$

the variable w is equal to the weight of measurements. The conversion process is known as *Black-Rangarajan Duality* [99]. For the GNC-TLS framework described in (5.6), the associated penalty function is defined as [99],

$$\Psi(w_i) = \frac{\mu(1-w_i)}{\mu + w_i}\Theta^2, \tag{5.8}$$

the detailed derivation can be found in the next Chapter 5.2.3. Fig. 5.4(a) illustrates the behavior of the penalty function $\Psi(w_i)$ for various values of μ. Subsequently, a new optimization problem is established that simultaneously performs global state estimation and outlier rejection. This is formulated as follows,

$$\min_{\substack{\boldsymbol{x}\in\mathcal{X} \\ w_i\in[0,1]}} \sum_{i\in\mathcal{M}} [w_i \|\boldsymbol{r}(\boldsymbol{z}_i, \boldsymbol{x})\|^2 + \frac{\mu(1-w_i)}{\mu + w_i}\Theta^2]. \tag{5.9}$$

where measurements with $w = 0$ are identified as outliers, thereby facilitating robust state estimation.

5.2.3 The Penalty Function Derivation of GNC-TLS

The *Black-Rangarajan Duality* is a mechanism for recovering the outlier process from a robust $\rho(\cdot)$, so that the cost function can be transformed as [99],

$$\rho(\boldsymbol{r}(\boldsymbol{x})) \Longrightarrow w\boldsymbol{r}^2(\boldsymbol{x}) + \Psi(w). \tag{5.10}$$

The steps of *Black-Rangarajan Duality* are summarized as follows,

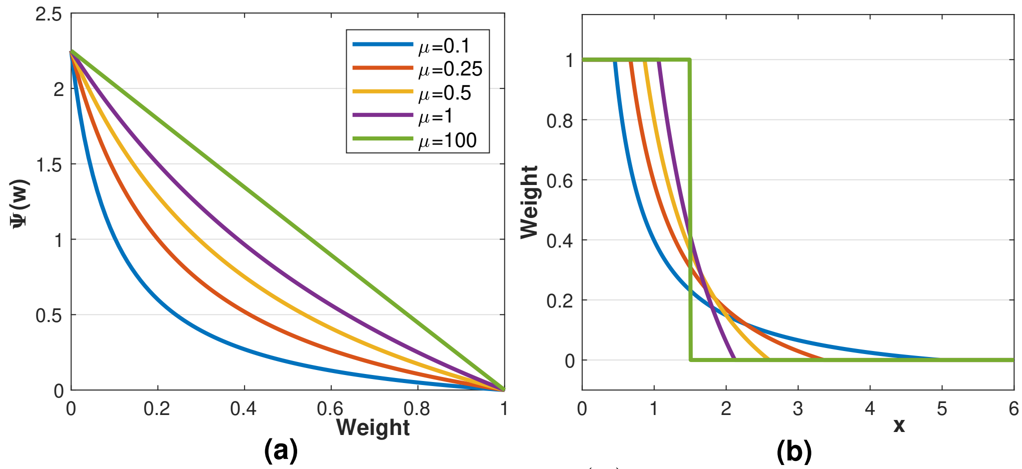


Figure 5.4: (a) the curve of penalty function $\Psi(w)$ in (5.8) with different μ. (b) the weight changes in (5.17) with residual under different μ. This figure reflects the influence of the control parameter μ in GNC towards the measurements' weight.

1. Define an auxiliary function $\phi(a) = \rho(\sqrt{a})$, and compute its first and second partial derivatives $\phi'(a)$, $\phi''(a)$ with respect to a.

2. If $\lim_{a\to 0} \phi'(a) = 1$, $\lim_{a\to\infty} \phi'(a) = 0$, and $\phi''(a) < 0$, the function $\phi(\cdot)$ can be derived.

3. Define outlier weight $w = \phi'(a)$, then $a = (\phi')^{-1}(w)$.

4. The penalty function $\Psi(w)$ is calculated as

$$\begin{aligned} \Psi(w) &= \phi(a) - wa \\ &= \phi((\phi')^{-1}(w)) - w(\phi')^{-1}(w). \end{aligned} \tag{5.11}$$

According to the GNC-TLS function in (5.6) and duality condition 1), the auxiliary function $\phi(\cdot)$, $\phi'(\cdot)$, and $\phi''(\cdot)$ are

$$\phi(r) = \begin{cases} r & |r| \in \left(0, \sqrt{\frac{\mu}{1+\mu}}\Theta\right] \\ 2\Theta\sqrt{r\mu(1+\mu)} - \mu(\Theta^2 + r) & |r| \in \left(\sqrt{\frac{\mu}{1+\mu}}\Theta, \sqrt{\frac{1+\mu}{\mu}}\Theta\right] \\ \Theta^2 & |r| \in \left(\sqrt{\frac{1+\mu}{\mu}}\Theta, +\infty\right) \end{cases}, \tag{5.12}$$

$$\phi'(r) = \begin{cases} 1 & |r| \in \left(0, \sqrt{\frac{\mu}{1+\mu}}\Theta\right] \\ \Theta\sqrt{\frac{\mu(1+\mu)}{r}} - \mu & |r| \in \left(\sqrt{\frac{\mu}{1+\mu}}\Theta, \sqrt{\frac{1+\mu}{\mu}}\Theta\right] \\ 0 & |r| \in \left(\sqrt{\frac{1+\mu}{\mu}}\Theta, +\infty\right) \end{cases}, \tag{5.13}$$

$$\phi''(r) = \begin{cases} 0 & |r| \in \left(0, \sqrt{\frac{\mu}{1+\mu}}\Theta\right] \\ -\frac{\Theta}{2}\sqrt{\frac{\mu(1+\mu)}{r^3}} & |r| \in \left(\sqrt{\frac{\mu}{1+\mu}}\Theta, \sqrt{\frac{1+\mu}{\mu}}\Theta\right] \\ 0 & |r| \in \left(\sqrt{\frac{1+\mu}{\mu}}\Theta, +\infty\right) \end{cases}. \tag{5.14}$$

From (5.13) and (5.14), it can be seen that the condition 2) of *Black-Rangarajan Duality* is satisfied.

Then, we focus on the middle section of $\phi(\cdot)$, since this phase includes information about the GNC approximation of the $\rho(\cdot)$. Based on the condition 3) $w = \phi'(\cdot)$, we get

$$r = (\phi')^{-1}(w) = \frac{\mu(1+\mu)}{(\mu+w)^2}\Theta^2, \tag{5.15}$$

substituting (5.12) and (5.15) into (5.11), the formula of penalty function is shown as

$$\begin{aligned} \Psi(w) &= 2\frac{\mu(1+\mu)}{\mu+w}\Theta^2 - \mu\Theta^2 - (\mu+w)r \\ &= \frac{\mu(1-w)}{\mu+w}\Theta^2 \end{aligned}, \tag{5.16}$$

which is consistent with (5.8). When the GNC control parameter $\mu \to \infty$ in (5.16), the $\Psi(w) = (1-w)\Theta^2$, and then the optimization function is the same with (5.5). This is consistent with the GNC property described in Chapter 5.2.2, i.e. as $\mu \to \infty$, GNC-TLS gradually returns to the original quadratic LS.

5.2.4 The Solver of GNC-TLS

To tackle the optimization problem in (5.9), we employ an alternating optimization strategy, focusing first on the state variables and then on the observation weights.

Initially, to compute the weights, we treat the state variables $\boldsymbol{x}$ as constants. We then take the first-order partial derivative of cost function in (5.9) with respect to weight w_i and set it equal to 0 [38, 99]. This leads to a closed-form expression for weights updating, given as follows,

$$w_i = \begin{cases} 1 & |r_i| \in \left(0, \sqrt{\frac{\mu}{1+\mu}}\Theta\right] \\ \frac{\Theta}{|r_i|}\sqrt{\mu(1+\mu)} - \mu & |r_i| \in \left(\sqrt{\frac{\mu}{1+\mu}}\Theta, \sqrt{\frac{1+\mu}{\mu}}\Theta\right] \\ 0 & |r_i| \in \left(\sqrt{\frac{1+\mu}{\mu}}\Theta, +\infty\right) \end{cases}. \tag{5.17}$$

The choice of threshold Θ is discussed in the section 5.3.2. Fig. 5.4 (b) shows how measurement weights change with residual for different μ.

Algorithm 1: The Global State Estimator with Outlier Rejection (GNC+TLS)

Input: Measurements $z_i, i \in \mathcal{M}$; Parameters : τ, P_{fa}, μ_0; GNC iteration times: N_{Iter}, WLS iteration times: N_{iter}.

Output: States $\boldsymbol{x}$; Jacobian matrix $\boldsymbol{J}$; Weight $\boldsymbol{w}$, μ.

▻ Initialization **begin**
- Threshold: $\Theta = \sqrt{chi2inv(1-P_{fa},k)} \cdot \tau$ by (5.29);
- Initial state: $s = 0$, $\boldsymbol{t} = 0_{3\times 1}$, $\boldsymbol{R} = \boldsymbol{I}_{3\times 3}$, $\boldsymbol{w}_0 = \boldsymbol{I}$;
- Computing $\boldsymbol{r}_0$ by (5.18).

▻ Optimization **for** $m < N_{Iter}$ **do**
- Weight update: $\boldsymbol{w}_m$ by (5.17).
- ▻ Gauss-Newton method **for** $n < N_{iter}$ **do**
  - State update: $\Delta\boldsymbol{x}$ by (5.25)
  - **if** $\Delta\boldsymbol{x} < 1e^{-4}$ **then**
    - **break**
  - $\boldsymbol{x}_n \leftarrow \boldsymbol{x}_{n-1} + \Delta\boldsymbol{x}$
- $\boldsymbol{x}_m \leftarrow \boldsymbol{x}_{m-1} + \boldsymbol{x}_n$
- $\mu_m \leftarrow 1.4 * \mu_{m-1}$
- **if** *IsBinary($\boldsymbol{w}_m$)* **then**
  - **break**

return state: $\boldsymbol{x}_m$, Jacobian: $\boldsymbol{J}$, Weight $\boldsymbol{w}_m$, μ_m

Next, when calculating the states, we consider the weights as constants and omit the penalty term from (5.9) because it is independent of the states. The current optimization problem degenerates into a classical weighted least squares (WLS) problem, which can be formulated as,

$$\min_{\substack{\boldsymbol{x}\in\mathcal{X} \\ w_i\in[0,1]}} \sum_{i\in\mathcal{M}} w_i \left\| \boldsymbol{r}(\boldsymbol{z}_i, \boldsymbol{x}) \right\|^2 . \tag{5.18}$$

This problem can be solved using the generalized optimization methods, such as the Gauss-Newton (GN) method. Although GN is classified as a local optimization approach, the GNC framework approximates the original problem as convex, enabling convergence gradually towards the global optimal without relying on an initial guess. Algorithm 1 summarizes the pseudo-code of the GNC-TLS method. In general, GNC serves to find high-quality regions of the search space for the original estimation problem. The essence of solving this nonlinear optimization problem is still rooted in the classical least squares algorithms.

5.3 State Error Bound Prediction

As mentioned above, we need to solve a weighted least squares (WLS) problem, as formulated in (5.18). In this section, we delve into the analysis of error propagation in Gauss-Newton (GN) optimization method under the assumption of Gaussian noise distribution, while also considering the impact of outliers present within the measurement set on state estimation. Based on this analysis, we aim to predict state error bounds to guarantee the safety and reliability of the system throughout the estimation phase.

5.3.1 Error Propagation in Gauss-Newton Method

For state estimation, the generalization of the observation model can be written as,

$$\boldsymbol{z}_i = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v}_i, \tag{5.19}$$

where $\boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{Q})$ is the measurement noise item, which follows a zero-mean Gaussian distribution. The corresponding least squares problem of (5.19) can be expressed as,

$$\min_{\boldsymbol{x} \in \mathcal{X}} \sum_{i \in \mathcal{M}} \|\boldsymbol{z}_i - \boldsymbol{h}(\boldsymbol{x})\|^2, \tag{5.20}$$

with the error function $\boldsymbol{r}$ is $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})$. When the observation error is caused only by stochastic measurement noise and the observations are mutually independent, we have $\boldsymbol{r} \sim \mathcal{N}(0, \boldsymbol{Q})$ based on (5.19) and $\boldsymbol{Q}$ is a real diagonal matrix, shown as

$$\boldsymbol{Q} = \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_m^2 \end{bmatrix}_{m \times m}, \tag{5.21}$$

where m is the number of measurements.

Taking GN method as an example, its solution scheme is to perform a first-order Taylor expansion on $\boldsymbol{f}(\boldsymbol{x})$, as shown below,

$$\boldsymbol{f}(\boldsymbol{x} + \Delta\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{J}_r \Delta\boldsymbol{x}, \tag{5.22}$$

where $\boldsymbol{J}_r$ is the derivative of $\boldsymbol{f}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$. The state increment update equation is

$$\boldsymbol{J}_r^\top \boldsymbol{J}_r \Delta\boldsymbol{x} = -\boldsymbol{J}_r \boldsymbol{f}(\boldsymbol{x}) \Rightarrow \Delta\boldsymbol{x} = (\boldsymbol{J}^\top \boldsymbol{J})^{-1} \boldsymbol{J}^\top \boldsymbol{f}(\boldsymbol{x}), \tag{5.23}$$

with $\boldsymbol{J}$ is the derivative of $\boldsymbol{h}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ and $-\boldsymbol{J}_r = \boldsymbol{J}$ (Jacobian matrix). Substituting (5.23) into (5.22), we have

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x} + \Delta\boldsymbol{x}) &= \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{J}(\boldsymbol{J}^\top \boldsymbol{J})^{-1} \boldsymbol{J}^\top \boldsymbol{f}(\boldsymbol{x}) \\ &= (\boldsymbol{I} - \boldsymbol{J}(\boldsymbol{J}^\top \boldsymbol{J})^{-1} \boldsymbol{J}^\top) \boldsymbol{f}(\boldsymbol{x}) \end{aligned}. \tag{5.24}$$

Formula (5.24) presents the residual propagation of GN method during optimization. Similarly, for the weighted GN method, we have

$$\Delta \boldsymbol{x} = (\boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{J})^{-1} \boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{f}(\boldsymbol{x}), \tag{5.25}$$

$$\boldsymbol{f}(\boldsymbol{x}+\Delta \boldsymbol{x}) = (\boldsymbol{I} - \boldsymbol{J}(\boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{J})^{-1} \boldsymbol{J}^\top \boldsymbol{W}) \boldsymbol{f}(\boldsymbol{x}), \tag{5.26}$$

where $\boldsymbol{W}$ is the weight matrix (or information matrix) and $\boldsymbol{W} = \boldsymbol{Q}^{-1}$.

5.3.2 The Setting of Threshold for Outlier Rejection

As known in the section 5.2, a threshold Θ is required to update the weight of each measurement and determine whether it is an outlier or not, which is shown in (5.17). In this section, we analyze the sources of system errors and discuss the selection strategy of the threshold Θ in detail. First, we acknowledge that system errors arise from two primary sources: stochastic measurement noise and deterministic measurement bias. The former is typically modeled as a zero-mean Gaussian distribution, the latter is treated as outliers within measurement set [151].

At the beginning, we assume that the system error is solely caused by random measurement noise, with no outliers present in the observation set. Under this condition, we can expand the cost function around the initial state using a Taylor series. By combining (5.18) and (5.26), the cost of the optimization problem can be expressed as

$$\begin{aligned} \sum_{i \in \mathcal{M}} w_i \left\| \boldsymbol{r}(\boldsymbol{z}_i, \boldsymbol{x}) \right\|^2 &= \boldsymbol{f}(\boldsymbol{x}+\Delta \boldsymbol{x})^\top \boldsymbol{W} \boldsymbol{f}(\boldsymbol{x}+\Delta \boldsymbol{x}) \\ &= \boldsymbol{r}^\top \boldsymbol{W} (\boldsymbol{I}-\boldsymbol{P})^2 \boldsymbol{r} \\ &= \boldsymbol{r}^\top \boldsymbol{W} (\boldsymbol{I}-\boldsymbol{P}) \boldsymbol{r} \end{aligned} , \tag{5.27}$$

where $\boldsymbol{P} = \boldsymbol{J}(\boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{J})^{-1} \boldsymbol{J}^\top \boldsymbol{W}$. $\boldsymbol{P}, \boldsymbol{I}-\boldsymbol{P}$ are idempotent matrices [130] and the cost in (5.27) is referred to as the weighted sum of squares errors (WSSE). Meanwhile, WSSE follows a central Chi-squared (χ^2) distribution with $m-n$ degree of freedoms (DoFs) under the mentioned assumption of Gaussian noise distribution and none of the observations are outliers, shown as

$$WSSE \sim \chi^2(m-n), \tag{5.28}$$

where m denotes the number of measurements, n is the DoFs of the state variables. The proof process has been derived in Chapter 3.3.3.

According to the definition of central χ^2 distribution with k DoFs, it represents the sum of the squares of k independent standard normal distributions. Thus, equation (5.28) is satisfied when observation errors contain only zero-mean Gaussian measurement noises. Therefore, when measurement biases (outliers) are present, the WSSE follows a noncentral χ^2 distribution [152]. For a given probability of false alarm P_{fa}, a threshold Td for the central χ^2 distribution with k DoF can be determined via a table look-up directly. This threshold can be used to determine whether the specified central χ^2 distribution is satisfied for the purpose of detecting outliers. For instance, if $WSSE < Td$, we hypothesize that that no outliers exist in the measurement set. If $WSSE > Td$, there is a probability of $1-P_{fa}$ that outliers are present in the measurements [118]. The threshold Td can be computed using MATLAB function $chi2inv(1-P_{fa},k)$.

This method of outlier detection, known as the classic Chi-square test, is also used in our previous work. However, its use of a greedy algorithm, which iteratively updates and compares the sum of squared residuals from all valid observations against the threshold Td to remove outliers one by one, is inefficient and lacks precision. To address these shortcomings, drawing inspiration from the Chi-square test, we propose an improved threshold-setting scheme within the framework of the GNC-TLS algorithm outlined in the previous section. The threshold Θ is defined as,

$$\Theta = \sqrt{chi2inv(1-P_{fa},k)} \cdot \tau, \tag{5.29}$$

where τ is an adjustable parameter that reflects the assumed percentage of residuals caused by outliers in the sum of residuals. This new method enables the weight updates for all measurements to be more flexible and efficient, improving both the speed and accuracy of outlier rejection.

5.3.3 Error Bound Calculation

As discussed in section 5.3.2, the sources of errors in state estimation are caused by measurement noises and measurement biases (outliers). Since the noise-induced error has been deduced in Chapter 3.3.4, which is shown in

$$\mathcal{E}_n = k\sqrt{[(\boldsymbol{J}^\top \boldsymbol{W}\boldsymbol{J})^{-1}]}, \tag{5.30}$$

we directly discuss the state estimation error induced by outliers in Gaussian Newton optimization methods in the following chapter.

Outlier-induced Error

Due to the absence of ground truth concerning outliers, we cannot confirm that the outlier rejection strategy will completely eliminate all outliers from the measurement set. Consequently, it is essential to consider the potential impact of outliers on the state estimation and incorporate this uncertainty into the state error bounds.

Based on the above assumption, we denote the additional residual caused by outliers as $\boldsymbol{f}$. The expansion of the error function in (5.22) can be rewritten as

$$\boldsymbol{f}(\boldsymbol{x}+\Delta\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{J}_r\Delta\boldsymbol{x} + \boldsymbol{f}, \tag{5.31}$$

so, the new state and residual update function using weighted GN method is expressed by

$$\Delta\boldsymbol{x} = (\boldsymbol{J}^\top \boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^\top \boldsymbol{W}(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{f}), \tag{5.32}$$

$$\boldsymbol{f}(\boldsymbol{x}+\Delta\boldsymbol{x}) = (\boldsymbol{I}-\boldsymbol{J}(\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^\top\boldsymbol{W})(\boldsymbol{f}(\boldsymbol{x})+\boldsymbol{f}). \tag{5.33}$$

According to (5.32), the state estimation bias induced by outliers can be expressed as

$$\boldsymbol{b} = (\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{f}. \tag{5.34}$$

However, there is still a problem to be solved, we cannot determine which measurements in the state estimation process are outliers. To address this, we introduce a matrix $\boldsymbol{A}_k$ comprised of binary values (0 and 1) to index potential outliers within the measurement set, where a value of 1 indicates the presence of a selected outlier. By changing the matrix $\boldsymbol{A}_k$, we can suppose the number and locations of outliers. Then, we have

$$\boldsymbol{f} = \boldsymbol{A}_k\boldsymbol{f}^*, \tag{5.35}$$

where $\boldsymbol{f}^*$ is the potential outlier-induced residual set. The potential state error caused by outliers in (5.34) is presented as

$$\boldsymbol{b}_k = (\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{A}_k\boldsymbol{f}^*. \tag{5.36}$$

Similar to (5.27), the WSSE induced by outliers is expressed as

$$\begin{aligned}\boldsymbol{\Gamma} &= \boldsymbol{f}^{*\top}\boldsymbol{A}_k^\top\boldsymbol{W}(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{A}_k\boldsymbol{f}^* \\ &= \boldsymbol{f}^{*\top}\boldsymbol{A}_k^\top\boldsymbol{S}\boldsymbol{A}_k\boldsymbol{f}^*\end{aligned}, \tag{5.37}$$

with

$$\boldsymbol{S} = \boldsymbol{W}(\boldsymbol{I}-\boldsymbol{J}(\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^\top\boldsymbol{W}). \tag{5.38}$$

On the other hand, to consider the impact of outliers on individual states, we similarly introduce a new index vector $\boldsymbol{H}_j$ composed of 0 and 1, for example,

$$x \Rightarrow \boldsymbol{H}_1 = [1,0,0,0,0,0], y \Rightarrow \boldsymbol{H}_2 = [0,1,0,0,0,0]. \tag{5.39}$$

Then, a specific state error vector caused by outliers is

$$\boldsymbol{b}_{jk} = \boldsymbol{H}_j(\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{A}_k\boldsymbol{f}^*, \tag{5.40}$$

the norm of the $\boldsymbol{b}_{jk}$ is used to express the scalar of the error, shown as

$$\begin{gathered}\|\boldsymbol{b}_{jk}\| = \sqrt{\boldsymbol{f}^{*\top}\boldsymbol{A}_k^\top\boldsymbol{D}_j\boldsymbol{A}_k\boldsymbol{f}^*} \\ \boldsymbol{D}_j = \boldsymbol{W}\boldsymbol{J}(\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-\top}\boldsymbol{H}_j^\top\boldsymbol{H}_j(\boldsymbol{J}^\top\boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^\top\boldsymbol{W}\end{gathered}. \tag{5.41}$$

For precautionary reasons, we expect to estimate the maximum of $\|\boldsymbol{b}_{jk}\|$ as the predicted error of states. Besides, since the outlier rejection module in the previous section has been passed, according to (5.17) and (5.37), the following constraint should be satisfied,

$$\boldsymbol{f}^{*\top}\boldsymbol{A}_k^\top\boldsymbol{S}\boldsymbol{A}_k\boldsymbol{f}^* \leqslant \frac{1+\mu}{\mu}\Theta^2, \tag{5.42}$$

where μ is the final control parameter when the state estimator (5.9) stops optimizing. As a result, we obtain an inequality constrained optimization problem as follows,

$$\begin{aligned}\max_{\boldsymbol{f}^*} \quad & \boldsymbol{f}^{*\top}\boldsymbol{A}_k^\top\boldsymbol{D}_j\boldsymbol{A}_k\boldsymbol{f}^* \\ s.t. \quad & \boldsymbol{f}^{*\top}\boldsymbol{A}_k^\top\boldsymbol{S}\boldsymbol{A}_k\boldsymbol{f}^* \leqslant \frac{1+\mu}{\mu}\Theta^2\end{aligned}. \tag{5.43}$$

It is worth noting that for problem (5.43), we do not need to solve $\boldsymbol{f}^*$, but only to obtain the maximum of the objective function. Through Lagrangian duality theory [41], this question can be solved by the following closed-form expression,

$$\max\|\boldsymbol{b}_{jk}\|^2 = \boldsymbol{\Lambda}_{\max}(\boldsymbol{A}_k, \boldsymbol{D}_j, \boldsymbol{S})\frac{1+\mu}{\mu}\Theta^2, \tag{5.44}$$

where $\boldsymbol{\Lambda}_{\max}(\boldsymbol{A}_k, \boldsymbol{D}_j, \boldsymbol{S})$ is the largest eigenvalue of matrix:

$$(\boldsymbol{A}_k^\top\boldsymbol{S}\boldsymbol{A}_k)^{-1}\boldsymbol{A}_k^\top\boldsymbol{D}_j\boldsymbol{A}_k. \tag{5.45}$$

The detailed derivation is shown in the next Chapter 5.3.4.

Besides, relying on the variation of matrix $\boldsymbol{A}_k$ and (5.44), we can obtain a set of potential state errors caused by measurement biases under different outlier assumptions. Next, we employ an enumeration method to find the maximum state error from this error set, which we ultimately designate as the error bound resulting from outliers. This can be expressed as

$$\mathcal{E}_{out} = \max_{\boldsymbol{A}_k \in \mathcal{A}} \sqrt{\frac{1+\mu}{\mu} \boldsymbol{\Lambda}_{\max}(\boldsymbol{A}_k, \boldsymbol{D}_j, \boldsymbol{S})} \Theta. \tag{5.46}$$

So far, we have obtained predicted state errors caused by measurement noises and outliers through (5.30) and (5.46), respectively. The final error bound of a specified state can be presented as the sum of both, as shown

$$\begin{aligned} \mathcal{E}_{bound} &= \mathcal{E}_n + \mathcal{E}_{out} \\ &= k\sqrt{[(\boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{J})^{-1}]_{j,j}} + \max_{\boldsymbol{A}_k \in \mathcal{A}} \sqrt{\frac{1+\mu}{\mu} \boldsymbol{\Lambda}_{\max}(\boldsymbol{A}_k, \boldsymbol{D}_j, \boldsymbol{S})} \Theta \end{aligned}. \tag{5.47}$$

The pseudo code of error bound calculation can be seen in algorithm 2, where the input Jacobian $\boldsymbol{J}$, Weight $\boldsymbol{w}$ and μ are derived from the output of the GNC-TLS algorithm 1.

5.3.4 The Derivation of the Closed-form Expression of the Outlier-induced Error

The inequality constrained optimization problem in 5.43 is shown as,

$$\begin{aligned} \max_{\boldsymbol{f}^*} \quad & \boldsymbol{f}^{*\top} \boldsymbol{A}_k^\top \boldsymbol{D}_j \boldsymbol{A}_k \boldsymbol{f}^* \\ s.t. \quad & \boldsymbol{f}^{*\top} \boldsymbol{A}_k^\top \boldsymbol{S} \boldsymbol{A}_k \boldsymbol{f}^* \leqslant \frac{1+\mu}{\mu} \Theta^2 \end{aligned}, \tag{5.48}$$

which can be rewritten as

$$\begin{aligned} \min_{\boldsymbol{f}^*} \quad & -\boldsymbol{f}^{*\top} \boldsymbol{A} \boldsymbol{f}^* \\ s.t. \quad & \boldsymbol{f}^{*\top} \boldsymbol{B} \boldsymbol{f}^* - \frac{1+\mu}{\mu} \Theta^2 \leqslant 0 \end{aligned}, \tag{5.49}$$

Algorithm 2: State Error Bound Calculation

Input: Jacobian $\boldsymbol{J}$; Weight $\boldsymbol{w}$; Covariance $\boldsymbol{Q}$; Parameters: P_{fa}, Θ, μ; Index vector $\boldsymbol{H}_j$; Number of measurements: M.

Output: State error bound $\mathcal{E}_{bound}$

begin

  $\boldsymbol{W} = \boldsymbol{w} \cdot \boldsymbol{Q}^{-1}$

  Noise-induced error: $\mathcal{E}_n = 3\sqrt{[(\boldsymbol{J}^\top \boldsymbol{W} \boldsymbol{J})^{-1}]_{j,j}}$

  Computing $\boldsymbol{S}$, $\boldsymbol{D}_j$ by (5.38) and (5.41)

  for $k < M$ **do**

    Outlier-induced error:

    $\mathcal{E}_{out} = \max\limits_{\boldsymbol{A}_k \in \mathcal{A}} \sqrt{\frac{1+\mu}{\mu} \boldsymbol{\Lambda}_{\max}(\boldsymbol{A}_k, \boldsymbol{D}_j, \boldsymbol{S})} \Theta$

  $\mathcal{E}_{bound} = \mathcal{E}_n + \mathcal{E}_{out}$

return error bound: $\mathcal{E}_{bound}$

with $\boldsymbol{A} = \boldsymbol{A}_k^\top \boldsymbol{D}_j \boldsymbol{A}_k$, $\boldsymbol{B} = \boldsymbol{A}_k^\top \boldsymbol{S} \boldsymbol{A}_k$. By introducing the Lagrange multiplier λ, we have the new Lagrangian function as shown

$$\boldsymbol{L}(\boldsymbol{f}^*, \lambda) = -\boldsymbol{f}^{*\top} \boldsymbol{A} \boldsymbol{f}^* + \lambda \left(\boldsymbol{f}^{*\top} \boldsymbol{B} \boldsymbol{f}^* - \frac{1+\mu}{\mu} \Theta^2 \right). \tag{5.50}$$

Then, the original problem (5.48) is converted into an equivalent unconstrained optimization problem, shown as

$$\min_{\boldsymbol{f}^*, \lambda} \boldsymbol{L}(\boldsymbol{f}^*, \lambda), \tag{5.51}$$

with the listed Karush-Kuhn-Tucker (KKT)[9] conditions:

$$\begin{aligned}
&(1) \quad \nabla_{\boldsymbol{f}^*} \boldsymbol{L} = -\boldsymbol{A} \boldsymbol{f}^* + \lambda \boldsymbol{B} \boldsymbol{f}^* = 0 \\
&(2) \quad \boldsymbol{f}^{*\top} \boldsymbol{B} \boldsymbol{f}^* - \frac{1+\mu}{\mu} \Theta^2 \leqslant 0 \\
&(3) \quad \lambda \geqslant 0 \\
&(4) \quad \lambda \left(\boldsymbol{f}^{*\top} \boldsymbol{B} \boldsymbol{f}^* - \frac{1+\mu}{\mu} \Theta^2 \right) = 0
\end{aligned} . \tag{5.52}$$

From condition (1), we have $\boldsymbol{B}^{-1} \boldsymbol{A} \boldsymbol{f}^* = \lambda \boldsymbol{f}^*$, i.e. the eigenvalues of matrix $\boldsymbol{B}^{-1} \boldsymbol{A}$ meet the requirement of extremes of (5.51), and $\lambda \in (\lambda_1, \lambda_2, \cdots, \lambda_n)$ is the eigenvalue set of matrix $\boldsymbol{B}^{-1}\boldsymbol{A}$. Since the matrix $\boldsymbol{B}^{-1}\boldsymbol{A}$ is nonnegative, we have $\lambda \geqslant 0$, the KKT condition (3) is satisfied. Also, when $\lambda = 0$, condition (4) is held. When $\lambda > 0$, we have

$$\boldsymbol{f}^{*\top}\boldsymbol{B}\boldsymbol{f}^{*} - \frac{1+\mu}{\mu}\Theta^2 = 0 \Rightarrow \boldsymbol{f}^{*\top}\boldsymbol{B}\boldsymbol{f}^{*} = \frac{1+\mu}{\mu}\Theta^2. \tag{5.53}$$

Substituting condition (1) and (5.53) into the original optimization function (5.48), we have

$$\max_{\boldsymbol{f}^*} \boldsymbol{f}^{*\top}\boldsymbol{A}\boldsymbol{f}^{*} = \max \lambda \boldsymbol{f}^{*\top}\boldsymbol{B}\boldsymbol{f}^{*} = \lambda_{max}\frac{1+\mu}{\mu}\Theta^2, \tag{5.54}$$

where λ_{max} is the maximum eigenvalue of matrix $\boldsymbol{B}^{-1}\boldsymbol{A}$. Thus, (5.44) is proved. Meanwhile, this problem can also be proved by Rayleigh Quotient properties [153].

5.4 Experiments & Discussion

In this section, we verify the performance of the proposed system using both simulation and real-world datasets. We compare the results of our method with others in terms of state estimation accuracy, outlier rejection, and error bound prediction. Regarding visual experiments, the accuracy comparison algorithms include the local Gauss-Newton (GN) optimization method [154]. For point cloud experiments, comparison methods are Go-ICP [85], FGR [92], and Teaser++ [44]. Outlier rejection methods for comparison are RANSEC and the Chi-squared test (χ^2-test). For error bound comparison, we compare our approach with the 3σ method and PURSE proposed by Yang [40]. All experiments are implemented on Ubuntu 20.04 and performed on a desktop PC equipped with an Inter-Core i9-12900K and 32 GB of memory.

5.4.1 Simulation Experiments

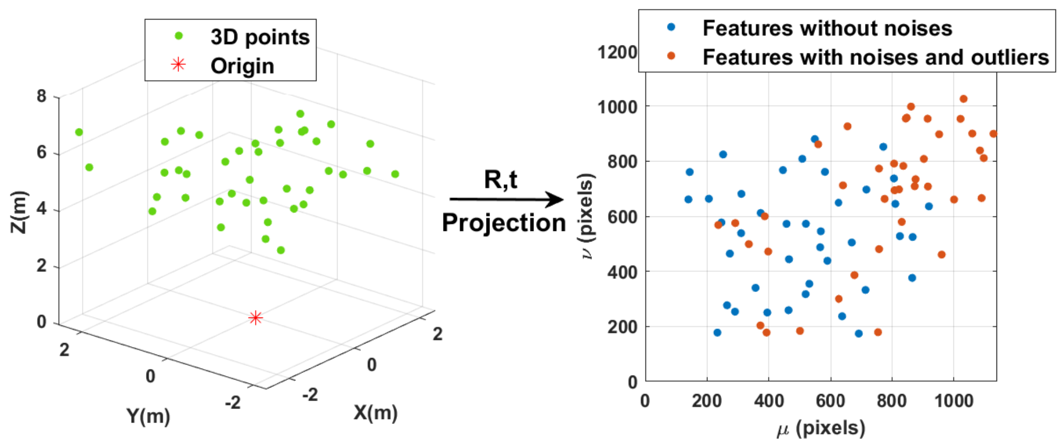


Figure 5.5: Demonstration of the synthetic dataset. The left figure shows simulated 3D spatial points, while the right figure depicts the projection of these points onto the pixel plane through a virtual camera model.

Setup

Taking classic visual state estimation as an example, the source of simulation dataset is shown below. First, we define a series of 3D spatial points as features and project them into the image pixel plane by a virtual camera model, resulting in noise-free correspondences of 2D pixel features and 3D spatial points [**?**]. Next, we apply a ground truth transformation to the 3D points and then introduce zero-mean Gaussian noise to the features to simulate measurement errors. Finally, we randomly select a percentage of features and impose a range of deviations to create outliers. The synthetic dataset is constructed as shown in Fig. 5.5, with detailed parameters available in the open-source implementation.

Accuracy Validation

We set up 40 measurements, with the ground truth transformation varying cyclically within the ranges x, y, z $\in [0, 0.3]$ meters and roll, pitch, and yaw angles $\in [0, 15]$ degree. Subsequently, we calculate the state estimation accuracies of multiple algorithms as the outlier rates increased from 0% (all inliners) to 70%. The

Table 5.1: RMSE of ATE (m) statistics on synthetic dataset

Outlier Rate	R-PnP	R-GN	χ^2-GN	GNCTLS-GN
0	0.0029	0.0028	0.0018	**0.0012**
10%	0.0041	0.0040	0.0037	**0.0033**
20%	0.0054	0.0054	0.0054	**0.0047**
30%	0.0067	0.0067	0.0069	**0.0057**
40%	0.0078	0.0072	0.1704	**0.0072**
50%	0.0095	0.0095	1.0948	**0.0089**
60%	0.0113	0.0114	3.3816	**0.0106**
70%	**0.0146**	0.0151	5.2704	0.2540

comparison methods include

1. GNCTLS-GN: is our proposed method, which applies the GNC-TLS for outlier rejection and Gauss-Newton method for state optimization.

2. R-GN/Ransac-GN: the classical visual state estimation method uses the RANSAC strategy [47] for outlier rejection and Gauss-Newton method for optimization with initial guesses given by Perspective-n-Point (PnP) method [155].

3. R-PnP/Ransac-PnP: a similar framework to R-GN but without Gauss-Newton optimization.

4. χ^2-GN: utilizes the mentioned χ^2-test for outlier rejection and PnP with Gauss-Newton for state estimation.

Among them, the four algorithms are implemented in MATLAB.

The root mean squared error (RMSE) of the absolute trajectory error (ATE) [148] under different outlier rates is presented in Table 5.1. It can be seen that our proposed method consistently outperforms the others until the outlier rate exceeds 70%. R-PnP and R-GN exhibit better accuracy and stability, highlighting the advantages of RANSAC. However, the time consumption of RANSAC grows

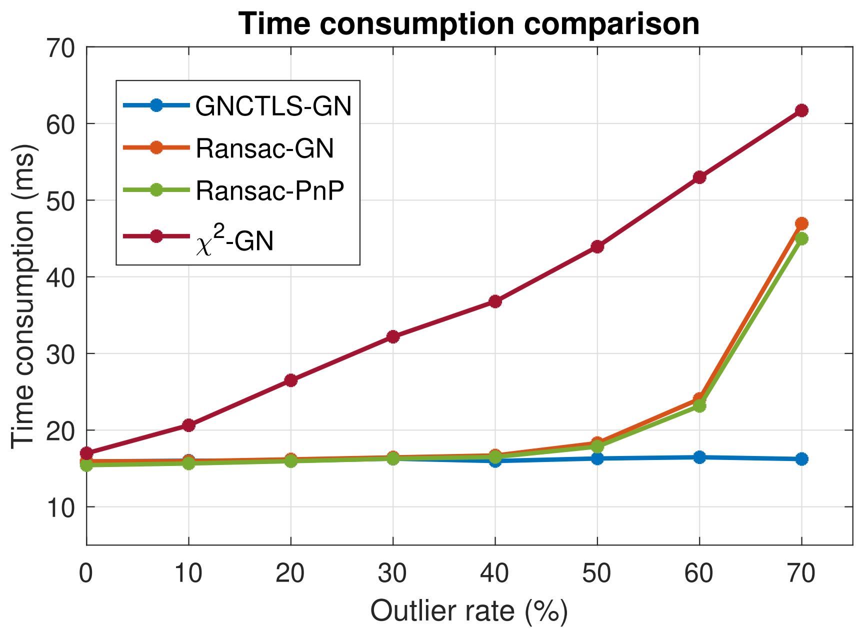


Figure 5.6: Comparison of the time consumption of various algorithms under different outlier ratios.

drastically as the percentage of outliers rises (see Fig. 5.6). Besides, the accuracy comparison of these two methods reveals that GN optimization relies on the initial guess provided by PnP for the non-convex transformation estimation problem. In contrast, our proposed method performs effectively without initialization, underscoring the efficacy of the global optimization of the GNC-TLS framework. Moreover, the accuracy of χ^2-GN declines gradually when the outlier rate reaches 40%, finally failing to converge. According to the principle of its algorithm, χ^2-test removes outliers sequentially during the optimization iteration, leading to increased time consumption as outlier rates increase, as shown in Fig. 5.6.

Meanwhile, Fig. 5.6 displays that the time consumption of our proposed method remains unaffected by the number of outliers. Fig. 5.7 and Fig. 5.8 provide a detailed comparison of the translation and rotation errors [156] of various methods, with the outlier rate changes from 0% to 50%. The results indicate that all algorithms demonstrate consistent estimation accuracy for both translation and ro-

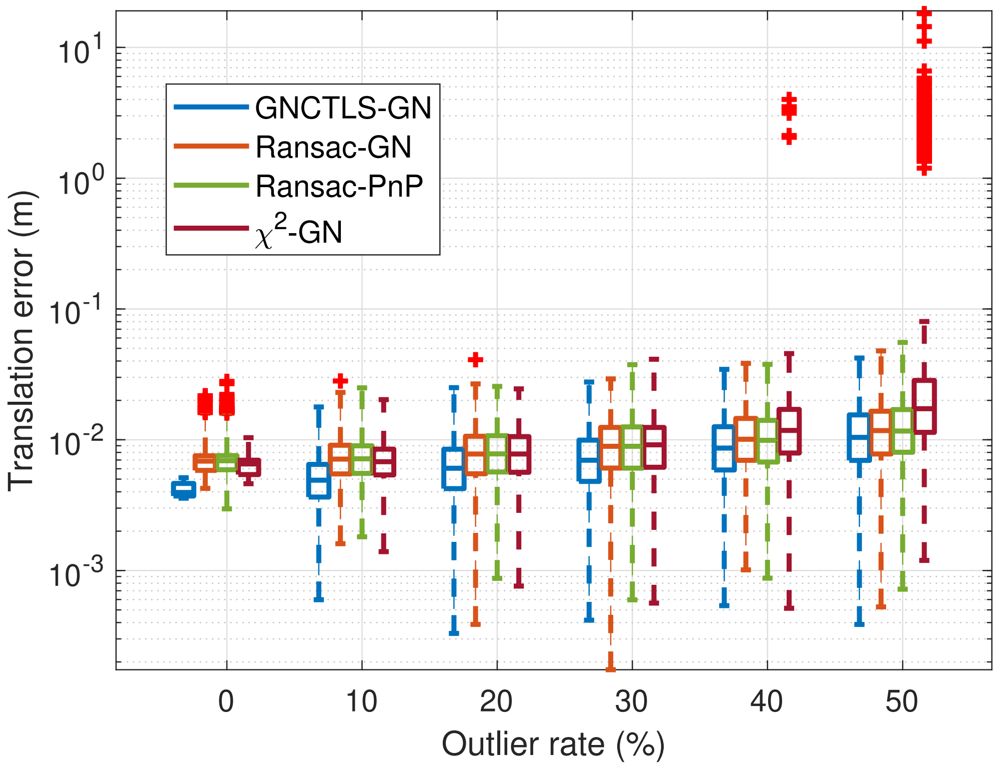


Figure 5.7: Position error comparison of simulation dataset under different outlier ratios.

tation, with our method being superior to the others. Therefore, considering the comprehensive performance, our method has a significant advantage in terms of high-precision, robustness, and real-time capability.

Error Bound Validation

In this part, we compare the state error prediction performance of our proposed error bound $\mathcal{E}_{bound}$ and classical 3σ under different state estimation frameworks with outlier rate changes from 10% to 60%. At the beginning, we assume that there is still one outlier remaining in the measurement set after outlier rejection, which is involved in state optimization and affects estimation results. However, it is impossible to determine which measurement is the outlier directly. At this point, we can adjust the form of the index matrix $\boldsymbol{A}_k$ in (5.35) to go through all situations,

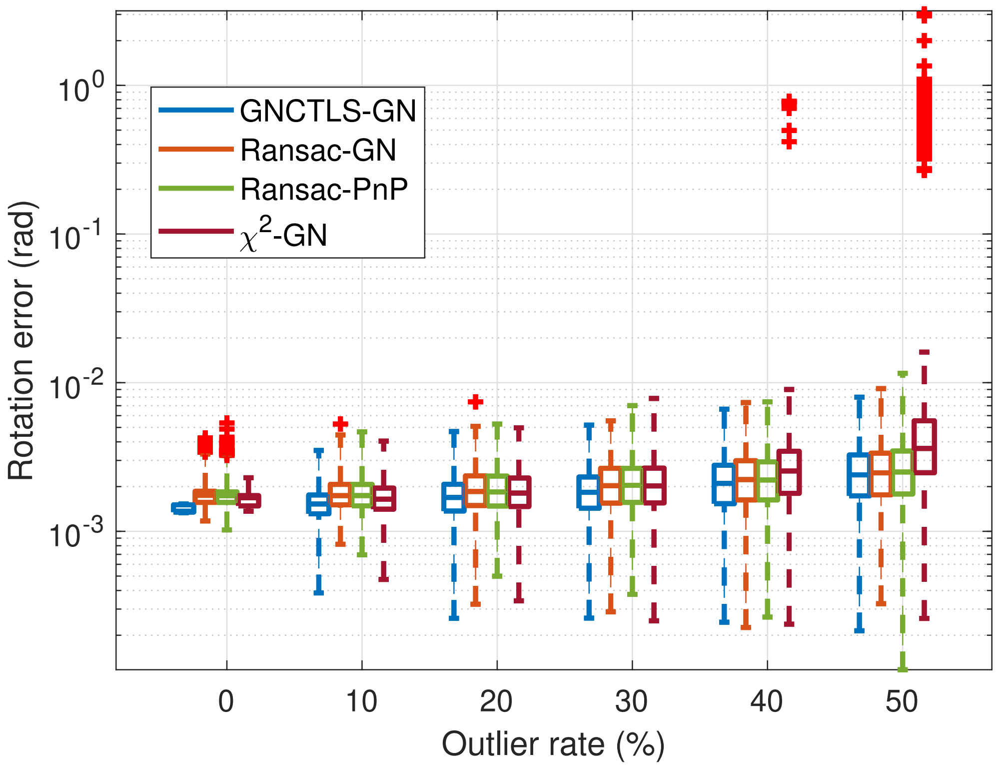


Figure 5.8: Rotation error comparison of simulation dataset under different outlier ratios.

such as

$$\boldsymbol{A}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}^{\top}_{2\times 2m}, \tag{5.55}$$

which implies that the first measurement is the outlier. Typically, we assess the validity of error bounds from two perspectives. First, the predicted error bound needs to be greater than the true error. Secondly predicted error is expected to be close to the error. For the first point, we use the bounded rate to evaluate the effectiveness of the $\mathcal{E}_{bound}$ and 3σ, which is expressed as

$$b_r = \frac{a}{b}, \tag{5.56}$$

where, a is the frame number of the predicted error $\geqslant$ the true error, b is the number of all frames.

Table 5.2: The b_r (%) of $\mathcal{E}_{bound}$ / 3σ on translation error

Outlier Rate	χ^2-GN	R-GN	GNCTLS-GN
10%	**99.30** / 66.95	**99.50** / 80.67	**100.0** / 97.13
20%	**97.45** / 56.90	**99.07** / 71.37	**99.97** / 90.43
30%	**93.85** / 50.25	**98.47** / 67.03	**99.47** / 83.07
40%	**84.75** / 39.75	**96.73** / 62.23	**98.53** / 76.97
50%	×	**93.83** / 59.30	**97.93** / 70.77
60%	×	**91.57** / 56.17	**92.43** / 60.27

Table 5.3: The b_r (%) of $\mathcal{E}_{bound}$ / 3σ on rotation error

Outlier Rate	χ^2-GN	R-GN	GNCTLS-GN
10%	**98.70** / 36.80	**99.37** / 57.73	**100.0** / 84.53
20%	**95.60** / 36.95	**99.00** / 55.77	**99.90** / 73.77
30%	**90.85** / 36.45	**97.97** / 54.53	**99.13** / 67.30
40%	**80.70** / 30.80	**96.00** / 51.83	**98.10** / 63.67
50%	×	**92.97** / 52.03	**97.50** / 59.57
60%	×	**90.67** / 51.13	**93.70** / 56.40

Table 5.2 and Table 5.3 display the bounded rate b_r of the proposed error bound $\mathcal{E}_{bound}$ and 3σ metric for translation and rotation directions, respectively, across different state estimation front-end methods. It is evident that our proposed method consistently outperforms 3σ in all cases. In principle, this improvement can be attributed to our method takes into account the effect of outliers on the state estimation accuracy in addition to the measurement noise. Furthermore, at a constant outlier rate, the choice of front-end state estimation methods affects the result of error bounds. From (3.44) and (5.47), the calculation of 3σ and $\mathcal{E}_{bound}$ is related to the Jacobian matrix. The Jacobian is defined as the first-order partial derivative of the error function with respect to the state variables. In non-convex problems, the Jacobian corresponding to both local and global minima of the objective function tends to be close to the zero matrix. However, the state errors associated with these local and global optima are distinct, which means that the convergence of the op-

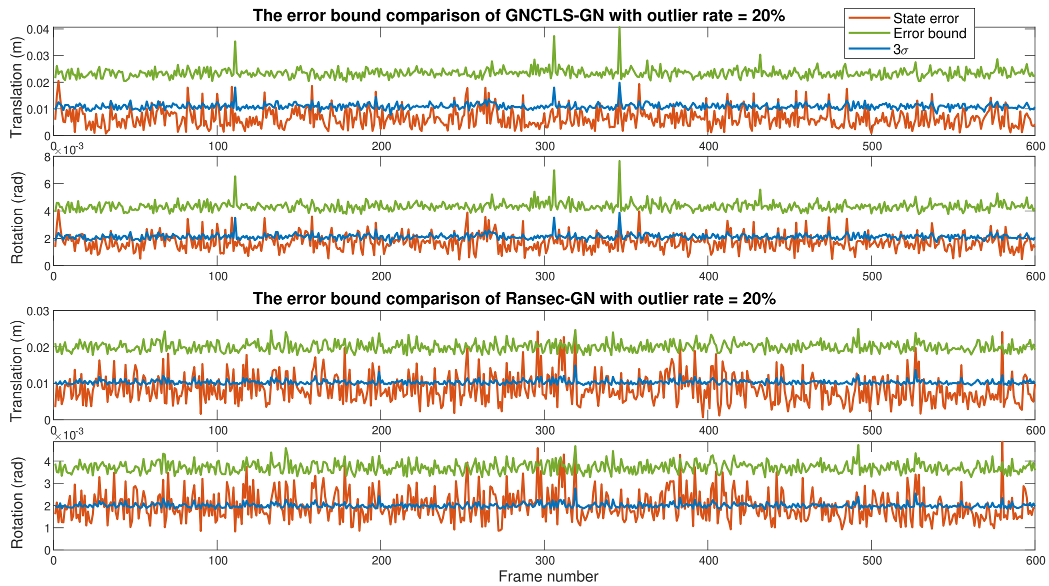


Figure 5.9: The $\mathcal{E}_{bound}$, 3σ and the state error comparison of GNCTLS-GN and Ransec-GN with outlier rate is 20%.

timization problem to the global optimum has a direct impact on the prediction of state errors.

The results in Table 5.2 and Table 5.3 validate this conclusion that our proposed framework achieves global optimality, with its error bound prediction outperforming the other local optimization algorithms. Fig. 5.9 and Fig. 5.10 show the $\mathcal{E}_{bound}$, 3σ, and the state error curves of our method and Ransac-GN during translation and rotation displacements at outlier rates of 20% and 40%. These figures clearly demonstrate that our proposed framework has remarkable reliability in state error prediction. Meanwhile, it illustrates the high fitness of the global optimization and the proposed error bound calculation method.

On the other hand, as the outlier rate increases, the b_r decreases gradually. We consider the effect of outliers to be intensifying, so we assume the number of remaining outliers in the measurement set to be 2. For example, the matrix $\boldsymbol{A}_k$ in

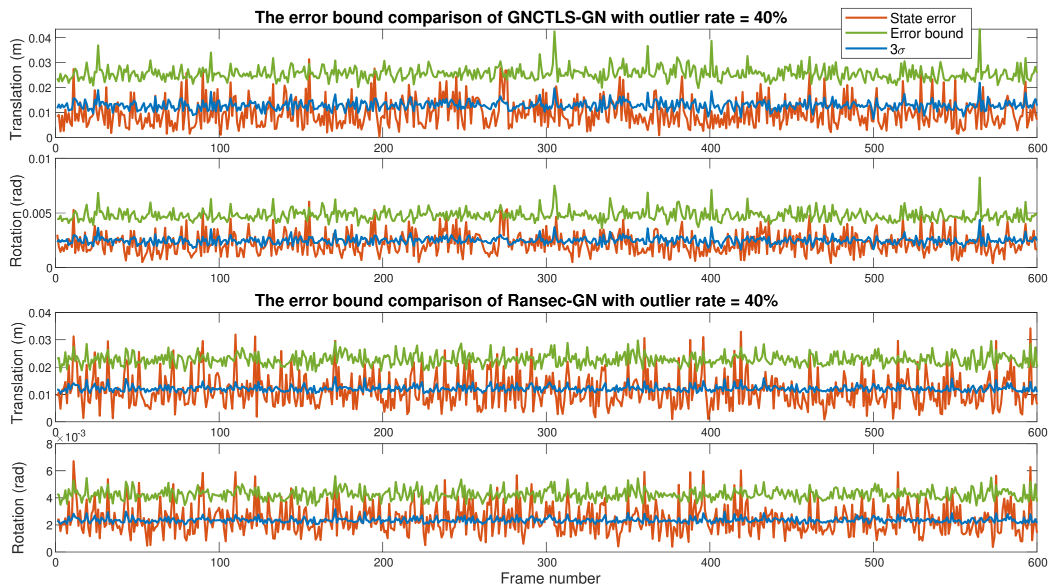


Figure 5.10: The $\mathcal{E}_{bound}$, 3σ and the state error comparison of GNCTLS-GN and Ransec-GN with outlier rate is 40%.

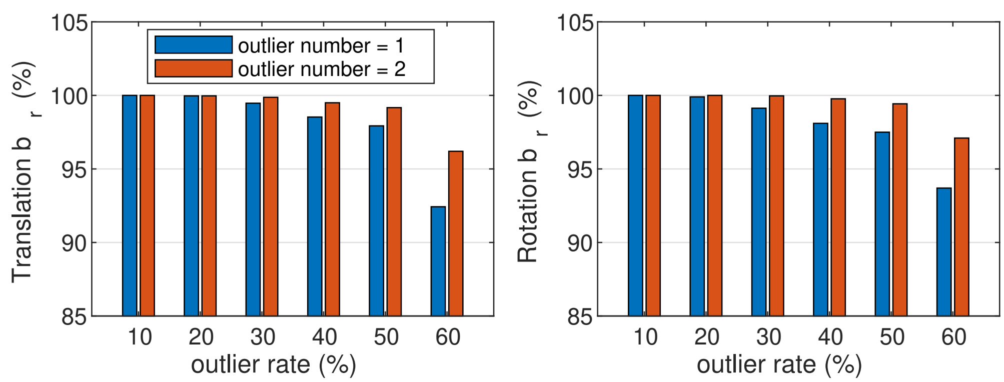


Figure 5.11: A comparison of the b_r varying with the increase in the number of outliers

(5.35) is

$$\boldsymbol{A}_k = \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 0 \end{bmatrix}_{2\times 2m}^{\top}, \tag{5.57}$$

which means that the first and second measurements are outliers. Taking the proposed GNCTLS-GN as an example, Fig. 5.11 shows that the b_r is boosted as the number of assumed outliers increases. This is because this hypothesis expands the

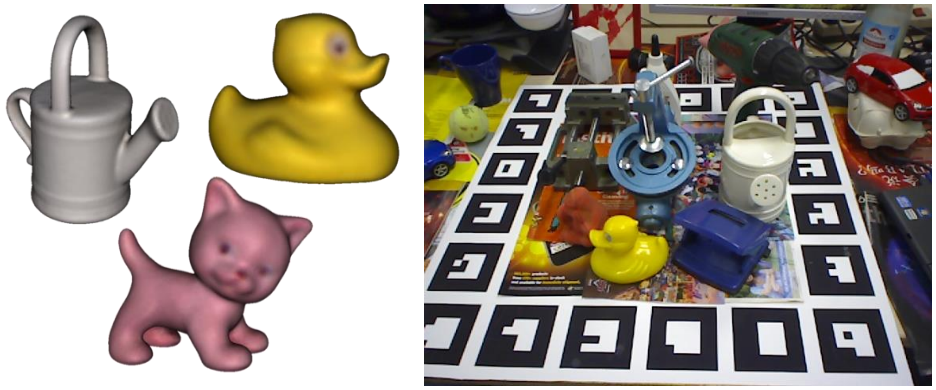


Figure 5.12: The benchmark of 6D pose estimation dataset. The left shows the 3D models of some targets, the right is one of the dataset sampling scenarios.

$\mathcal{E}_{out}$ part in (5.47). However, the time consumption of $\mathcal{E}_{bound}$ also surges. Therefore, a trade-off is necessary in practical applications.

5.4.2 Real-world Experiments

In real-world experiments, we employ public point cloud and object pose estimation datasets to verify the effectiveness of our proposed framework.

Camera Pose Estimation

Table 5.4: RMSE of ATE (m) statistics on BOP dataset

Corres Number	PURSE	R-PnP	R-GN	χ^2-GN	GNCTLS-GN
20	0.0925	0.0436	0.0423	0.0465	**0.0279**
40	0.0656	0.0376	0.0377	0.0410	**0.0138**
All	×	0.0385	0.0385	0.0410	**0.0091**

All: around 60 in total.

The benchmark of 6D pose estimation (BOP) serves as a deep learning dataset within the computer vision (CV) domain, designed to estimate the pose of objects relative to the camera for each frame. It includes a set of images (pixel: 640×480), multiple object models (3D point cloud with mesh information) and corresponding

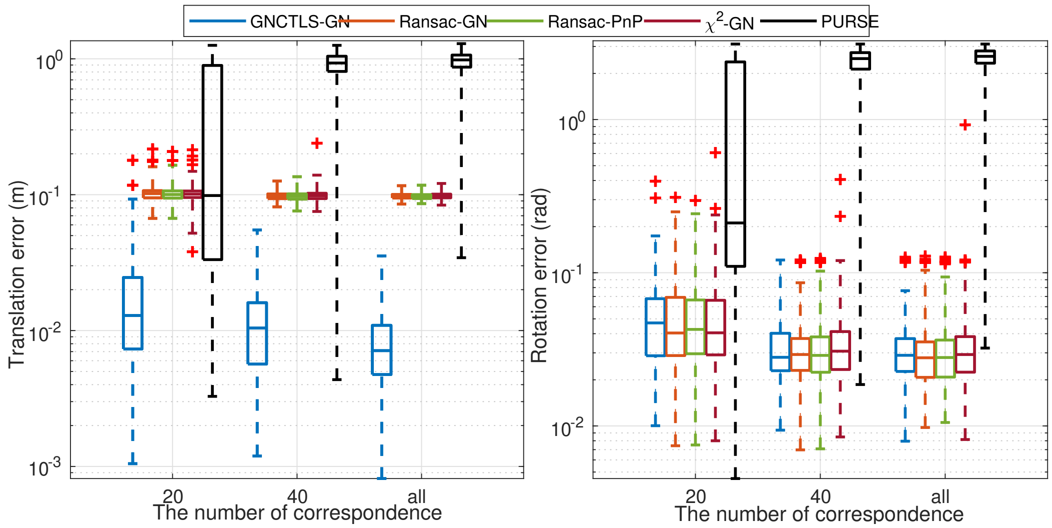


Figure 5.13: The translation and rotation error comparison of BOP dataset with different numbers of correspondences.

ground-truth data (shown in the Fig. 5.12). We leverage an existing deep learning network [157] to train the 2D feature and 3D point correspondences present in each frame, using these pairs for camera pose estimation. The comparison methods include GNCTLS-GN (ours), R-GN/Ransac-GN, R-PnP/Ransac-GN, χ^2-GN, and PURSE, where PURSE is a PnP-based random sample averaging estimator and computes the worst-case error bounds via semidefinite relaxation [40].

We calculate the state estimation results using varying numbers of correspondences, with accuracy measured by the RMSE of ATE, as displayed in Table 5.4. It can be seen that our method is better than the others across all scenarios. Meanwhile, accuracy improves progressively with an increasing number of observations. However, the success rate of PURSE declines significantly, ultimately failing when utilizing all correspondences. Additionally, Fig .5.13 illustrates the comparison of translation and rotation estimation errors, which reflect that our method excels particularly in translation estimation.

For error bound evaluation, we compare the b_r of the proposed $\mathcal{E}_{bound}$ and 3σ under Ransac-GN, χ^2-GN, and GNCTLS-GN methods, alongside the b_r of PURSE.

Table 5.5: The b_r (%) comparison of translation error on BOP dataset

Corres Number	PURSE	R-GN	χ^2-GN	GNCTLS-GN
20	66.50	**97.49** / 81.91	**97.49** / 82.91	**98.24** / 90.00
40	53.50	**82.32** / 33.33	**77.78** / 32.83	**98.04** / 77.12
All	$\times$	**67.68** / 22.73	**72.00** / 18.50	**97.35** / 81.46

Table 5.6: The b_r (%) comparison of rotation error on BOP dataset

Corres Number	PURSE	R-GN	χ^2-GN	GNCTLS-GN
20	64.80	**92.46** / 76.88	**92.46** / 77.89	**94.71** / 79.41
40	40.31	**93.43** / 70.71	**93.94** / 67.68	**92.16** / 56.21
All	$\times$	**91.92** / 63.13	**91.00** / 61.50	**90.13** / 33.77

Specifically, we assume that one outlier escapes rejection when computing $\mathcal{E}_{bound}$. The translation and rotation error bounded rates are shown in Table 5.5 and Table 5.6, respectively. Fig .5.14 demonstrates the state error prediction results of $\mathcal{E}_{bound}$, 3σ, and PURSE for both translation and rotation motions. It is clear that b_r for $\mathcal{E}_{bound}$ surpasses both 3σ and PURSE. The specific error curve presented in Fig .5.14 reveals that PURSE provides a significantly large error bound, which is ineffective for predicting state errors. In contrast, $\mathcal{E}_{bound}$ offers a bound closely aligned with the actual state error and follows a similar trend with the error, which is essential for safety-critical systems. Moreover, due to the global optimization required for semidefinite relaxation, the error bound computation of PURSE is extremely time-consuming. The average computational cost of PURSE with 20 measurements is 26.47s (using MATLAB), which does not meet real-time requirements. By contrast, the computation of $\mathcal{E}_{bound}$ is swift. The average computation times are 0.66ms, 2.34ms, and 8.16ms for 1, 2, and 3 remaining outliers assumptions with 20 measurements, respectively.

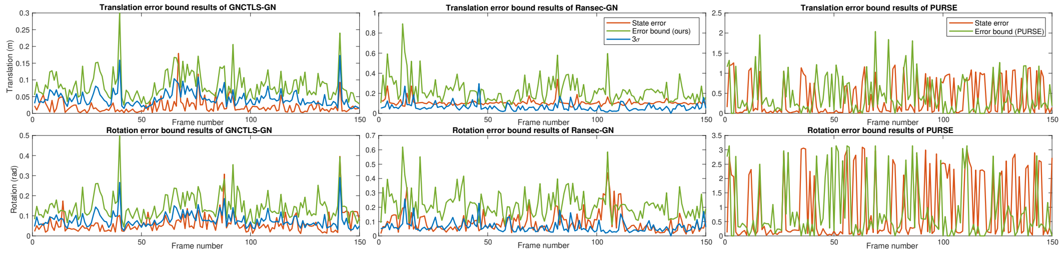


Figure 5.14: The $\mathcal{E}_{bound}$, 3σ, and the state error comparison of GNCTLS-GN and PURSE method with correspondence number is 20.

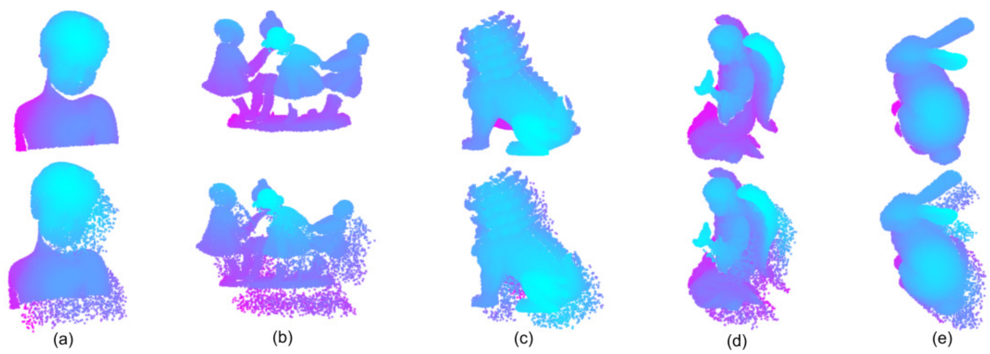


Figure 5.15: Display of five point clouds. The top row shows clean point clouds without noise. Point clouds in the bottom row have added zero-mean Gaussian noise as well as outliers.

Point Cloud Registration

In this part, we apply the point cloud datasets from literature [92, 44] for testing, randomly selecting points in cloud to add random offsets as outliers (0.05-0.1 times the range of the point cloud). The original and noise-contained point clouds are displayed in Fig .5.15. We compare the point cloud registration results of several global optimization methods for outlier percentages of 20%, 40%, and 60%. These include Go-ICP [85], TEASER++ [44], FGR [92], and the proposed method (GNCTLS-GN). Among that, TEASER++ uses GNC to estimate rotation subproblems and Douglas-Rachford Splitting (DRS) to certify global optimality [44]. The translation accuracy of point cloud registration is characterized by RMSE of ATE. The criteria for rotation error is adopted from the method in [156].

Table 5.7: RMSE of ATE (m) statistics on point cloud dataset

Outlier Rate	Go-ICP	TEASER++	FGR	GNCTLS-GN
20%	0.1597	0.0879	0.0690	**0.0630**
40%	0.1646	0.0900	**0.0752**	0.0961
60%	0.1694	0.2464	0.2383	**0.1623**

Table 5.8: Rotation error (rad) statistics on point cloud dataset

Outlier Rate	Go-ICP	TEASER++	FGR	GNCTLS-GN
20%	0.0626	0.0723	0.0621	**0.0333**
40%	0.0743	0.0935	0.0805	**0.0549**
60%	**0.0782**	0.1849	0.1792	0.1179

Statistics are shown in Table 5.7 and Table 5.8. The Go-ICP demonstrates stable accuracy under varying proportions of outliers, attributable to its inherent ability to consistently find the global optimal solution. However, this comes at the cost of significant variability in computational time, with execution times ranging from milliseconds to hours in certain scenarios. Meanwhile, due to the presence of outliers, the optimal solution to the estimation problem may not equal the ground truth [158]. The accuracy of TEASER++, FGR, and our algorithm decreases as the proportion of outliers increases, with our method showing only a marginal improvement over the others. Additionally, the computation times for the three methods are approximately at the millisecond level. This can be attributed to the fact that the underlying approach to state estimation in all three algorithms is fundamentally similar, as they all utilize the GNC convergence strategy. Specifically, the main distinction between GT and TEASER++ lies in the choice of the optimizer solver, while the difference from FGR stems from the variation in the loss function. The TLS method we employed demonstrates slightly better performance in outlier removal compared to the GM function used by FGR, which is consistent with the findings reported in [101].

Figure 5.16: The $\mathcal{E}_{bound}$, 3σ, and the state error comparison of GNCTLS-GN method with outlier rates are 20%, 40%, and 60%.

Under the assumption of one outlier remained, the comparison results of $\mathcal{E}_{bound}$, 3σ, and state estimation error of point clouds registration are presented in Fig .5.16. The figure indicates that the $\mathcal{E}_{bound}$ has a highly consistent fluctuation trend with state errors. However, the effect of the $\mathcal{E}_{bound}$ gradually degrades as the percentage of outliers increases, which means that the single outlier assumption no longer satisfies the demand. and more remaining outliers need to be set. Meanwhile, the results show that the proposed $\mathcal{E}_{bound}$ method has consistent effectiveness on the point cloud registration and visual pose estimation problem, which verifies the generality of this method.

5.5 Summary

In this work, we propose a comprehensive spatial transformation framework that effectively addresses the challenges of pose estimation in the presence of outliers and measurement errors. By integrating the TLS-GNC approach for outlier rejection and global optimization, our method significantly improves estimation accuracy while maintaining real-time performance, even in scenarios with high outlier rates. Furthermore, we introduce a dynamic thresholding mechanism based on the Gaussian noise distribution to enhance outlier detection, and we derive the propagation of estimation errors within the framework to predict state estimation errors with over 90% error coverage. Our experimental validation on both simulated and real-world datasets demonstrates that the proposed method outperforms existing approaches in both accuracy and robustness. The contributions of this work, including the error prediction method and the optimization framework, offer practical insights and solutions for improving the reliability of state estimation in safety-critical robotics and autonomous systems. The code and datasets have been made publicly available, encouraging further research and practical applications in the field.

Additionally, the error bound we propose is a versatile error prediction method based on a least squares optimization framework. It is not limited to the GNC-TLS framework utilized in this work. Users can easily implement error bound predictions by making simple adaptive adjustments tailored to their specific optimization algorithms. We outline some potential applications for this error bound: 1) *Positioning safety alerts:* For example, a predefined safety threshold can be set, and when the predicted positioning error exceeds this threshold, an alert can be triggered, indicating a potential risk to the system's positioning accuracy. 2) *For planning and control phase:* The error bound derived from our theoretical analysis and experimental validation can serve as a more reliable alternative to previously used empirical upper bounds. This can provide a reference for planning and control phase, enabling more reasonable and effective safety threshold settings in autonomous robotic systems.

Chapter 6

Conclusions and Future Work

This thesis presents a comprehensive exploration of advanced localization and state estimation techniques for autonomous systems, focusing on safety-critical applications such as UGVs and UAVs. The research presented in this study addresses key challenges in visual localization, including accuracy, outlier rejection, and error quantification, by leveraging innovative methodologies across three distinct studies.

6.1 Conclusion of Researches

6.1.1 Safety-Quantifiable Visual Localization with Loosely Coupled Architecture

In Chapter 3, a loosely-coupled monocular visual localization framework is presented, which achieves safety quantification through line feature matching with 3D prior maps. The system's modular design consists of two main components: (1) an independent visual localization module that mitigates VIO drift by optimizing 2D-3D line correspondences using a foot-point-based approach, and (2) a safety quantification module that employs a Chi-squared test for outlier rejection and computes a GNSS-inspired protection level (PL). This loosely-coupled structure enhances computational efficiency and provides the first six-degree-of-freedom (six-DoF) error prediction, encompassing both position and orientation, under multiple-fault assumptions.

The effectiveness of the proposed method is validated through a series of experiments conducted with UAVs in indoor environments and UGVs in both real and simulated outdoor settings. The experimental results demonstrate that, on the UAV dataset in real environments, the proposed algorithm improves localization accuracy by 12% compared to VINS and by 35.4% compared to the benchmark. On the UGV dataset, the accuracy improvements are 5% and 26.9%, respectively. In terms of error bound prediction, the proposed safety quantification module achieves bounded rates exceeding 60% and 80% for localization errors on UAV and UGV data, respectively, after excluding the yaw angle direction. The undesirable performance in the yaw direction is attributed to the degradation and unobservability of the current localization system in this orientation.

6.1.2 Tightly-Coupled Map-Aided VIO with Observability Analysis Localization

In Chapter 4, a tightly coupled state estimation framework is introduced, integrating IMU measurements, visual point features, and cross-modal line features within a sliding window structure. This approach constructs a factor graph-based optimization model, enhancing stability and accuracy through multi-frame constraints. The observability analysis of the proposed framework shows that global translations (x, y, and z) are observable, with only the yaw rotation remaining unobservable.

The effectiveness of the proposed method is validated through experiments conducted in both simulated and real-world environments. The results demonstrate that the system achieves significant improvements in localization precision and robustness compared to traditional VIO systems. For the simulation data, our proposed tightly coupled method demonstrates an average improvement of 11.44% in positioning accuracy compared to the loosely coupled method, 26.42% compared to VINS, and 30.15% compared to the benchmark. On the Hong Kong Polytechnic University dataset, the proposed method outperforms the others, with accuracy improvements of 13.3%, 17.7%, and 40.3% over the loosely coupled framework, VINS, and the benchmark, respectively.

6.1.3 Globally Robust State Estimation with Error Prediction

In Chapter 5, a comprehensive spatial transformation framework designed to address pose estimation challenges in robotics and autonomous driving is presented. This framework integrates a global optimization algorithm with outlier rejection, specifically utilizing the Graduated Non-Convexity (GNC) strategy combined with Truncated Least Squares (TLS) to enhance both accuracy and robustness. By approximating the non-convex problem as convex, the method gradually converges to the global optimum, even in scenarios with high outlier rates. The framework also introduces the error propagation theory during the used GNC-TLS optimizer, which predicts state estimation uncertainties and provides robust error bounds. This approach ensures the reliability and safety of the estimation process, crucial for safety-critical applications.

Experimental validation conducted on both simulated and real-world datasets demonstrates that our algorithm significantly outperforms existing methods in terms of outlier rejection and estimation accuracy. The experimental results indicate that the optimization structure we employed is capable of excluding over 60% of outliers while maintaining estimation accuracy. Notably, the time consumption for outlier rejection does not significantly increase with a higher proportion of outliers. Additionally, the error prediction method achieves over 90% coverage, providing a reliable measure of state estimation errors.

6.2 Future Directions

While the presented works have made significant strides in advancing localization and state estimation techniques for autonomous robotic systems, they also highlight several areas where further research and development are needed. The promising results achieved in various experimental settings underscore the potential for these methodologies to be refined and expanded. As the field continues to evolve, addressing the limitations and exploring new avenues for improvement will be crucial. This sets the stage for future work, which aims to build upon the current findings and tackle the remaining challenges to enhance the robustness, scalability, and applicability of these systems in increasingly complex and dynamic environments.

6.2.1 Prior Map-aided Visual Localization

1. **Enhanced Utilization of Prior Map Information:** Future research can focus on leveraging more comprehensive information from prior maps to improve localization accuracy. This includes integrating semantic information and planar features, which can provide additional context and constraints for more precise localization.

2. **Cross-Modal Fusion of 2D and 3D Data:** The accurate registration of initial positions is crucial for subsequent localization precision. Developing a robust initial position registration scheme will be essential for enhancing the accuracy of cross-modal data fusion, thereby improving the overall localization performance.

3. **Handling Large-Scale Prior Maps:** As the coverage area of prior maps expands, the volume of data that needs to be preloaded for real-time localiza-

tion becomes a challenge. Future work should explore engineering solutions to manage large datasets efficiently, such as utilizing GPU acceleration and cloud-based map updates, to ensure the system remains responsive and scalable.

6.2.2 Globally Robust State Estimation

1. **Incorporation of Diverse Constraints:** The current global state estimator primarily relies on single-type constraints, such as visual features and point cloud data. Future research could explore the development of adaptive sensor fusion strategies that dynamically adjust the weighting of different sensor inputs based on environmental conditions and sensor reliability. This approach would enhance the robustness of state estimation by allowing the system to prioritize the most reliable data sources in real-time, thereby improving performance in diverse and challenging environments.

2. **Analysis and Improvement of Global Optimization Strategies:** The current global optimization approach employs a heuristic graduated non-convexity strategy, which can occasionally fail to converge to the global optimum due to non-convexity issues. Future work could focus on conducting a detailed analysis of these failure cases to understand the underlying causes. Additionally, exploring alternative global optimization strategies that are more robust to GNC, such as stochastic optimization methods or hybrid approaches combining convex relaxation techniques, could enhance the reliability and optimality of the state estimation process.

6.2.3 The Application of Error Bound Prediction

1. **Improving Accuracy and Stability:** By incorporating dynamic measurement error models and accounting for feature degradation phenomena, future work can enhance the accuracy and stability of error estimation, leading to more reliable predictions.

2. **Versatile Error Prediction Method:** The proposed error bound is a flexible prediction method based on a least squares optimization framework. It is not limited to the proposed frameworks used in this study. Researchers can implement error bound predictions by making adaptive adjustments tailored to their specific optimization algorithms, broadening their applicability.

3. **Broader Applicability in Robotic Systems:** A promising direction for future research is to explore the broader applicability of error bounds within robotic systems. By predefining localization accuracy thresholds and utilizing predicted errors, we can evaluate whether current localization results meet operational requirements, thereby enabling safer real-world robotic deployments.

4. **Extending Applications Beyond Localization:** The potential applications of error bounds extend beyond localization. Its error predictions could also be leveraged in mapping and planning processes, opening new avenues for exploration and enhancing the overall capabilities of autonomous systems.

Collectively, these works demonstrate a comprehensive approach to enhancing the accuracy, reliability, and safety of autonomous robotic systems. The innovative methodologies and experimental validations provide a solid foundation for future research, with potential applications extending beyond localization to include mapping, planning, and control in dynamic environments. The open-source implementations further contribute to the research community, encouraging continued exploration and development in this vital field.

6.3 Harnessing Artificial Intelligence for Advanced Robotic Systems: Challenges and Opportunities

In recent years, the rapid advancement of artificial intelligence (AI) has profoundly impacted various industries, presenting both challenges and opportunities for the traditional field of robotics. Learning-based networks are increasingly being employed by researchers to enhance various modules within robotic systems. Similarly, the application of AI is indispensable for the research presented in this work. Future research is expected to leverage advanced AI technologies, such as deep learning and reinforcement learning, to further enhance the intelligence of robotic systems.

Specifically, deep learning can be utilized to improve perception modules by enabling more accurate feature extraction and classification, thereby enhancing environmental understanding. Reinforcement learning can optimize decision-making and control strategies, allowing robots to autonomously learn and adapt in complex and dynamic environments. Additionally, predictive models powered by AI can improve the accuracy and robustness of state estimation, particularly when dealing with uncertainty and nonlinear problems.

By integrating AI technologies with traditional robotic methods, future robotic systems will achieve greater autonomy and efficiency, driving the application and development of robotics across more domains. This integration not only helps overcome current technical bottlenecks but also opens new research directions and application scenarios, laying a solid foundation for the future advancement of robotics technology.

Bibliography

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

[2] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.

[3] X. Zheng, W. Wen, and L.-T. Hsu, "Safety-quantifiable line feature-based monocular visual localization with 3d prior map," *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[4] W. Wen, X. Bai, Y. C. Kan, and L.-T. Hsu, "Tightly coupled gnss/ins integration via factor graph and aided by fish-eye camera," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10 651–10 662, 2019.

[5] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[6] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[7] C. Zheng, W. Xu, Z. Zou, T. Hua, C. Yuan, D. He, B. Zhou, Z. Liu, J. Lin, F. Zhu *et al.*, "Fast-livo2: Fast, direct lidar-inertial-visual odometry," *IEEE Transactions on Robotics*, 2024.

[8] P. B. Ober, "Integrity prediction and monitoring of navigation systems." 2004.

[9] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[10] J. E. Angus, "Raim with multiple faults," *Navigation*, vol. 53, no. 4, pp. 249–257, 2006.

[11] A. Broggi, M. Buzzoni, S. Debattisti, P. Grisleri, M. C. Laghi, P. Medici, and P. Versari, "Extensive tests of autonomous driving technologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1403–1415, 2013.

[12] W. Wen, Y. Zhou, G. Zhang, S. Fahandezh-Saadi, X. Bai, W. Zhan, M. Tomizuka, and L.-T. Hsu, "Urbanloco: A full sensor suite dataset for mapping and localization in urban scenes," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2310–2316.

[13] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu *et al.*, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm5954, 2022.

[14] X. Wen, Y. Wang, X. Zheng, K. Wang, C. Xu, and F. Gao, "Simultaneous time synchronization and mutual localization for multi-robot system," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2603–2609.

[15] P. K. Enge, "The global positioning system: Signals, measurements, and performance," *International Journal of Wireless Information Networks*, vol. 1, no. 2, pp. 83–105, 1994.

[16] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.

[17] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[18] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 834–849.

[19] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "Svo: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

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

[21] S. Cao, X. Lu, and S. Shen, "Gvins: Tightly coupled gnss–visual–inertial fusion for smooth and consistent state estimation," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2004–2021, 2022.

[22] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4666–4672.

[23] C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang, "Fast-livo: Fast and tightly-coupled sparse-direct lidar-inertial-visual odometry," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4003–4009.

[24] L. Liu, H. Li, and Y. Dai, "Efficient global 2d-3d matching for camera localization in a large-scale 3d map," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2372–2381.

[25] Y. Xu, V. John, S. Mita, H. Tehrani, K. Ishimaru, and S. Nishino, "3d point cloud map based vehicle localization using stereo camera," in *2017 IEEE intelligent vehicles symposium (IV)*. IEEE, 2017, pp. 487–492.

[26] K. Yabuuchi, D. R. Wong, T. Ishita, Y. Kitsukawa, and S. Kato, "Visual localization for autonomous driving using pre-built point cloud maps," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 913–919.

[27] Y. Lu, H. Ma, E. Smart, and H. Yu, "Real-time performance-focused localization techniques for autonomous vehicle: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6082–6100, 2021.

[28] C. Ye, H. Zhao, L. Ma, H. Jiang, H. Li, R. Wang, M. A. Chapman, J. M. Junior, and J. Li, "Robust lane extraction from mls point clouds towards hd maps especially in curve road," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1505–1518, 2020.

[29] X. Xia, N. P. Bhatt, A. Khajepour, and E. Hashemi, "Integrated inertial-lidar-based map matching localization for varying environments," *IEEE Transactions on Intelligent Vehicles*, 2023.

[30] M. Brown, D. Windridge, and J.-Y. Guillemaut, "A family of globally optimal branch-and-bound algorithms for 2d–3d correspondence-free registration," *Pattern Recognition*, vol. 93, pp. 36–54, 2019.

[31] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Pl-slam: Real-time monocular visual slam with points and lines," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 4503–4508.

[32] J. Lu, Z. Fang, Y. Gao, and J. Chen, "Line-based visual odometry using local gradient fitting," *Journal of Visual Communication and Image Representation*, vol. 77, p. 103071, 2021.

[33] H. Lim, J. Jeon, and H. Myung, "Uv-slam: Unconstrained line-based slam using vanishing points for structural mapping," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1518–1525, 2022.

[34] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "Pl-vio: Tightly-coupled monocular visual–inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, p. 1159, 2018.

[35] D. G. Kottas and S. I. Roumeliotis, "Efficient and consistent vision-aided inertial navigation using line observations," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1540–1547.

[36] H. Yu, W. Zhen, W. Yang, J. Zhang, and S. Scherer, "Monocular camera localization in prior lidar maps with 2d-3d line correspondences," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4588–4594.

[37] X. Zheng, W. Wen, and L.-T. Hsu, "Tightly-coupled visual/inertial/map integration with observability analysis for reliable localization of intelligent vehicles," *IEEE Transactions on Intelligent Vehicles*, 2024.

[38] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated nonconvexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, 2020.

[39] W. A. Fuller, *Measurement error models*. John Wiley & Sons, 2009.

[40] H. Yang and M. Pavone, "Object pose estimation with statistical guarantees: Conformal keypoint detection and geometric uncertainty propagation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8947–8958.

[41] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[42] J. Briales, L. Kneip, and J. Gonzalez-Jimenez, "A certifiably globally optimal solution to the non-minimal relative pose problem," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 145–154.

[43] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 95–125, 2019.

[44] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, 2020.

[45] P.-Y. Lajoie, S. Hu, G. Beltrame, and L. Carlone, "Modeling perceptual aliasing in slam via discrete–continuous graphical models," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1232–1239, 2019.

[46] P. Antonante, D. I. Spivak, and L. Carlone, "Monitoring and diagnosability of perception systems," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 168–175.

[47] P. H. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer vision and image understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[48] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[49] T. Walter and P. Enge, "Weighted raim for precision approach," in *PROCEEDINGS OF ION GPS*, vol. 8, no. 1. Institute of Navigation, 1995, pp. 1995–2004.

[50] C. Zhu, M. Meurer, and C. Günther, "Integrity of visual navigation—developments, challenges, and prospects," *NAVIGATION: Journal of the Institute of Navigation*, vol. 69, no. 2, 2022.

[51] X. Zheng, W. Wen, and L.-T. Hsu, "Tightly-coupled line feature-aided visual inertial localization within lightweight 3d prior map for intelligent vehicles," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 6019–6026.

[52] R. W. Wolcott and R. M. Eustice, "Visual localization within lidar maps for automated urban driving," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 176–183.

[53] A. D. Stewart and P. Newman, "Laps-localisation using appearance of prior structure: 6-dof monocular camera localisation using prior pointclouds," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2625–2632.

[54] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.

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

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

[57] X. Zuo, P. Geneva, Y. Yang, W. Ye, Y. Liu, and G. Huang, "Visual-inertial localization with prior lidar map constraints," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3394–3401, 2019.

[58] Y. Kim, J. Jeong, and A. Kim, "Stereo camera localization in 3d lidar maps," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[59] T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard, "Monocular camera localization in 3d lidar maps," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1926–1931.

[60] L. Meng, "The "here and now" of hd mapping for connected autonomous driving," in *New Thinking in GIScience*. Springer, 2022, pp. 329–340.

[61] M. Schreiber, C. Knöppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 449–454.

[62] H. Ye, H. Huang, and M. Liu, "Monocular direct sparse localization in a prior 3d surfel map," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8892–8898.

[63] H. Huang, H. Ye, Y. Sun, and M. Liu, "Gmmloc: Structure consistent visual localization with gaussian mixture models," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5043–5050, 2020.

[64] S. Liang, Y. Zhang, R. Tian, D. Zhu, L. Yang, and Z. Cao, "Semloc: Accurate and robust visual localization with semantic and structural constraints from prior maps," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4135–4141.

[65] C. Zhang, H. Zhao, C. Wang, X. Tang, and M. Yang, "Cross-modal monocular localization in prior lidar maps utilizing semantic consistency," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4004–4010.

[66] T. Qin, Y. Zheng, T. Chen, Y. Chen, and Q. Su, "A light-weight semantic map for visual localization towards autonomous driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 248–11 254.

[67] Y. Zhou, X. Li, S. Li, and X. Wang, "Visual mapping and localization system based on compact instance-level road markings with spatial uncertainty," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 802–10 809, 2022.

[68] A. M. Andrew, "Multiple view geometry in computer vision," *Kybernetes*, 2001.

[69] Z. Kukelova, M. Bujnak, and T. Pajdla, "Automatic generator of minimal problem solvers," in *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part III 10*. Springer, 2008, pp. 302–315.

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

[71] G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based consistent ekf estimators for multi-robot cooperative localization," *Autonomous Robots*, vol. 30, pp. 99–122, 2011.

[72] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended kalman filtering for robot state estimation," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 402–430, 2020.

[73] X. Gao, T. Zhang, Y. Liu, and Q. Yan, "14 lectures on visual slam: from theory to practice," *Publishing House of Electronics Industry, Beijing*, 2017.

[74] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[75] A. R. Conn, N. I. Gould, and P. L. Toint, *Trust region methods*. SIAM, 2000.

[76] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[77] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 3607–3613.

[78] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," *Georgia Institute of Technology, Tech. Rep*, vol. 2, p. 4, 2012.

[79] S. Agarwal and K. Mierle, "Ceres solver: Tutorial & reference," *Google Inc*, vol. 2, no. 72, p. 8, 2012.

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

[81] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[82] 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.

[83] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[84] L. Carlone and F. Dellaert, "Duality-based verification techniques for 2d slam," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4589–4596.

[85] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-icp: A globally optimal solution to 3d icp point-set registration," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 11, pp. 2241–2254, 2015.

[86] A. Parra Bustos, T.-J. Chin, and D. Suter, "Fast rotation search with stereographic projections for 3d registration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3930–3937.

[87] A. P. Bustos and T.-J. Chin, "Guaranteed outlier removal for point cloud registration with correspondences," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2868–2882, 2017.

[88] L. Carlone, D. M. Rosen, G. Calafiore, J. J. Leonard, and F. Dellaert, "Lagrangian duality in 3d slam: Verification techniques and optimal solutions," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 125–132.

[89] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 545–565, 2016.

[90] T. Wu, G. Zaitian, Q. Wang, and F. Gao, "Scalable distance-based multi-agent relative state estimation via block multiconvex optimization," *arXiv preprint arXiv:2405.20883*, 2024.

[91] T. Wu and F. Gao, "A decoupled and linear framework for global outlier rejection over planar pose graph," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4150–4156.

[92] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European conference on computer vision*. Springer, 2016, pp. 766–782.

[93] T.-J. Chin and D. Suter, *The maximum consensus problem: recent algorithmic advances*. Springer Nature, 2022.

[94] H. Li, "Consensus set maximization with guaranteed global optimality for robust geometry estimation," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 1074–1080.

[95] T.-J. Chin, Y. Heng Kee, A. Eriksson, and F. Neumann, "Guaranteed outlier removal with mixed integer linear programs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5858–5866.

[96] A. Das and S. L. Waslander, "Outlier rejection for visual odometry using parity space methods," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3613–3618.

[97] P. J. Huber and E. M. Ronchetti, *Robust statistics*. John Wiley & Sons, 2011.

[98] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics: Methodology and distribution*. Springer, 1992, pp. 492–518.

[99] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International journal of computer vision*, vol. 19, no. 1, pp. 57–91, 1996.

[100] H. Yang and L. Carlone, "A polynomial-time solution for robust registration with extreme outlier rates," *arXiv preprint arXiv:1903.08588*, 2019.

[101] P. Antonante, V. Tzoumas, H. Yang, and L. Carlone, "Outlier-robust estimation: Hardness, minimally tuned algorithms, and applications," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 281–301, 2021.

[102] K. MacTavish and T. D. Barfoot, "At all costs: A comparison of robust cost functions for camera correspondence outliers," in *2015 12th conference on computer and robot vision*. IEEE, 2015, pp. 62–69.

[103] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Transactions on automatic control*, vol. 22, no. 5, pp. 728–740, 1977.

[104] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

[105] J. Hernandez, K. Tsotsos, and S. Soatto, "Observability, identifiability and sensitivity of vision-aided inertial navigation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2319–2325.

[106] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based rules for designing consistent ekf slam estimators," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 502–528, 2010.

[107] Y. Yang and G. Huang, "Observability analysis of aided ins with heterogeneous features of points, lines, and planes," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1399–1418, 2019.

[108] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[109] C. X. Guo and S. I. Roumeliotis, "Imu-rgbd camera 3d pose estimation and extrinsic calibration: Observability analysis and consistency improvement," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2935–2942.

[110] Y. Yang, P. Geneva, X. Zuo, and G. Huang, "Online self-calibration for visual-inertial navigation: Models, analysis, and degeneracy," *IEEE Transactions on Robotics*, 2023.

[111] A. Martinelli, "Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2011.

[112] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis, "Camera-imu-based localization: Observability analysis and consistency improvement," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 182–201, 2014.

[113] C. X. Guo and S. I. Roumeliotis, "Imu-rgbd camera navigation using point and plane features," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3164–3171.

[114] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 8, pp. 1207–1223, 2006.

[115] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[116] H. Yang and L. Carlone, "Certifiably optimal outlier-robust geometric perception: Semidefinite relaxations and scalable global optimization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 3, pp. 2816–2834, 2022.

[117] R. G. Brown, "A baseline gps raim scheme and a note on the equivalence of three raim methods," *Navigation*, vol. 39, no. 3, pp. 301–316, 1992.

[118] C. Li and S. L. Waslander, "Visual measurement integrity monitoring for uav localization," in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2019, pp. 22–29.

[119] O. A. Hafez, M. Joerger, and M. Spenko, "How safe is particle filtering-based localization for mobile robots? an integrity monitoring approach," *IEEE Transactions on Robotics*, 2024.

[120] G. Duenas Arana, O. Abdul Hafez, M. Joerger, and M. Spenko, "Integrity monitoring for kalman filter-based localization," *The International Journal of Robotics Research*, vol. 39, no. 13, pp. 1503–1524, 2020.

[121] O. A. Hafez, G. D. Arana, M. Joerger, and M. Spenko, "Quantifying robot localization safety: A new integrity monitoring method for fixed-lag smoothing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3182–3189, 2020.

[122] X. Lu, Y. Liu, and K. Li, "Fast 3d line segment detection from unorganized point cloud," *arXiv preprint arXiv:1901.02532*, 2019.

[123] A. Bartoli and P. Sturm, "Structure-from-motion using lines: Representation, triangulation, and bundle adjustment," *Computer vision and image understanding*, vol. 100, no. 3, pp. 416–441, 2005.

[124] G. Zhang, J. H. Lee, J. Lim, and I. H. Suh, "Building a 3-d line-based map using stereo slam," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1364–1377, 2015.

[125] J. H. Lee, S. Lee, G. Zhang, J. Lim, W. K. Chung, and I. H. Suh, "Outdoor place recognition in urban environments using straight lines," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5550–5557.

[126] V. S. Varadarajan, *Lie groups, Lie algebras, and their representations*. Springer Science & Business Media, 2013, vol. 102.

[127] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.

[128] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[129] F. Gustafsson, "Statistical signal processing approaches to fault detection," *Annual Reviews in Control*, vol. 31, no. 1, pp. 41–54, 2007.

[130] K. Hoffmann and R. A. Kunze, *Linear algebra*. Prentice-Hall New Jersey, 1971.

[131] E. B. Wilson and M. M. Hilferty, "The distribution of chi-square," *Proceedings of the National Academy of Sciences*, vol. 17, no. 12, pp. 684–688, 1931.

[132] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[133] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[134] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[135] K. Wu, A. M. Ahmed, G. A. Georgiou, and S. I. Roumeliotis, "A square root inverse filter for efficient vision-aided inertial navigation on mobile devices." in *Robotics: Science and Systems*, vol. 2. Rome, Italy, 2015.

[136] S. Yi, S. Worrall, and E. Nebot, "Metrics for the evaluation of localisation robustness," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1247–1253.

[137] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 809–816.

[138] E. W. Cheney and D. R. Kincaid, *Numerical mathematics and computing*. Cengage Learning, 2012.

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

[140] L. Zhang and T. Zhang, "A fast calibration method for dynamic lever-arm parameters for imus based on the backtracking scheme," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6946–6956, 2020.

[141] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 61–76, 2011.

[142] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Robotics: Science and Systems XI*, 2015.

[143] J.-Y. Bouguet *et al.*, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel corporation*, vol. 5, no. 1-10, p. 4, 2001.

[144] J. Svacha, J. Paulos, G. Loianno, and V. Kumar, "Imu-based inertia estimation for a quadrotor using newton-euler dynamics," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 3861–3867, 2020.

[145] K. Zhang, C. Jiang, J. Li, S. Yang, T. Ma, C. Xu, and F. Gao, "Dido: Deep inertial quadrotor dynamical odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9083–9090, 2022.

[146] P. Furgale, T. D. Barfoot, and G. Sibley, "Continuous-time batch estimation using temporal basis functions," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2088–2095.

[147] P. C. Hughes, *Spacecraft attitude dynamics*. Courier Corporation, 2012.

[148] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.

[149] A. Blake and A. Zisserman, *Visual reconstruction*. MIT press, 1987.

[150] A. Rangarajan and R. Chellappa, "Generalized graduated nonconvexity algorithm for maximum a posteriori image estimation," in *[1990] Proceedings. 10th International Conference on Pattern Recognition*, vol. 2. IEEE, 1990, pp. 127–133.

[151] J. Liu, M. Lu, Z. Feng, and J. Wang, "Gps raim: Statistics based improvement on the calculation of threshold and horizontal protection radius," in *International Symposium on GPS/GNSS*. Citeseer, 2005, pp. 8–10.

[152] B. W. Parkinson and P. Axelrad, "Autonomous gps integrity monitoring using the pseudorange residual," *Navigation*, vol. 35, no. 2, pp. 255–274, 1988.

[153] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[154] K. Schmeckpeper, P. R. Osteen, Y. Wang, G. Pavlakos, K. Chaney, W. Jordan, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Semantic keypoint-based pose estimation from single rgb frames," *arXiv preprint arXiv:2204.05864*, 2022.

[155] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 8, pp. 930–943, 2003.

[156] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8601–8610.

[157] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13*. Springer, 2014, pp. 536–551.

[158] L. Carlone *et al.*, "Estimation contracts for outlier-robust geometric perception," *Foundations and Trends® in Robotics*, vol. 11, no. 2-3, pp. 90–224, 2023.