UNIVERSITY
OF TECHNOLOGY
SYDNEY

Towards Robust Visual-Inertial SLAM

by Hongkyoon Byun

Thesis submitted in fulfilment of the requirements for
the degree of

Doctor of Philosophy

under the supervision of Prof. Shoudong Huang
and Dr. Liang Zhao

University of Technology Sydney
Faculty of Engineering and Information Technology

January 2024

Certificate of Original Authorship

I, HONGKYOON BYUN, declare that this thesis, is submitted in fulfilment of the requirements for the award of the degree of Doctor of Philosophy, in the School of Mechanical and Mechatronics Engineering, Faculty of Engineering and Information Technology (FEIT) at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature:

Production Note:
Signature removed prior to publication.

Date: 24 January 2024

Towards Robust Visual-Inertial SLAM

by

Hongkyoon Byun

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

Abstract

In recent decades, the development of autonomous navigation systems for mobile robots has been a key area of research. Among the solutions gaining prominence, the Monocular Visual-Inertial Navigation System (VINS) stands out for its compact size, cost-effectiveness, and robustness, addressing challenges in this domain.

Achieving optimal performance within resource constraints requires a delicate balance between computational efficiency and estimation accuracy. Choosing a Visual-Inertial SLAM (VI-SLAM) approach for VINS holds substantial significance, encompassing two primary categories: filtering-based methods and optimization-based methods. These methods offer versatile strategies tailored to specific application needs and resource constraints.

In this thesis, Compressed-MSCKF (Comp-MSCKF) is introduced as a filtering-based approach. This method effectively incorporates loop closure constraints for long-term navigation based on MSCKF. It achieves this by partitioning the extensive map into local and global maps, ensuring that the global map is updated whenever the local boundary changes. This approach leads to updates limited to $O\left(N_L^2\right)$, where N_L represents the size of the local map—typically smaller than the total number of states N.

To further enhance system accuracy and robustness, a novel optimization-based method called Parallax Visual-Inertial SLAM (PVI-SLAM) is then proposed. This approach leverages the parallax angle for feature parametrization, combining feature observations and preintegrated inertial measurement unit (IMU) data to formulate a nonlinear least squares problem. By doing so, it adeptly avoids singularity issues linked to problematic features, enabling PVI-SLAM to outperform VI-SLAM methods using XYZ parametrization. Incorporating Gaussian Process (GP)-based preintegration and using the observation ray as an objective function contribute to additional performance improvements. These enhancements not only address challenges posed by traditional methods but also elevate PVI-SLAM, bestowing it with superior robustness and accuracy.

However, the high-dimensional nonlinear optimization problem does not always ensure convergence, and even when it does, reaching the global minimum is not guaranteed. Additionally, it poses a significant computational burden, especially in large-scale scenarios with a very large number of poses. To tackle these challenges, a linear submap joining method using the Linear SLAM framework is proposed. In this approach, local submaps are constructed using the PVI-SLAM method, seamlessly joined through a combination of linear least squares and nonlinear coordinate transformations. This technique aims to enhance computational efficiency and overall system robustness, making it well-suited for challenging and resource-intensive scenarios.

A comprehensive series of quantitative analyses was conducted on a range of challenging datasets, validating the effectiveness of the proposed VI-SLAM algorithms.

Acknowledgements

I want to express my gratitude to Professor Shoudong Huang, my supervisor, for all the unwavering support and guidance provided throughout my journey. The opportunities you offered have not only contributed to my growth as a researcher but also as a better person. I truly wouldn't be where I am today without your encouraging mentorship.

I also want to extend my thanks to Dr. Liang Zhao, my co-supervisor, for your immense support, valuable advice, and genuine kindness. Whenever challenges arose, you devoted yourself to assisting and guiding me in the right direction. Your motivation played a pivotal role in sustaining my momentum in the research process.

Reflecting on my 3.5-year research journey, I appreciate the support from many individuals. Firstly, I wish to express my gratitude to Professor Jonghyuk Kim for granting me the opportunity to embark on this academic journey. I acknowledge the support provided by the Australian Research Council Discovery Project DP200101640.

Brenton Leighton, your friendship and guidance during the pandemic were invaluable. I vividly recall the day when Dr. Maleen Jayasuriya helped me set up the VINS-fusion and introduced me to everyone at our institute, the Robotic Institute; it was genuinely appreciated. To Sheila Sujipto, Dr. Yujun Lai, and Dr. Julian Collart, your support made my adaptation as a PhD student much smoother, and I am grateful for your friendship. A special thanks to Dr. Richardo Khonasty for standing by me during challenging times. A special thanks to everyone in our SLAM group who inspired and motivated me. The Brisbane conference trip with Tiancheng Li, Kai Pan, Yang Song, and Yang Xu was an enriching experience. To Dr. Raphael Falque, I regret not getting to know you sooner, but thank you for the invaluable advice, especially on our journey back home. Benny Dai, Rohan Patel, Dac Dang Khoa Nguyen, Dominik Slomma, and Dr. Cedric Le Gentil, your help and support meant a lot and brought me to this point.

I also want to express my gratitude to my family for their endless support. Thanks always to Soo Chung Han and Soo Young Han. Especially, Inha Yoon, your consistent empowerment in everything I do, right by my side, has been invaluable. Lastly, a big thank you to my parents, who have always believed in me.

Table of Contents

List of Figures

List of Tables

Acronyms & Abbreviations

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
AUVs	Autonomous Underwater Vehicles
BA	Bundle Adjustment
BRIEF	Binary Robust Independent Elementary Features
CP-SLAM	Compressed Pseudo-SLAM
Comp-MSCKF	Compressed-MSCKF
DBoW	Distributed Bag-of-Words
DoF	Degree of Freedom
DTAM	Dense Tracking and Mapping
DVP	Da Vinci Precinct
EKF	Extended Kalman Filter
FAST	Features from Accelerated Segment Test
FEJ-EKF	First Estimates Jacobian EKF
GCKF	Generalized Compressed Kalman Filter
GN	Gauss-Newton

GNSS Global Navigation Satellite System

GP Gaussian Process

GPS Global Positioning System

IDP Inverse Depth Parametrization

I-EKF Invariant EKF

IMU Inertial Measurement Units

LIDAR Light Detection And Ranging

LLS Linear Least Squares

LM Levenberg-Marquardt

MAV Micro Aerial Vehicle

MLE Maximum Likelihood Estimation

MSCKF Multi-State Constraint Kalman Filter

NLLS Nonlinear Least Squares

OC-EKF Observability Constrained EKF

ORB Oriented FAST and Rotated BRIEF

PBA Parallax Bundle Adjustment

PM Standard Preintegrated Measurement

PTAM Parallel Tracking and Mapping

PVI-SLAM Parallax Visual-Inertial SLAM

QUT Queensland University of Technology

RANSAC RANdom SAmple Consensus

RI-EKF Right Invariant Error EKF

RMSE	Root Mean Square Error
SBA	Standard Bundle Adjustment
SEIF	Sparse Extended Information Filters
SERF	Samford Ecological Research Facility
SIFT	Scale-Invariant Feature Transform
SKF	Schmidt Kalman Filter
SLAM	Simultaneous Localization And Mapping
SE(3)	Special Euclidean group in three dimensions
$\mathfrak{so}(3)$	Lie algebra of special orthogonal group in three dimensions
SO(3)	Special Orthogonal group in three dimensions
SONAR	Sound Navigation And Ranging
SURF	Speeded-Up Robust Features
SVO	Semi-direct Visual Odometry
SVs	Satellite Vehicles
UAV	Unmanned Aerial Vehicle
UGPM	Unified Gaussian Preintegrated Measurement
VINS	Visual-Inertial Navigation System
VI-SLAM	Visual-Inertial SLAM
VO	Visual Odometry
V-SLAM	Visual SLAM

Chapter 1

Introduction

1.1 Background

In the contemporary landscape, the significance of robotics has become increasingly undeniable, assuming a pivotal role in our daily lives. Notably, mobile robots have emerged as a burgeoning and versatile field of research, holding immense promise for advancing our society, both socially and economically. These robots, encompassing aerial, terrestrial, and underwater varieties, are being harnessed to replace humans across a spectrum of applications, including but not limited to service, surveillance, planetary exploration, patrolling, emergency rescue, and reconnaissance.

Mobile robots have gained widespread recognition due to their practicality, maneuverability, and agility, making them essential in both military and civilian operations for carrying out a wide variety of tasks. These robots typically receive predefined mission plans from ground control stations, relying on Global Positioning System (GPS) for accurate localization. However, challenges arise in environments with high levels of interference, potentially compromising the reliability of GPS, leading to localization inaccuracies. Such inaccuracies pose safety concerns during the execution of critical missions, highlighting the need for robust solutions in navigating complex and cluttered environments.

To overcome this challenge, there has been an increased research focus on the development of autonomous navigation systems over the past few decades. The primary objective of an autonomous navigation system is to enable robots to navigate independently in environments where they lack prior knowledge, determining the optimal path. To achieve this, the Simultaneous Localization And Mapping (SLAM) was introduced, allowing mobile robots to perceive unknown environments in real-time, construct maps, and simultaneously estimate their own positions based on sensor data.

Numerous SLAM systems have been developed, employing a variety of sensors, including cameras, Inertial Measurement Units (IMU), Light Detection And Ranging (LIDAR), Sound Navigation And Ranging (SONAR), and more. Recognizing the inherent limitations of individual sensors, such as the level of uncertainty in the observations, the field has embraced the adoption of multi-sensor fusion algorithms to enhance the overall accuracy and reliability of these systems.

In the context of addressing the challenges faced by mobile robots, the Visual-Inertial Navigation System (VINS) has emerged as a fundamental solution, capitalizing on the complementary nature of its components. The ability of the visual sensor to detect and track numerous features enriches the data with image-based information, which can then be used to improve the state estimation accuracy significantly. Simultaneously, the IMU plays a crucial role in bridging gaps and compensating for errors, particularly during instances where visual tracking encounters difficulties.

1.2 Visual-Inertial Navigation System

The central concept of VINS is sensor fusion, where data from both visual and inertial sensors are combined to provide accurate estimations of the device's motion and pose. The complementary nature of these two sensor types enhances the accuracy and robustness of the system.

1.2.1 Visual Sensor

Referring to the visual sensor, the VINS utilizes a camera for capturing image frames. In the process of estimating the motion or structure of the scene within VINS, information from these captured image frames can be extracted using two primary methods.

The first method is the direct approach, which bypasses the extraction of distinct features and instead directly utilizes entire pixel-intensity information from images. This approach, exemplified by methods like Dense Tracking and Mapping (DTAM) [15], leverages all available pixel data in each frame, thereby enhancing accuracy and robustness, particularly in featureless environments. However, processing all pixel intensities in each frame can be computationally expensive. To address this computational challenge, sparse mapping techniques have been developed. These techniques focus on selected sparse sets of pixels within the image frame, particularly those associated with high-gradient regions of the scene [16]. Additionally, the selection of specific frames within these high-gradient regions is implemented to further enhance computational efficiency [17].

The second method is the indirect (feature-based) approach, which initially extracts a set of feature observations from the image. Subsequently, it calculates the camera's position and scene geometry based solely on these extracted observations. These observations are typically derived from points that are readily recognizable or specific line and curve segments. This method utilizes well-known feature descriptors such as Harris [18], Speeded-Up Robust Features (SURF) [19], Scale-Invariant Feature Transform (SIFT) [20], Features from Accelerated Segment Test (FAST) [21], and Oriented FAST and Rotated BRIEF (ORB) [22]. The extracted features can be tracked through various techniques, including descriptor matching, filter-based tracking, optical flow tracking, and direct pixel processing [4, 23, 24].

1.2.2 Inertial Measurement Unit

The IMU, an integral component equipped with accelerometers and gyroscopes, serves as a valuable source of information, delivering crucial data pertaining to both angular rate and acceleration. Within this sensory system, gyroscopes play a pivotal role in precisely calculating the platform's attitude, providing insights into its orientation with respect to a given reference frame. Simultaneously, accelerometers contribute significantly to the estimation of the platform's position and velocity. Their function involves incorporating specific forces into their calculations, thereby offering a comprehensive understanding of the platform's dynamic state. Synthesizing the information from both accelerometers and gyroscopes yields a detailed and accurate 6 Degree of Freedom (DoF) description, encapsulating the platform's orientation and motion concerning the desired reference frame.

Nevertheless, in VINS, utilizing raw IMU measurements at high frequencies for each time step can impose a significant computational burden. Additionally, challenges arise due to the noise in sensor readings, which can adversely affect position and velocity estimate accuracy. To address this inherent issue, sophisticated integration techniques are employed. These integration methods [1, 8, 25] play a critical role in managing the growth of estimate errors. This helps in maintaining a reasonably high level of precision and reliability in the resulting orientation and motion estimations, all while minimizing computational complexity.

1.2.3 Visual-Inertial SLAM

Visual-Inertial SLAM (VI-SLAM) stands as a major advancement in navigation systems, marking a decisive step toward achieving better precision in VINS. The fundamental objective of this technology is to surmount the inherent challenges of SLAM by seamlessly integrating data from visual and inertial sensors.

Earlier methods within VI-SLAM focus on filtering-based approaches. These methods involve a continuous update of the system's location through the assimilation of incoming sensor data. Renowned for their real-time performance and efficiency, these filtering-based methods are particularly adept in scenarios demanding prompt updates. This attribute renders them highly suitable for applications characterized by dynamic environmental conditions or those requiring rapid adjustments to the navigation state [26].

On the other hand, the alternative approach within VI-SLAM leverages optimization-based methods. This approach embraces nonlinear optimization techniques to refine the estimation of the system's pose and map. Despite their computational demands, optimization-based methods stand out for achieving robust accuracy. Moreover, they offer the notable advantage of lower memory utilization, proving beneficial for applications requiring extended operational periods [26]. This careful trade-off between computational demands and enhanced accuracy positions optimization-based methods as valuable assets in scenarios where prolonged and reliable navigation is of paramount importance.

1.2.4 Initialization

Initialization in VI-SLAM is a pivotal phase in setting up the system and its sensors to provide an accurate starting point for the estimation of the camera or robot's initial pose and the initial map of the environment. Proper initialization is of paramount importance as it significantly influences the robustness and precision of the ensuing VI-SLAM operation.

The initialization process typically commences with the meticulous calibration of the system's sensors, with special emphasis on the cameras and the IMU. This calibration involves determining the exact relative positions and orientations of the sensors concerning one another, ensuring the precise fusion of data from both sensor types.

Subsequently, the system needs to estimate the initial pose of the camera or robot within the environment. This estimate is critical for providing a starting point for the SLAM system. Often, this involves employing techniques such as visual odometry or IMU integration. Visual odometry tracks visual features in the camera images over time, while IMU integration utilizes data from the IMU to estimate motion. A combination of both methods can yield a more accurate initial pose estimate.

The initialization process also entails selecting and tracking visual features in the camera images during the initial frames. These features are then used to initialize their Three-Dimensional (3D) positions in the map.

Scale estimation is another essential element of initialization. Since monocular cameras cannot directly estimate scale, the inclusion of IMU data is critical to resolve the scale ambiguity, ensuring that distances in the map are accurately represented.

The robustness of the initialization process is vital, and it should be capable of handling various conditions, including changes in lighting, dynamic scenes, and sensor noise. This robustness ensures that the system can effectively deal with challenging situations right from the outset.

1.2.5 Long-term Navigation

Loop-closure is another essential process in VI-SLAM, triggered when the system detects that the platform has returned to a previously visited location. This action is vital for enhancing overall map accuracy through a global optimization process. Loop-closure detection can be achieved through either odometry-based geometric relationships or appearance-based approaches. However, appearance-based methods, which assess the similarity between two different images, are often preferred over odometry-based techniques due to concerns about cumulative errors that can accumulate throughout the trajectory [27].

The loop-closure recognition process is essential, which can be done utilizing Distributed Bag-of-Words (DBoW) proposed by [28] to achieve a binary bag of words with Binary Robust Independent Elementary Features (BRIEF) and FAST features. To address the limitations of the BRIEF descriptor, which lacks rotation and scale invariance and is primarily suited for Two-Dimensional (2D) environments, [12] proposed a method based on DBoW and ORB that incorporates covisibility information. This innovation significantly enhances the system's ability to detect loop-closures in challenging environments.

1.3 Motivation

The deployment of small-scale systems for mobile robots, even in GPS-denied environments, has been made possible by the advantages offered by VINS. This capability has proven highly effective in addressing the unique challenges faced by these robots.

However, it is essential to acknowledge the significant computational complexity introduced by the substantial volume of data generated by the visual-inertial sensors. Achieving a delicate balance between computational complexity and estimation accuracy is crucial, especially in resource-constrained systems. This equilibrium is vital for ensuring the robustness of the system, particularly in scenarios demanding real-time performance. The development of reliable algorithms within the VI-SLAM system becomes imperative to effectively leverage onboard sensors for safe environment mapping and accurate pose estimation.

In the field of VI-SLAM, while highly efficient, conventional filtering-based methods can pose significant processing challenges in systems characterized by high dimensionality and high-frequency processing requirements. To overcome this, practical heuristic methods, such as sliding windows that marginalize past information, are commonly utilized. However, these methods introduce a trade-off, as they may lead to considerable information loss, resulting in substantial drift accumulation. The incorporation of keyframes is a strategy to address certain challenges. However, treating them as static variables, even with the continuous updating of correlation covariance, has the potential to introduce a compromise in accuracy.

In the optimization-based method, especially within the context of this thesis, problematic features like collinear features can lead to system divergence. In practice, heuristic methods are often employed by discarding these features and treating them as outliers through filtering. However, this approach can result in considerable information loss, affecting the accuracy of the system. Additionally, directly addressing the high-dimensional nonlinear optimization problem may lead to getting stuck in local minima.

Understanding the challenges inherent in both approaches within VI-SLAM unveils a complex landscape of possibilities and trade-offs. Ongoing improvements in these methodologies are poised to transform navigation systems, aiming for a balance between computational complexity and accuracy. Mitigating these challenges holds the potential to elevate the robustness and performance of navigation systems, particularly in the realm of mobile robot deployments.

1.4 Contributions

The main contributions of this thesis are:

- **Utilizing the Compressed Filtering Framework to Reduce Computational Complexity:** The thesis introduces the application of the compressed filtering framework to Multi-State Constraint Kalman Filter (MSCKF) including loop-closure. This approach preserves key-frame poses within the state vector while effectively managing computational complexity, achieving a complexity of $\mathcal{O}(N_L^2)$, where N_L denotes the number of local key-frames.

- **Enhanced Visual-Inertial SLAM with Parallax Bundle Adjustment:** The thesis evaluates the efficacy of parallax parametrization in VI-SLAM to address singularity issues common in VI-SLAM with Standard Bundle Adjustment (SBA). It highlights favourable attributes such as convergence, robustness, and high accuracy. The robustness of the Parallax Visual-Inertial SLAM (PVI-SLAM) system is significantly strengthened by leveraging the pre-integrated IMU method with Gaussian Process (GP) and incorporating the observation ray as an objective function.

- **Efficient Optimization through Linear Submap Joining utilizing PVI-SLAM:** The thesis presents Linear Submap Joining algorithms designed to tackle high-dimensional optimization challenges in PVI-SLAM. These algorithms significantly contribute to improving computational efficiency and reinforcing the overall robustness of the system. Rigorous evaluations on multiple datasets underscore their effectiveness, even in instances of suboptimal initialization.

1.5 Publications

1.5.1 Directly Related Publications

Parallax Visual-Inertial SLAM: Parallax Bundle Adjustment with IMU and Linear Submap Joining (Byun H, Zhao L, Kim J, Huang S, The 41st IEEE Conference on Robotics and Automation 2024, ICRA) ***(Under review)***

- This paper first proposes a new method for VI-SLAM. It uses a parallax angle for feature parametrization. The feature observation and the pre-integrated IMU information are used together to formulate a Nonlinear Least Squares (NLLS) problem.

To improve computational efficiency for large-scale problems involving a large number of poses, a linear submap joining method is proposed using the Linear SLAM framework. Local submaps are built using PVI-SLAM, and these submaps are then joined together through linear least squares and nonlinear coordinate transformations.

Comparison Between MATLAB Bundle Adjustment Function and Parallax Bundle Adjustment (Byun H, Kim J, Zhao L, Huang S, The 17th International Conference on Control, Automation, Robotics and Vision 2022, ICARCV)

- This paper evaluates two bundle adjustment techniques using SBA functions from MATLAB and Parallax Bundle Adjustment (PBA). The two Bundle Adjustment (BA) techniques are compared using data from the "Starry Night" and "MALAGA Parking-6L" with different initial inputs. In most cases, the results of PBA show better accuracy with lower final reprojection error and are less sensitive to the initialization values. Furthermore, VI-SLAM, based on PBA, has been presented.

Schmidt or Compressed filtering for Visual-Inertial SLAM? (Byun H, Kim J, Vanegas F, Gonzalez F, Australasian Conference on Robotics and Automation, Australasian Conference on Robotics and Automation 2021, ARAA)

- Focusing on VI-SLAM, computational complexity is a significant factor that needs to be considered, especially with small-scale applications. However, the accuracy of the system still needs to be ensured. Therefore, Compressed-MSCKF (Comp-MSCKF) has been proposed to ensure both the computational cost and accuracy of the system while Schmidt-MSCKF can yield sub-optimal performance.

Compressed Pseudo-SLAM: Pseudorange Integrated Generalised Compressed SLAM (Kim J, Byun H, Guivant J, Johansen T, 10 Dec 2020, Australasian Conference on Robotics and Automation, Australasian Conference on Robotics and Automation 2020, ARAA)

- The compressed SLAM has been proposed to acquire stable computational complex and accurate estimation by dividing the state vector into local and global to accumulate the information gained from the local part and update the global part much lower rate. It has been evaluated using the flight dataset from Unmanned Aerial Vehicle (UAV) with Global Navigation Satellite System (GNSS) and the visual-inertial sensor.

Cascaded Nonlinear Attitude Observer and Simultaneous Localisation and Mapping (Kim J, Bhambhani Y, Byun H, Johansen T, Australasian Conference on Robotics and Automation, Australasian Conference on Robotics and Automation 2020, ARAA)

- This paper presented a system that integrates the nonlinear observer theory and SLAM for aerial navigation. Using a nonlinear observer, the attitude of the platform can be estimated and the feedback term from utilizing the pseudo-inverse of a skew-symmetric matrix from the linear SLAM estimator increased the accuracy of the system. A simplified Lyapunov-based stability was also implemented.

1.5.2 Partially Related Publications

Towards a Pantograph-based Interventional AUV for Under-ice Measurement (Byun H, Kim J, Liu D, Woolfrey J, Australasian Conference on Robotics and Automation, Australasian Conference on Robotics and Automation 2021, ARAA)

- In this paper, the pantograph mechanism is presented with the concept design working with Autonomous Underwater Vehicles (AUVs). With the ability of the pantograph, it can effectively generate a constant interaction force to the surface during the contact, which aims to perform an autonomous sampling and measurement under the thin ice in the Antarctic environment.

Iterative Smoothing and Outlier Detection for Underwater Navigation (Hassan S, Byun H, Kim J, Australasian Conference on Robotics and Automation, Australasian Conference on Robotics and Automation 2021, ARAA)

- Due to the poor visibility causing significant outliers in underwater visual-inertial navigation, outlier detection and elimination became an essential part of the system. Existing methods show accurate outlier detection, yet, it is not valid for low-cost applications. Therefore, iterative smoothing and outlier detection utilizing Biswas-Mahalanabis Fixed-lag Smoother is proposed and demonstrated with the dataset collected from the underwater robots and fiducial makers.

1.6 Thesis Outline

This thesis is organized into six chapters, primarily focusing on presenting the technical contributions of the research in three of these chapters. Additionally, the appendices contain supplementary derivations and algorithms that complement and support the content presented in the technical sections.

Chapter 2 is dedicated to the literature review, exploring existing research in the field of VI-SLAM. The existing work is explored, particularly delving into two distinct classes of methods: filtering-based and optimization-based approaches. The chapter provides detailed insights into fundamental methodologies within each approach and also highlights benchmark studies conducted in the domain.

Chapter 3 delves into the first contribution, Comp-MSCKF, designed to enhance accuracy while maintaining moderate computational costs. The chapter elucidates the foundational methodology behind Comp-MSCKF and underscores the conceptual benefits demonstrated through the work on Compressed-Pseudo-SLAM. A detailed examination of Comp-MSCKF is provided, concluding with a comprehensive set of experiments conducted in simulated and real-world environments to showcase the effectiveness of the proposed algorithmic framework.

Moving on to **Chapter 4**, a novel method for VI-SLAM is proposed. This method utilizes the parallax angle for feature parametrization, combining feature observations and pre-integrated IMU information to formulate a nonlinear least squares problem. PVI-SLAM exhibits improved convergence properties compared to traditional methods using Euclidean XYZ as feature parametrization. To enhance system robustness in dynamic scenarios or with challenging initial values, alternative methods in objective functions and IMU pre-integration are integrated into the system.

Chapter 5 tackles the challenges posed by high-dimensional nonlinear optimization problems, which often lack guaranteed convergence and computational efficiency, especially in large-scale scenarios with numerous poses. A Linear Submap Joining method leveraging the Linear SLAM framework is proposed. The construction of local submaps is facilitated using the PVI-SLAM approach, and these submaps are smoothly joined through a fusion of linear least squares and nonlinear coordinate transformations. Importantly, this submap joining algorithm eliminates the necessity for initial guesses or iterative processes, as linear least squares problems offer closed-form solutions. Consequently, it provides results that closely approximate full nonlinear optimization.

Finally, in **Chapter 6**, a thorough summary of the contributions is provided, accompanied by a discussion of potential future work.

Chapter 2

Review of Related Work

Over the years, Visual SLAM (V-SLAM) systems have seen substantial advancements. The journey began with the introduction of Mono-SLAM [29], the first real-time monocular V-SLAM system, which utilized the Extended Kalman Filter (EKF) algorithm to estimate camera motion and 3D elements. Following Mono-SLAM, Parallel Tracking and Mapping (PTAM) [30] emerged, splitting the V-SLAM process into separate tracking and mapping threads to enhance computational efficiency. DTAM [15] introduced detailed mapping through dense tracking and mapping modules, albeit with a high computational cost. Subsequent innovations included an RGB-D camera-based method [31], tailored for cost-effective implementations in small robots, and SLAM++ [32], which integrated semantic information to enhance mapping accuracy. Further advancements brought Semi-direct Visual Odometry (SVO) [33], which combined feature-based and direct methods for robust motion estimation, and LSD-SLAM [16], specialized for large-scale map reconstruction. ORB-SLAM [34] and its successor ORB-SLAM 2 [35] effectively utilized ORB features for localization and mapping, though they faced challenges in texture-less environments and with unknown scales. ORB-SLAM 3 [12] addressed these challenges by supporting various camera types and advancing pose estimation methodologies [36].

Despite significant advancements in pure V-SLAM algorithms, challenges persist in handling image blur from fast camera movements and poor illumination when relying solely on cameras as sensors. The integration of cameras with IMUs has emerged as a key area of research, significantly enhancing the robustness and accuracy of V-SLAM systems in various scenarios [37].

Initially, researchers explored the loose coupling of IMU data with existing V-SLAM methods [38–40]. While this approach is relatively straightforward to implement, it suffers from error susceptibility and has not undergone extensive research [41]. The development of hybridization filters marked a significant advancement towards “tightly coupled” visual-inertial methods. These methods, now widely used in systems equipped with both IMUs and cameras, offer improved performance and reliability by more effectively fusing visual and inertial data [41].

In this chapter, the related work on VI-SLAM is introduced and categorized into two main approaches: filtering-based and optimization-based methods. Firstly, basic filtering methods, specifically the EKF, are explained. This is followed by an overview of related work in the field of filtering-based approaches. Subsequently, the chapter delves into least square problems and provides an explanation of optimization-based methods along with a discussion of related works in this category. The structured presentation aims to provide a comprehensive understanding of the existing literature and approaches in the domain of VI-SLAM.

2.1 Filtering-Based Methods

2.1.1 Extended Kalman Filter SLAM

The EKF serves as a classic solution in SLAM, historically pioneering the field. It operates by estimating the state, encompassing the current robot pose, $\mathcal{P}$, and environmental feature parameters, $\mathcal{F}$:

$$\mathcal{X} = \begin{bmatrix} \mathcal{P} \\ \mathcal{F} \end{bmatrix} = \begin{bmatrix} \mathcal{P} \\ \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_n \end{bmatrix}. \tag{2.1}$$

Notably, EKF excludes the past robot pose from the state. The EKF continuously expands the state vector by incorporating feature parameters as they become available, offering insights into the uncertainty of both pose and map through the covariance matrix:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathcal{P}\mathcal{P}} & \mathbf{P}_{\mathcal{P}\mathcal{F}} \\ \mathbf{P}_{\mathcal{F}\mathcal{P}} & \mathbf{P}_{\mathcal{F}\mathcal{F}} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{\mathcal{P}\mathcal{P}} & \mathbf{P}_{\mathcal{P}\mathbf{f}_1} & \cdots & \mathbf{P}_{\mathcal{P}\mathbf{f}_n} \\ \mathbf{P}_{\mathbf{f}_1\mathcal{P}} & \mathbf{P}_{\mathbf{f}_1\mathbf{f}_1} & \cdots & \mathbf{P}_{\mathbf{f}_1\mathbf{f}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{\mathbf{f}_n\mathcal{P}} & \mathbf{P}_{\mathbf{f}_n\mathbf{f}_1} & \cdots & \mathbf{P}_{\mathbf{f}_n\mathbf{f}_n} \end{bmatrix} \tag{2.2}$$

Upon receiving sensor measurements, the EKF exhibits an adaptive behavior, dynamically refining its state and covariance matrix to mitigate uncertainty actively. In the context of feature-based SLAM, the available information can be categorized into two types: odometry information, represented by the motion model, and observation information, delineated by the observation model [42].

During the prediction phase, the EKF anticipates the subsequent state by harnessing the motion model. This model intricately captures the expected movements of the system, factoring in the current state and received control inputs. Simultaneously, the observation information is crucially considered during the update phase, refining the system's predictions by aligning them with the observed measurements.

2.1.1.1 Prediction Step

When the control input vector from the IMU is received at time $k-1$, denoted as $\mathbf{u}_{k-1}$, the generic motion model using the function $f(\cdot)$ can be expressed as:

$$\mathcal{X}_k \leftarrow f(\mathcal{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{n}_{I_{k-1}}). \tag{2.3}$$

Given that the motion model specifically pertains to the robot pose, it is applied differently to the pose and feature positions, resulting in distinct expressions:

$$\mathcal{P}_k \leftarrow f_{\mathcal{P}}(\mathcal{P}_{k-1}, \mathbf{u}_{k-1}, \mathbf{n}_{I_{k-1}}), \tag{2.4}$$

$$\mathcal{F}_k \leftarrow \mathcal{F}_{k-1}, \tag{2.5}$$

where $\mathbf{n}_I$ represents the zero-mean Gaussian process noise from IMU measurements with the covariance matrix $\mathbf{Q}$. The prediction step of the EKF for the state and its corresponding covariance is described by:

$$\hat{\mathcal{X}}_{k|k-1} \leftarrow f(\hat{\mathcal{X}}_{k-1|k-1}, \mathbf{u}_{k-1}, 0), \tag{2.6}$$

$$\mathbf{P}_{k|k-1} \leftarrow \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^\top + \mathbf{G}\mathbf{Q}_{k-1}\mathbf{G}^\top. \tag{2.7}$$

Here, $\hat{\mathcal{X}}_{k|k}$ represents the estimate of $\mathcal{X}$ at time k given observations up to and including time k. The matrices $\mathbf{F} = \frac{\partial f(\mathcal{X},\mathbf{u},\mathbf{n})}{\partial \mathcal{X}}$ and $\mathbf{G} = \frac{\partial f(\mathcal{X},\mathbf{u},\mathbf{n})}{\partial \mathbf{n}}$ are system Jacobians with respect to the state vector and noise, respectively. Following the application of the motion model to the state at time $k-1$ in the prediction step (as described by Equation (2.6)), and in accordance with Equation (2.4) and Equation (2.5), the pose and features in the prediction step can be expressed as:

$$\hat{\mathcal{P}}_{k|k-1} \leftarrow f_{\mathcal{P}}(\hat{\mathcal{P}}_{k-1|k-1}, \mathbf{u}_{k-1}, 0), \tag{2.8}$$

$$\hat{\mathcal{F}}_{k|k-1} \leftarrow \hat{\mathcal{F}}_{k-1|k-1}. \tag{2.9}$$

This results in sparse Jacobians:

$$\mathbf{F} = \begin{bmatrix} \frac{\partial f_{\mathcal{P}}}{\partial \mathcal{P}} & 0 \\ 0 & \mathbf{I} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \frac{\partial f_{\mathcal{P}}}{\partial \mathrm{n}} \\ 0 \end{bmatrix}. \tag{2.10}$$

2.1.1.2 Update Step

The subsequent phase initiates when observations from the vision sensor are received. The generalized nonlinear camera measurement model, denoted as $\mathbf{z}_k$, for the EKF can be expressed as:

$$\mathbf{z}_k = h(\mathcal{X}_k) + \mathbf{n}_{f_k}, \tag{2.11}$$

where $h(\cdot)$ denotes the observation function, and $\mathbf{n}_f$ represents white Gaussian noise with covariance $\mathbf{R}$. Utilizing this model, the standard EKF update unfolds through the following steps:

$$\mathbf{e}_k = \mathbf{z}_k - h(\hat{\mathcal{X}}_{k|k-1}), \tag{2.12}$$

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \mathbf{R}_k, \tag{2.13}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^\top\mathbf{S}_k^{-1}, \tag{2.14}$$

$$\hat{\mathcal{X}}_{k|k} \leftarrow \hat{\mathcal{X}}_{k|k-1} + \mathbf{K}_k\mathbf{e}_k, \tag{2.15}$$

$$\mathbf{P}_{k|k} \leftarrow (\mathbf{I} - \mathbf{K}_k\mathbf{H})\,\mathbf{P}_{k|k-1}. \tag{2.16}$$

Here, $\mathbf{H} = \frac{\partial h(\mathcal{X})}{\partial \mathcal{X}}$ represents the measurement Jacobian with respect to the state. $\mathbf{e}$ signifies the measurement residual, and $\mathbf{S}$ stands for its covariance matrix. The state and its covariance are updated using the Kalman gain, $\mathbf{K}$, as outlined in Equation (2.15) and Equation (2.16).

2.1.2 Filtering-based Visual-Inertial SLAM

The EKF is a widely used robust state estimation algorithm, especially in nonlinear dynamics systems. However, challenges like partial observability can introduce inconsistencies in the EKF, potentially causing suboptimal performance and even leading to divergence or biased estimates [43, 44].

To overcome the limitation arising from the underestimation of uncertainty associated with the estimate, resulting in an overly confident outcome [45, 46], enhanced variants of the EKF SLAM have been proposed. Castellanos et al. introduced the Robocentric EKF SLAM [47]. This variant addresses linearization errors by dynamically adapting the coordinate frame based on the robot's local coordinate frame. The First Estimates Jacobian EKF (FEJ-EKF) [45], proposed by Huang et al. computes the Jacobian within EKF using initial available estimates. This results in an error-state system model with an observable subspace dimension matching the underlying nonlinear SLAM system. The Observability Constrained EKF (OC-EKF), introduced in works by Li et al. [48], Hesch et al. [49], and Huang et al. [50], also offers enhancements in terms of consistency. The incorporation of Lie group representation [51, 52] facilitates the introduction of the Invariant EKF (I-EKF) [53–55]. This variant includes a geometrically adapted correction term based on an invariant output error. This approach prevents covariance reduction in directions of the state space where no information is available. Right Invariant Error EKF (RI-EKF) [46, 56] has been further improved to demonstrate that the output of the filter remains invariant under any stochastic rigid body transformation.

However, filtering methods present a computational challenge for embedded processors in small-scale platforms. To address this, Thrun et al. proposed the Sparse Extended Information Filters (SEIF) [57], leveraging sparsity in the information matrix. In many large-scale systems, not all variables are interconnected, leading to a sparse covariance matrix. SEIF exploits this sparsity to significantly reduce computational requirements. FastSLAM [58] takes a particle filtering-based approach to represent the belief about the robot's pose and the map of the environment. This algorithm can leverage parallelism by processing particles independently, enabling efficient implementation on parallel computing platforms. An improved version was presented in [59], where the distribution relies not only on the motion estimate but also on the most recent sensor measurement. The concept of Compression is applied to EKF SLAM in [60]. This involves partitioning the local and global components, thereby reducing computational complexity related only to the number of features in the defined local map. The global part is updated only when a new local boundary is defined. This approach has been successfully implemented in various filtering-based approaches [2, 61, 62]. The adaptation of the Schmidt Kalman Filter (SKF) [63] is intended to reduce computational complexity by considering certain parameters as static [64]. These parameters are no longer updated, but their covariance and correlated covariance with other states are still utilized in the EKF update. Through this method, the computational complexity becomes linear with respect to the number of features, making it more feasible for implementation in resource-constrained environments.

In the field of VI-SLAM, researchers have successfully integrated filtering-based approaches, showcasing notable examples such as VIO-ROVIO [65] and maplab [66]. More recently, attention has been directed towards harnessing the capabilities of the MSCKF within the context of VI-SLAM, as evidenced by various works [6, 64, 67, 68]. The MSCKF is an EKF-based algorithm that strategically maintains a sliding window of camera poses in the state vector. It uses feature observations to establish probabilistic constraints among these poses. Unlike traditional EKF SLAM approaches, the MSCKF does not approximate the feature position probability density function with a Gaussian. This unique characteristic sets the MSCKF apart from traditional EKF SLAM methods, holding the potential for superior performance. One key advantage of the MSCKF lies in its linear computational complexity with the number of features, contrasting with the cubic complexity often associated with feature-based SLAM approaches. This linear complexity enhances the computational efficiency of the MSCKF, making it particularly advantageous in resource-constrained environments [69]. Modifications have been introduced to ensure correct observability properties without incurring additional computational costs [69]. A stereo version of the MSCKF has been proposed in [24].

Despite the advancements in MSCKF, a limitation in the long-term consistency of the VI-SLAM system arises from the fact that MSCKF does not retain all past poses using sliding windows. To enhance long-term consistency, introducing loop-closure constraints becomes essential. However, it's crucial to acknowledge that incorporating loop-closure constraints may result in increased computational costs. To overcome the challenge of long-term consistency, Schmidt-MSCKF [6] strategically incorporates keyframes of camera poses in the loop-closure process, focusing on keyframes rather than adding all camera poses. By treating keyframe state vectors as nuisance parameters, significant computational savings are achieved. This strategic approach ensures long-term consistency without imposing excessive computational burdens. Furthermore, OpenVINS [11] introduces an open platform online system built upon the MSCKF. This system provides a flexible and extensible framework for the online VI-SLAM system.

2.2 Optimization-Based Methods

2.2.1 Least Squares SLAM

In the realm of optimization-based SLAM, where achieving superior estimation results is paramount, the method distinguishes itself by executing re-linearization at each step, ensuring the system's consistency. This approach involves incorporating not only the current pose but also all past poses and observed features into the state vector. This state vector can be represented as:

$$\mathcal{X} = \begin{bmatrix} \mathcal{X}_{\mathcal{P}} \\ \mathcal{F} \end{bmatrix} = \begin{bmatrix} \mathcal{P}_1 \\ \vdots \\ \mathcal{P}_m \\ \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_n \end{bmatrix}. \tag{2.17}$$

The dedicated framework of Least Squares in SLAM revolves around treating SLAM as an optimization problem, seeking the optimal state vector denoted as $\mathcal{X}^*$. This optimization task is approached as a Maximum Likelihood Estimation (MLE) problem, aiming to minimize the negative log-likelihood of the available sensor measurements (denoted as $\mathcal{Z}$) given the state [70]:

$$\mathcal{X}^* = \arg\max_{\mathcal{X}} \ (P(\mathcal{Z} \mid \mathcal{X})) = \arg\min_{\mathcal{X}} \ (-\log(P(\mathcal{Z} \mid \mathcal{X}))). \tag{2.18}$$

Under the Gaussian model, Equation (2.18) is equivalent to minimizing the objective function, $J(\mathcal{X})$,

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} J(\mathcal{X}), \tag{2.19}$$

where the comprehensive objective function consolidates the accumulated cost over all time steps:

$$J(\mathcal{X}) = \sum_{i=1}^{N-1} \mathcal{C}_{imu}^{(i)} + \sum_{i=1}^{N} \mathcal{C}_{\text{cam}}^{(i)}. \tag{2.20}$$

Equation (2.19) aims to minimize the sum of squared residuals (Equation (2.20)) where the cost functions of IMU, $\mathcal{C}_{imu}$, and camera, $\mathcal{C}_{cam}$, at time i can be expressed as:

$$\mathcal{C}_{imu}^{(i)} = \left\| \mathbf{e}_{imu}^{(i)} \right\|_{\mathbf{Q}_i}^2 , \tag{2.21}$$

$$\mathcal{C}_{cam}^{(i)} = \left\| \mathbf{e}_{cam}^{(i)} \right\|_{\mathbf{R}_i}^2 . \tag{2.22}$$

This residual, $\boldsymbol{e} = \left[\begin{array}{cc} \boldsymbol{e}_{imu}^\top & \boldsymbol{e}_{cam}^\top \end{array} \right]^\top$, represent the disparity between predicted and observed measurements across all time steps. The residual at time i for the IMU, $\mathbf{e}_{imu}^{(i)}$, and vision sensors, $\mathbf{e}_{cam}^{(i)}$, in VI-SLAM, utilizing the functions described in Equation (2.8) and Equation (2.12), respectively, can be expressed as:

$$\mathbf{e}_{imu}^{(i)} = \mathcal{P}_i - f_p \left(\mathcal{P}_{i-1}, \mathbf{u}_{i-1}, 0 \right), \tag{2.23}$$

$$\mathbf{e}_{cam}^{(i)} = \mathbf{z}_i - h \left(\mathcal{X}_i \right). \tag{2.24}$$

The One-Dimensional (1D) SLAM problem utilizes a Linear Least Squares (LLS) approach with a closed-form solution since the functions $f_p(\cdot)$ and $h(\cdot)$ are linear. In contrast, the more complex 2D and 3D SLAM scenarios require a NLLS formulation [42].

2.2.2 Gauss-Newton Iteration and Levenberg-Marquardt Method

For the standard minimization method, the Gauss-Newton (GN) and Levenberg-Marquardt (LM) algorithm are usually used to solve Eqaution (2.19). In the GN method, the update to the state vector at each iteration, k, is given by:

$$\mathcal{X}_{k+1} = \mathcal{X}_k + \Delta_k, \tag{2.25}$$

where Δ_k is the update calculated as:

$$\Delta_k = -(\mathbf{J}^\top \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^\top \mathbf{W}\ \mathbf{e}, \tag{2.26}$$

Here, $\mathbf{J}$ is the Jacobian matrix, capturing the Jacobian of the residual with respect to X evaluated at X_k, and $\mathbf{W}$ is the weight matrix, obtainable by inverting the covariance matrix stacked with $\mathbf{Q}_i$ and $\mathbf{R}_i$. The corresponding covariance matrix for the optimized state vector can then be obtained as $(\mathbf{J}^\top \mathbf{W} \mathbf{J})^{-1}$.

The introduction of a damping parameter λ to the GN method results in the LM algorithm. The update expression is then modified to:

$$\Delta_k = -\left(\mathbf{J}^\top \mathbf{W} \mathbf{J} + \lambda \mathbf{E}\right)^{-1} \mathbf{J}^\top \mathbf{W}\ \mathbf{e}, \tag{2.27}$$

where $\mathbf{E}$ is the identity matrix. The inclusion of the damping term enhances the stability of the optimization process, particularly in scenarios where the GN method may encounter numerical challenges.

2.2.3 Gauss-Newton on Manifold

The optimization on the manifold follows the "lift-solve-retract" scheme, a well-established methodology detailed in [71]. This systematic approach is particularly prevalent within the framework of trust-region methods. It provides an efficient means of performing optimization on manifolds, striking a balance between leveraging the advantages of Euclidean space for optimization and ensuring the maintenance of valid solutions on the manifold. This is crucial for respecting any inherent constraints or structures present in the problem domain [71].

The process initiates with a "lifting" operation, wherein the optimization problem in Eqaution (2.19) is reparametrized to operate in a tangent space associated with the current estimate on the manifold [1]. This transformation is denoted as:

$$\begin{gathered} \mathcal{X}^* = \underset{\mathcal{X} \in \mathcal{M}}{\arg\min}\ J(\mathcal{X}) \\ \Downarrow \\ \delta \mathbf{x}^* = \underset{\delta \mathbf{x} \in \mathbb{R}^n}{\arg\min}\ J\left(\mathcal{R}_x(\delta \mathbf{x})\right). \end{gathered} \tag{2.28}$$

Here, $\mathcal{R}_x$ serves as a bijective retraction map, facilitating the mapping between an element $\delta \mathbf{x}$ in the tangent space, $\mathbb{R}^n$, and a neighbourhood around the current estimate $\mathcal{X}$ on the manifold in n dimension, $\mathcal{M}$ [1]. This lifting operation transforms the optimization problem from a manifold-based representation to an auxiliary Euclidean space, allowing the application of standard optimization techniques.

Once the problem is in the lifted Euclidean space, conventional optimization techniques like Gauss-Newton (Section 2.2.2) can be applied to minimize the cost function. The cost function is typically near quadratic in $\delta \mathbf{x}$ around the current estimate, resulting in a

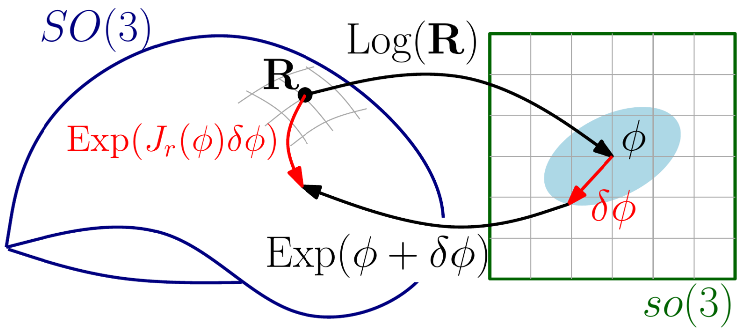


FIGURE 2.1: The right Jacobian J_r establishes a connection between an additive perturbation $\delta\boldsymbol{\phi}$ in the tangent space and a multiplicative perturbation on the manifold SO(3) [1].

quadratic approximation. The solution to this quadratic approximation provides a vector $\delta\mathbf{x}^*$ in the tangent space.

The final step involves "retracting" the updated estimate from the lifted space back to the manifold using the inverse of the lifting operation. The retraction map $\mathcal{R}_x$ is crucial in this step, as it maps the updated tangent space element $\delta\mathbf{x}$ back to a new estimate on the manifold [1]:

$$\hat{\mathcal{X}} \leftarrow \mathcal{R}_{\hat{x}}\left(\delta\mathbf{x}^{\star}\right). \tag{2.29}$$

This updated estimate becomes the starting point for the subsequent iteration of the optimization process, facilitating a coherent and effective procedure for optimization on manifold.

2.2.4 Geometric Concepts on Manifold

The choice of rotation characterized by the Lie group known as Special Orthogonal group in three dimensions (SO(3)) is advantageous due to its freedom from singularities; however, it introduces certain constraints. In contrast, Lie algebra of special orthogonal group in three dimensions ($\mathfrak{so}(3)$) avoids these constraints but faces challenges associated with singularities. A strategic approach is adopted to address these issues effectively. The dominant and nominal components are retained within SO(3), ensuring singularity-free representation. Simultaneously, the smaller, noisy components are accommodated in $\mathfrak{so}(3)$, which is constraint-free and treated as a vector space. This approach provides a balanced solution to the challenges associated with rotations in the context of state representation [72].

Formally, SO(3) is defined as $\mathrm{SO}(3) \doteq \left\{\mathbf{R} \in \mathbb{R}^{3\times3} : \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\right\}$ [25]. This group constitutes a smooth manifold, capturing the essence of rotational transformations. Now, consider the tangent space to this manifold, $\mathfrak{so}(3)$. It coincides with the set of 3×3 skew-symmetric matrices. These skew-symmetric matrices find expression as vectors in $\mathbb{R}^3$ through the $\wedge$ operator:

$$\boldsymbol{\phi}^\wedge = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathfrak{so}(3). \tag{2.30}$$

Here, $\boldsymbol{\phi}$ represents a 3-by-1 axis-angle vector. A noteworthy property of skew-symmetric matrices, crucial in this context, is given by:

$$\boldsymbol{a}^\wedge \boldsymbol{b} = -\boldsymbol{b}^\wedge \boldsymbol{a}, \quad \forall \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3. \tag{2.31}$$

The exponential map $\mathfrak{so}(3)$ to SO(3) is a fundamental concept in rotational transformations and is defined by:

$$\exp\left(\boldsymbol{\phi}^\wedge\right) = I + \frac{\sin(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|}\boldsymbol{\phi}^\wedge + \frac{1 - \cos(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|^2}\left(\boldsymbol{\phi}^\wedge\right)^2. \tag{2.32}$$

This operation maps a skew-symmetric matrix to a rotation matrix. Conversely, the logarithm map associates a matrix $\mathbf{R}$ with a skew-symmetric matrix:

$$\log(\mathbf{R}) = \frac{\varphi \cdot \left(\mathbf{R} - \mathbf{R}^\top\right)}{2\sin(\varphi)}, \tag{2.33}$$

where the rotation angle, φ, is determined by $\varphi = \cos^{-1}\left(\frac{\mathrm{tr}(\mathbf{R})-1}{2}\right)$. In another form, the logarithm map can be expressed as $\log(\mathbf{R}) = \mathbf{c}^\wedge \varphi$, where $\mathbf{c}$ represents the rotation axis. When $\mathbf{R}$ is equal to the identity matrix, the rotation angle φ becomes 0, and the rotation axis $\mathbf{c}$ cannot be defined [73]. In such scenarios, the choice of a rotation axis is arbitrary due to the absence of rotation [1].

Several key properties of the exponential map are:

$$\exp\left(\boldsymbol{\phi}^\wedge\right) \approx I + \boldsymbol{\phi}^\wedge, \tag{2.34}$$

$$\exp\left(\boldsymbol{\phi}^\wedge\right)^{-1} = \exp\left(-\boldsymbol{\phi}^\wedge\right), \tag{2.35}$$

$$\mathbf{R}\exp(\boldsymbol{\phi}^\wedge)\mathbf{R}^\top = \exp\left(\left(\mathbf{R}\boldsymbol{\phi}^\wedge\mathbf{R}^\top\right)^\wedge\right) = \exp\left((\mathbf{R}\boldsymbol{\phi}))^\wedge\right), \tag{2.36}$$

$$\exp(\boldsymbol{\phi}^\wedge)\mathbf{R} = \mathbf{R}\exp\left(\left(\mathbf{R}^\top\boldsymbol{\phi}\right)^\wedge\right). \tag{2.37}$$

Additionally, first-order approximations for the exponential and logarithm with additive perturbation, $\delta\boldsymbol{\phi}$, can be derived:

$$\exp\left((\boldsymbol{\phi}+\delta\boldsymbol{\phi})^\wedge\right) \approx \exp(\boldsymbol{\phi}^\wedge)\exp\left((J_r(\boldsymbol{\phi})\delta\boldsymbol{\phi})^\wedge\right), \tag{2.38}$$

$$\log(\exp(\boldsymbol{\phi}^\wedge)\exp(\delta\boldsymbol{\phi}^\wedge)) \approx \boldsymbol{\phi} + J_r^{-1}(\boldsymbol{\phi})\delta\boldsymbol{\phi}. \tag{2.39}$$

As can be seen in Figure 2.1, the right Jacobian of SO(3), $J_r(\boldsymbol{\phi})$ connects $\delta\boldsymbol{\phi}$ in the tangent space to a multiplicative perturbation on the manifold SO(3) [1]. It's essential to emphasize that both $J_r(\boldsymbol{\phi})$ and its inverse $J_r^{-1}(\boldsymbol{\phi})$ become to the identity matrix when $\|\boldsymbol{\phi}\| = 0$.

$$J_r(\boldsymbol{\phi}) = I - \frac{1-\cos(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|^2}\boldsymbol{\phi}^\wedge + \frac{\|\boldsymbol{\phi}\| - \sin(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}^3\|}\left(\boldsymbol{\phi}^\wedge\right)^2. \tag{2.40}$$

The inverse of the right Jacobian is

$$J_r^{-1}(\boldsymbol{\phi}) = I + \frac{1}{2}\boldsymbol{\phi}^\wedge + \left(\frac{1}{\|\boldsymbol{\phi}\|^2} + \frac{1+\cos(\|\boldsymbol{\phi}\|)}{2\|\boldsymbol{\phi}\|\sin(\|\boldsymbol{\phi}\|)}\right)\left(\boldsymbol{\phi}^\wedge\right)^2. \tag{2.41}$$

For notational convenience, Exp and Log are adopted from [1]:

$$\begin{aligned} \mathrm{Exp}: &\quad \mathbb{R}^3 \to \mathrm{SO}(3) \quad ; \quad \boldsymbol{\phi} \mapsto \exp\left(\boldsymbol{\phi}^\wedge\right), \\ \mathrm{Log}: &\quad \mathrm{SO}(3) \to \mathbb{R}^3 \quad ; \quad \mathbf{R} \mapsto \log(\mathbf{R})^\vee. \end{aligned} \tag{2.42}$$

2.2.5 Optimization-based Visual-Inertial SLAM

In the domain of SLAM, the computational complexity presents a notable challenge, especially in optimization-based methodologies. An effort to tackle these computational challenges can be found in the work of Ranganathan et al. [74] and Sibley et al. [75], which introduces fixed-lag smoothing approaches. This processes sensor measurements and refines the estimated state exclusively within a predetermined fixed-lag time window. To manage computational complexity, fixed-lag smoothing employs the marginalization of older states and measurements located outside the fixed-lag window. While this strategy helps control computational costs, it introduces a potential drawback—loss of sparsity in the information matrix [76]. Sparse representations can enhance the stability and numerical properties of optimization algorithms. Dense matrices may result in ill-conditioned problems, posing challenges for optimization algorithms to converge reliably. Dong-Si et al. [77] introduced a modification to the algorithm's linearization process with the specific aim of preventing the introduction of information along directions in the state space where no actual information is provided by the measurements.

Taking a different approach, iSAM [78] and its enhanced version, iSAM2 [79], employ incremental processing of sensor measurements, dynamically refining the state estimate as fresh data unfolds. These algorithms adopt a factor graph framework, offering a graphical depiction that captures the intricate relationships among variables and the constraints imposed by sensor measurements. To achieve efficient incremental updates, iSAM and iSAM2 employ Givens rotations, an orthogonal transformation technique. This approach allows for the incremental enhancement of QR decomposition and Cholesky factorization without necessitating a complete recomputation, optimizing the computational efficiency of the algorithms. However, it is acknowledged that this method may face challenges related to accuracy, particularly with the accumulation of linearization errors in scenarios involving frequent loop-closures [76].

Klein et al. introduced PTAM [30], a system employing a keyframe-based strategy for environmental mapping. This approach selectively chooses keyframes and calculates a 3D map for this subset at a reduced frame rate, discarding non-keyframes to streamline the process [76]. Unlike approaches that discard information from non-keyframes, C-KLAM [76] maximizes the use of this data. It leverages most of the information to establish consistent pose constraints between keyframes while preserving the sparsity of the information matrix.

Despite the efficacy of keyframes in SLAM, challenges emerge as the trajectory expands, primarily stemming from the increased size of the state vector. This growth in computational complexity raises concerns about the system's robustness, particularly when confronted with high-dimensional nonlinear optimization. To address these issues, various research endeavours within the realm of SLAM have strategically tackled the balance between computational efficiency and system resilience [80–84]. A notable contribution in this context is the Linear SLAM framework proposed by Zhao et al. [85]. Unlike methods that require initial guesses or iterations, this framework leverages closed-form solutions for linear least squares problems, enhancing computational efficiency while maintaining accuracy in the optimization process.

In the realm of VI-SLAM, there is a growing emphasis on nonlinear optimization techniques driven by the advancements in computer technology. These techniques are known to offer higher accuracy when compared to traditional filtering-based methods. Many researchers have adopted the above-mentioned techniques to manage computational costs effectively. OKVIS [86] introduces an optimization-based approach centered around a keyframe-based framework. This method optimally integrates inertial and reprojection errors while marginalizing past poses. VINS-Fusion [10], on the other hand, adopts a graph-based approach, implementing local window optimization with loop-closure to enhance performance. It employs a 4-DoF pose graph optimization technique to ensure global consistency. Balancing accuracy and computational complexity, optimization-based VI-SLAM often leads to keyframe-based systems like ORB-SLAM3 [12], which uses ORB descriptors for feature matching and operates with three parallel threads: tracking, local mapping, and loop closing.

In the specific context of IMU, Lupton et al. [25] pioneered the pre-integration method. This approach aims to mitigate the issue of repeated constraints arising from the parametrization of relative motion integration, ultimately reducing computational complexity in VI-SLAM. Forster et al. [1] modified the pre-integration method, offering a more formal treatment of rotation noise. This modification is crucial for addressing the manifold structure of the SO(3), providing a more accurate representation of rotational dynamics. Additionally, Le Gentil et al. introduced a novel pre-integration method known as Unified Gaussian Preintegrated Measurement (UGPM) [8], addressing the challenge of continuous pre-integration over Special Euclidean group in three dimensions (SE(3)) using GP. The incorporation of GP models enables accurate pre-integrated measurements, thereby enhancing accuracy, particularly in dynamic motion scenarios.

In modern VI-SLAM [11], [86], [10], [12], [87], the BA algorithm plays a pivotal role as the central back-end process. BA typically involves representing feature locations using Euclidean XYZ coordinates. An alternative method is to parametrize feature positions using the inverse-depth method, as elaborated in [88]. However, both XYZ parametrizations and Inverse Depth Parametrization (IDP) exhibit limitations, particularly in scenarios where camera motion aligns with the feature's direction or when the feature is at a considerable distance, resulting in a zero parallax angle. To address these challenges, [89] introduced the parallax parametrization, which incorporates the parallax angle directly into the state vector. This approach has demonstrated superior performance in terms of accuracy, efficiency, and convergence compared to traditional methods. The parallax parametrization proves particularly advantageous in scenarios where standard parametrizations may face limitations, highlighting its significance in advancing the capabilities of VI-SLAM systems.

Chapter 3

Compressed Visual-Inertial SLAM

To ensure the efficiency of monocular VI-SLAM within resource-constrained environments, it is essential to balance computational cost and estimation accuracy, thereby ensuring robust and reliable performance. This chapter is centered on filtering-based methodology, commencing with strategies to manage computational complexity without compromising accuracy. As a result, **Comp-MSCKF**, a novel approach incorporating loop-closure, is introduced. This entails defining the system state in a compressed manner based on MSCKF principles. The compression methodology is detailed for both the propagation and update steps. The study incorporates an analysis of the convergence of uncertainty in key-frame states, evaluated using a MATLAB simulator. Furthermore, the performance of the proposed system is assessed with real-world datasets, showcasing superior accuracy with reasonable computational demands. Overall, these considerations typically result in a computational complexity of $\mathcal{O}(N_L^2)$, where N_L denotes the number of local key-frames, while maintaining the incorporation of loop-closure into the system.

3.1 Reducing Computational Complexity in SLAM: Compressed SLAM and MSCKF

As mentioned in Section 2.1, filtering-based SLAM solutions play a pivotal role in estimating the state, which includes both the agent's current pose and the environmental feature parameters encountered during exploration. As the agent navigates new regions, these solutions consistently integrate incoming feature parameters into the state vector, leading

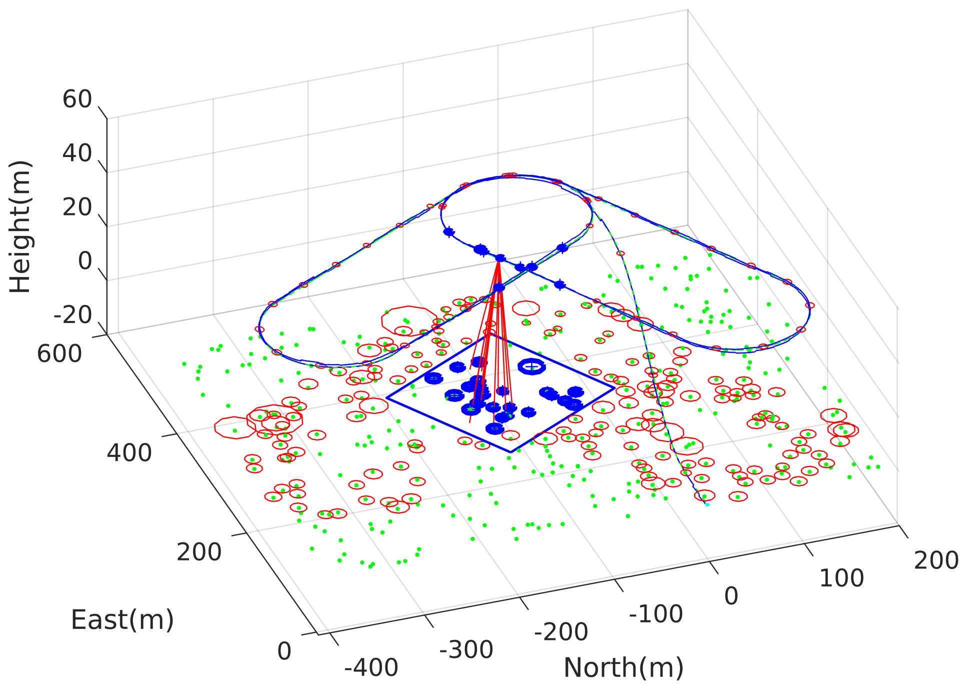


FIGURE 3.1: The compressed filter divides the environment into two regions: a local area (depicted as a rectangular box beneath the vehicle, with map uncertainty ellipses in blue) and a global region (outside the box, with map uncertainty ellipses in red). The local area is redefined each time the vehicle crosses its boundary.

to a continuous expansion of the state size. The continual growth in the state vector profoundly impacts the overall cost of the SLAM solution, resulting in high computational complexity. Typically, this complexity scales quadratically, denoted as $\mathcal{O}(N^2)$, where N signifies the total number of features present in the system.

Various approaches have been explored to address the computational challenge associated with the increasing state size. This section specifically delves into the compression technique introduced by Guivant et al. [60] and standard MSCKF proposed by Mourikis et al. [4]. These methodologies are integrated into the proposed Comp-MSCKF approach, as discussed in Section 3.2.

3.1.1 Compressed SLAM

The concept of compression was initially introduced in [60] to manage the computational cost of SLAM solutions effectively. As can be seen in Figure 3.1, the compression approach divides a large map of the state, $\mathcal{X}$, into local, $\mathcal{X}_L$, and global, $\mathcal{X}_G$, maps as follows:

$$\mathcal{X} = \left[\begin{array}{c} \mathcal{X}_L \\ \hline \mathcal{X}_G \end{array}\right] = \left[\begin{array}{c} \mathcal{P}_I \\ \mathcal{F}_L \\ \hline \mathcal{F}_G \end{array}\right]. \tag{3.1}$$

Here, $\mathcal{P}_I$ represents the current IMU state, and $\mathcal{F}_L$ and $\mathcal{F}_G$ denote features located in the local and global maps, respectively. The corresponding covariance matrix can be expressed as:

$$\mathbf{P} = \left[\begin{array}{c|c} \mathbf{P}_{LL} & \mathbf{P}_{LG} \\ \hline \mathbf{P}_{GL} & \mathbf{P}_{GG} \end{array}\right] = \left[\begin{array}{cc|c} \mathbf{P}_{II} & \mathbf{P}_{IF_L} & \\ \mathbf{P}_{F_L I} & \mathbf{P}_{F_L F_L} & \mathbf{P}_{LG} \\ \hline \multicolumn{2}{c|}{\mathbf{P}_{GL}} & \mathbf{P}_{GG} \end{array}\right]. \tag{3.2}$$

It updates the local map with a quadratic complexity of $\mathcal{O}\left(N_L^2\right)$, where N_L represents the size of the local map, which is usually much smaller than the total number of features, denoted as N. Additionally, the method compresses the correlation information between local and global map and propagates it to the global map only when the vehicle crosses the boundary of the local map. This strategy effectively manages computational complexity while maintaining map accuracy.

In [62], Guivant et al. proposed Generalized Compressed Kalman Filter (GCKF). Unlike the standard compressed filtering, where the local and global correlation is explicitly computed using a closed-form expression, the generalized approach is formulated based on the Bayesian framework. This not only simplifies the process with various local filters but also facilitates its extension to multiple vehicle applications.

Utilizing the GCKF, the Compressed Pseudo-SLAM (CP-SLAM) was introduced as described in [2], which fuses pseudo-range observations from GNSS with VI-SLAM. The primary aim of this integration was to enhance navigation reliability and robustness for UAV operating in near-Earth environments, where GNSS signals are typically available. The fusion filter models and estimates the receiver clock and drift, which is crucial for integrating pseudorange rate measurements. Subsequently, efficient accumulation of information from a local map and updating the global map at a lower rate was achieved using GCKF. This approach allows us to observe the impact of incorporating the concept of compression into VI-SLAM.

The method is validated using a flight dataset recorded from a UAV platform [90]. The system incorporates data from an IMU, a GPS receiver, and a camera installed in a downlooking configuration. On-ground artificial visual landmarks are strategically placed, and

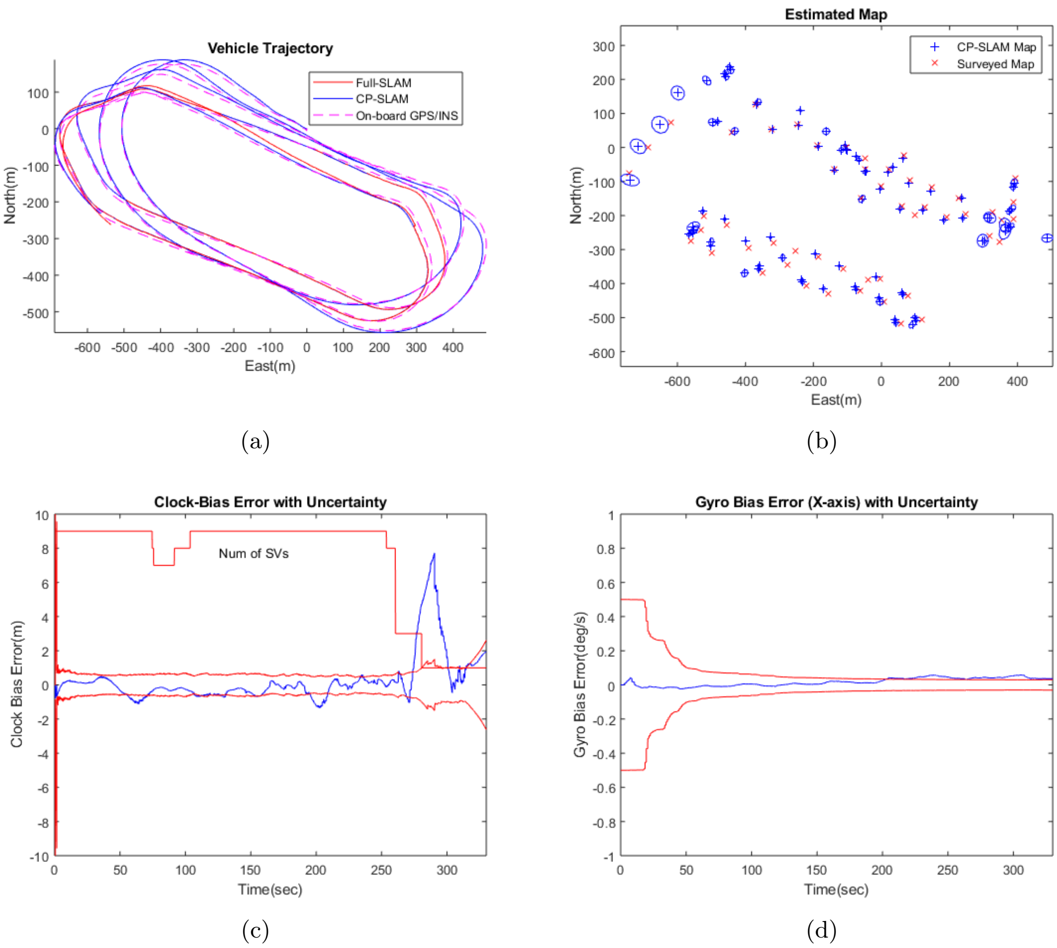


FIGURE 3.2: The result of Compressed Pseudo-SLAM [2]: (a) The estimated vehicle trajectory, (b) map with uncertainty, (c) receiver clock-bias error with uncertainty, and (d) x-gyro bias error with uncertainty. The CP-SLAM trajectory is compared with the full-SLAM (with no compression) and the on-board loosely-coupled GPS/INS solution, showing consistent performance. The receiver clock-bias error shows large errors when the number of SVs drops to 3 and 1. However, thanks to the SLAM aiding, the gyro bias error is constrained adequately.

their positions are surveyed using a real-time-kinematic GPS receiver for reference. As depicted in Figure 3.2(a) and Figure 3.2(b), the estimated trajectory of CP-SLAM closely resembles that of the full EKF SLAM. Furthermore, the estimated map and its associated uncertainty align well with the actual surveyed map positions. In Figure 3.2(c), receiver clock bias error is noticeable as it results in drifts when the number of Satellite Vehicles (SVs) drops to 3. However, the gyro bias error (Figure 3.2(d)), particularly along the x-axis, is still constrained adequately, indicating the robustness of the SLAM system. The total number of landmarks in the system amounts to 85, but the number of local landmarks

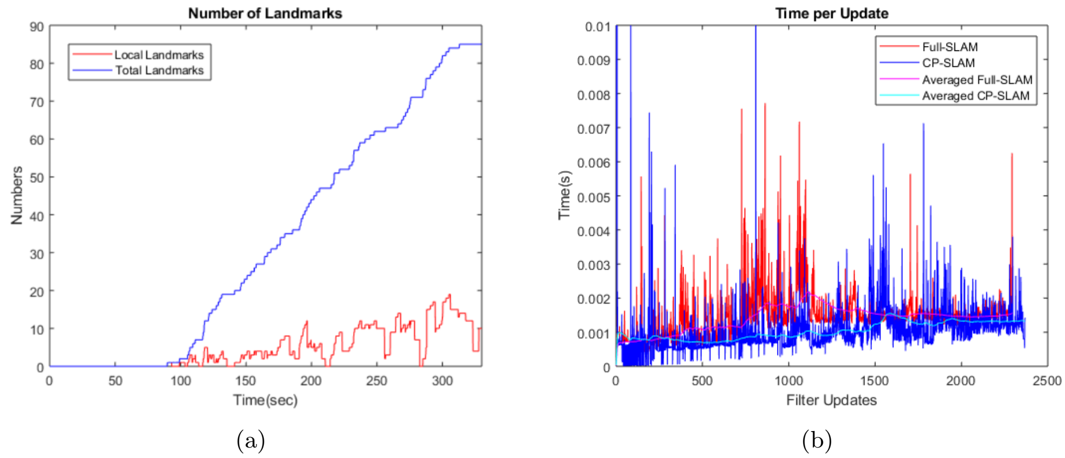


Figure 3.3: Computational time result of Compressed Pseudo-SLAM [2]. (a) The comparison of the total number of landmarks registered (in blue) and the number of local landmarks in CP-SLAM (in red), and (b) the comparison of the update time of the Full-SLAM (in red) and CP-SLAM (in blue).

consistently remains below 20, as depicted in Figure 3.3(a). Regarding computational complexity, occasional peaks are observed during the local-to-global updates, primarily influenced by the association of additional data and the sorting process associated with map transitions. Nevertheless, these results confirm the effectiveness of the compressed filtering approach, demonstrating its suitability for real-time processing, with an average processing time of just 1.5 milliseconds as in Figure 3.3(b).

It is important to note that in this work, the validity was restricted to the downward-looking camera configuration, as the camera field-of-view naturally defines the boundary of the local map. Additionally, in restricted environments, the reliability of GNSS can be compromised. Therefore, a robust VI-SLAM system without using the GNSS is imperative to facilitate diverse applications and enhance overall reliability.

3.1.2 MSCKF

MSCKF [4] is a classic VI-SLAM algorithm based on EKF. The state of MSCKF, denoted as an active state, $\mathcal{X}_A$, includes the IMU state at time k represented by $\mathcal{P}_{I_k}$, and sliding windows, $\mathcal{X}_{C_k} = \begin{bmatrix} \mathcal{P}_{C_{k-M}}^\top & \cdots & \mathcal{P}_{C_{k-1}}^\top \end{bmatrix}^\top$, containing the states of the past M camera poses. The structure is as follows:

$$\mathcal{X}_{A_k} = \begin{bmatrix} \mathcal{P}_{I_k}^\top & \mathcal{X}_{C_k}^\top \end{bmatrix}^\top . \tag{3.3}$$

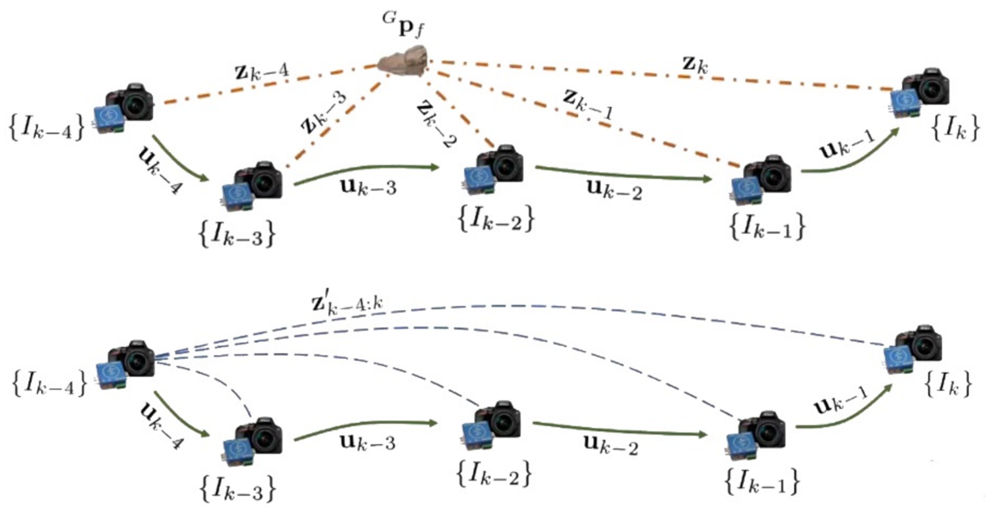


FIGURE 3.4: The geometric constraints in MSCKF are expressed without incorporating features into the state vector, utilizing the null-space technique [3, 4].

Unlike feature-based SLAM, MSCKF provides localization information using multiple visual feature measurements without including the 3D feature positions in the filter state vector, as illustrated in Figure 3.4. This strategy ensures linear computational complexity with respect to the number of features, employing the null-space technique.

The null-space technique demonstrates its ability to modify the general nonlinear residual form, as defined in Equation (2.12), to align with the requirements of MSCKF for the execution of a general EKF update. Upon linearizing around the estimated poses and feature positions within the MSCKF framework, the resulting expression for the residual takes the following form:

$$\mathbf{e}_f = \mathbf{H}_x \widetilde{\mathcal{X}}_{A_{k|k-1}} + \mathbf{H}_f \widetilde{\mathcal{F}}_{k|k-1} + \mathbf{n}_{f_k}, \tag{3.4}$$

where, the measurement noise, $\mathbf{n}_f$, is characterized as white Gaussian noise with covariance $\mathbf{R}$, and $\mathcal{F}$ represents the 3D position of the features. $\mathbf{H}_x$ and $\mathbf{H}_f$ represent the Jacobians of the measurement with respect to the state and feature position, respectively. Additionally, $\widetilde{\mathcal{X}}_{A_{k|k-1}}$ and $\widetilde{\mathcal{F}}_{k|k-1}$ denote the differences between the true and estimated values of the state and feature position. However, in the MSCKF context, the feature position is not included in the state vector. Therefore, a standard EKF update cannot be performed with Equation (3.4) by simply ignoring the feature position, as this is unfeasible due to the correlation between $\mathcal{X}_A$ and $\mathcal{F}$.

To overcome this challenge, $\mathbf{e}_f$ is projected to the left nullspace of $\mathbf{H}_f$, transforming it into a residual model independent of the feature's position:

$$\mathbf{N}^\top \mathbf{e}_f = \mathbf{N}^\top \mathbf{H}_x \widetilde{\mathcal{X}}_{A_{k|k-1}} + \mathbf{N}^\top \mathbf{H}_f \widetilde{\mathcal{F}}_{k|k-1} + \mathbf{N}^\top \mathbf{n}_{f_k}, \tag{3.5}$$

$$\mathbf{e}'_f = \mathbf{H}'_x \widetilde{\mathcal{X}}_{A_{k|k-1}} + \mathbf{n}'_{f_k}, \quad \left(\mathbf{N}^\top \mathbf{H}_f = 0\right). \tag{3.6}$$

Here, $\mathbf{n}'_{f_k}$ is white Gaussian noise with covariance $\mathbf{R}'_k = \mathbf{N}^\top \mathbf{R}_k \mathbf{N}$. This approach enables updating as a general EKF without requiring the feature position to be part of the state vector.

3.1.3 Including Loop-Closure

The standard MSCKF lacks consideration for loop-closures within its algorithm, posing a limitation in handling extended trajectories. Effectively managing the accumulation of drift over prolonged trajectories is imperative to uphold the precision of trajectory estimation. Drift may arise from inherent uncertainties and errors in sensor measurements, leading to deviations from the actual trajectory. To address this challenge, it becomes essential to integrate loop-closure mechanisms. loop-closure entails the identification and closure of loops in the trajectory by recognizing previously visited locations. This integration empowers the system to detect and rectify accumulated errors through loop-closure mechanisms.

Hence, in this chapter, the inclusion of key-frame poses, denoted as $\mathcal{X}_{S_k}$, into the state vector plays a significant role in enhancing this process:

$$\mathcal{X} = \begin{bmatrix} \mathcal{X}_{A_k}^\top & \mathcal{X}_{S_k}^\top \end{bmatrix}^\top = \begin{bmatrix} \mathbf{P}_{I_k}^\top & \mathcal{X}_{C_k}^\top & \mathcal{X}_{S_k}^\top \end{bmatrix}^\top = \begin{bmatrix} \mathcal{P}_{I_k}^\top & \mathcal{X}_{C_k}^\top & \mathcal{P}_{S_1}^\top & \cdots & \mathcal{P}_{S_M}^\top \end{bmatrix}^\top. \tag{3.7}$$

Additionally, the covariance matrix is extended to accommodate these key-frame poses:

$$\mathbf{P} = \left[\begin{array}{c|c} \mathbf{P}_{AA} & \mathbf{P}_{AS} \\ \hline \mathbf{P}_{SA} & \mathbf{P}_{SS} \end{array}\right] = \left[\begin{array}{cc|c} \mathbf{P}_{II} & \mathbf{P}_{IC} & \multirow{2}{*}{$\mathbf{P}_{AS}$} \\ \mathbf{P}_{CI} & \mathbf{P}_{CC} & \\ \hline \multicolumn{2}{c|}{\mathbf{P}_{SA}} & \mathbf{P}_{SS} \end{array}\right]. \tag{3.8}$$

In the MSCKF framework, the growth rate of the state vector size is considerably slower than when features are added to the state vector. Despite this, the accumulation of keyframes along the trajectory leads to an increasing number of states for estimation, potentially challenging the real-time performance.

In Schmidt-MSCKF [6], the adapted concept from [63] efficiently addresses the issues of unbounded localization error and computational cost by treating the key-frames as static. This approach leads to linear growth in computational complexity. However, despite the reduction in computational cost, the strategy of treating the key-frame states as a 'nuisance' throughout the trajectory introduces a significant loss of information that cannot be overlooked.

Balancing computational requirements and accuracy is a critical consideration for real-time SLAM systems. In response to these challenges, I introduce the Comp-MSCKF, a variant that incorporates loop-closure to manage the trade-off between computational efficiency and information loss.

3.2 Compressed Multi-State Constraint Kalman Filter

In this section, **Comp-MSCKF is presented, a method that incorporates loop-closure [91].** Within Comp-MSCKF, the key-frame states $\mathcal{X}_S$ (in Equation (3.7)), which are continuously integrated into the state vector for loop-closure, are further categorized. $\mathcal{X}_S$ are divided into states within the local boundary, denoted as $\mathcal{X}_{S_L}$, and states situated outside the local boundary, represented as $\mathcal{X}_{S_G}$.

$$\mathcal{X}_S = \begin{bmatrix} \mathcal{X}_{S_L}^\top & \mathcal{X}_{S_G}^\top \end{bmatrix}^\top . \tag{3.9}$$

Subsequently, the state vector of Comp-MSCKF and the corresponding covariance matrix in the local and global map are now structured as follows:

$$\mathcal{X} = \left[\begin{array}{c} \mathcal{X}_L \\ \hline \mathcal{X}_G \end{array}\right] = \left[\begin{array}{c} \mathcal{X}_A \\ \mathcal{X}_{S_L} \\ \hline \mathcal{X}_{S_G} \end{array}\right], \tag{3.10}$$

$$\mathbf{P} = \left[\begin{array}{c|c} \mathbf{P}_{LL} & \mathbf{P}_{LG} \\ \hline \mathbf{P}_{GL} & \mathbf{P}_{GG} \end{array}\right] = \left[\begin{array}{cc|c} \mathbf{P}_{AA} & \mathbf{P}_{AS_L} & \mathbf{P}_{LG} \\ \mathbf{P}_{S_LA} & \mathbf{P}_{S_LS_L} & \\ \hline \multicolumn{2}{c|}{\mathbf{P}_{GL}} & \mathbf{P}_{GG} \end{array}\right]. \tag{3.11}$$

Here, the active state, $\mathcal{X}_A$, mentioned in Equation (3.3) is structured as follows:

$$\mathcal{X}_{A_k} = \begin{bmatrix} \mathcal{P}_{I_k}^\top & \mathcal{X}_{C_k}^\top \end{bmatrix}^\top = \begin{bmatrix} \mathcal{P}_{I_k}^\top & {}^{C_{k-M}}_{W}\bar{\mathbf{q}}^\top & {}^{W}\mathbf{t}_{C_{k-M}}^\top & \cdots & {}^{C_{k-1}}_{W}\bar{\mathbf{q}}^\top & {}^{W}\mathbf{t}_{C_{k-1}}^\top \end{bmatrix}^\top . \tag{3.12}$$

The representation of $\mathcal{P}_{I_k}$ is detailed as:

$$\mathcal{P}_{I_k} = \left[\begin{array}{cccc} {}^{I_k}_{W}\bar{\mathbf{q}}^\top & \mathbf{b}_{\omega_k}^\top & \mathbf{b}_{v_k}^\top & {}^{W}\mathbf{t}_{I_k}^\top \end{array} \right]^\top . \tag{3.13}$$

In this representation, ${}^{I_k}_{W}\bar{\mathbf{q}}$ denotes the unit quaternion that describes the rotation between the world frame, {W}, and the IMU frame, {I}. $\mathbf{b}_w$ and $\mathbf{b}_v$ represent the biases associated with gyro and velocity measurements, respectively. ${}^{W}\mathbf{t}_{I_k}$ signifies the IMU position relative to frame {W}. The camera rotation, represented as ${}^{C}_{W}\bar{\mathbf{q}}$, and the camera position, denoted as ${}^{W}\mathbf{t}_C$, are determined using the extrinsic matrix that relates the IMU frame, {I}, to the camera frame, {C}. This is achieved through the following equations:

$${}^{C}_{W}\bar{\mathbf{q}} = {}^{C}_{I}\bar{\mathbf{q}} \otimes {}^{I}_{W}\bar{\mathbf{q}}, \tag{3.14}$$

$${}^{W}\mathbf{t}_C = {}^{W}\mathbf{t}_I + \mathbf{R}_I^{W\,I}\mathbf{t}_C, \tag{3.15}$$

where $\otimes$ represents the quaternion multiplication and $\mathbf{R}_I^W$ is the rotation matrix from {I} to {W}. Each key-frame state, $\mathcal{P}_{S_i}$, is defined as:

$$\mathcal{P}_{S_i} = \left[\begin{array}{cc} {}^{C_i}_{W}\bar{\mathbf{q}}^\top & {}^{W}\mathbf{t}_{C_i}^\top \end{array} \right]^\top . \tag{3.16}$$

3.2.1 Propagation

In the propagation step, the estimated state vector and its associated covariance are continually propagated as they evolve with incoming IMU measurements [5]. In this chapter, the gravity-corrected linear velocity, $\mathbf{v}_m$, and angular velocity, $\boldsymbol{\omega}_m$, are considered as IMU measurements. The "Starry Night" dataset [5] in Section 3.3.2 provides only the gravity-corrected linear velocities. However, the "KITTI" dataset [7] used in Section 3.3.3 provides both gravity-corrected linear velocities and raw linear acceleration. To maintain consistency with the previous dataset, only gravity-corrected linear velocities are utilized.

Unlike the prediction step outlined in Section 2.1.1.1, the motion model equations are obtained by discretizing the continuous-time IMU system model [4]. The following continuous-time motion model describes the evolution of the estimated IMU state $\hat{\mathcal{P}}_I$ over time:

$$\begin{gathered} {}^{I}_{W}\dot{\hat{\mathbf{q}}} = \frac{1}{2}\Omega({}^{I}\hat{\boldsymbol{\omega}}){}^{I}_{W}\hat{\bar{\mathbf{q}}}, \quad \dot{\hat{\mathbf{b}}}_\omega = \mathbf{0}_{3\times1}, \\ \dot{\hat{\mathbf{b}}}_v = \mathbf{0}_{3\times1}, \quad {}^{W}\dot{\hat{\mathbf{t}}}_I = \hat{\mathbf{R}}_I^{W\,I}\hat{\mathbf{v}}. \end{gathered} \tag{3.17}$$

The rotational velocity, $\hat{\boldsymbol{\omega}}$, and linear velocity, $\hat{\mathbf{v}}$, are both expressed in the IMU frame. These can be computed using the IMU's measurements of velocity, $\mathbf{v}_m$, and gyro, $\boldsymbol{\omega}_m$, as follows:

$$ {}^{I}\hat{\mathbf{v}} = {}^{I}\mathbf{v}_m - \hat{\mathbf{b}}_v, \quad {}^{I}\hat{\boldsymbol{\omega}} = {}^{I}\boldsymbol{\omega}_m - \hat{\mathbf{b}}_\omega. \tag{3.18} $$

The linearized continuous-time model of the IMU error state can be expressed as:

$$ \dot{\widetilde{\mathcal{P}}}_I = \mathbf{F}\widetilde{\mathcal{P}}_I + \mathbf{G}\mathbf{n}_I, \tag{3.19} $$

where the error-state, $\widetilde{\mathcal{P}}_I$, is defined as:

$$ \widetilde{\mathcal{P}}_I = \begin{bmatrix} \delta\boldsymbol{\theta}_I^\top & \widetilde{\mathbf{b}}_\omega^\top & \widetilde{\mathbf{b}}_v^\top & {}^{W}\widetilde{\mathbf{t}}_I^\top \end{bmatrix}^\top. \tag{3.20} $$

Here, $\mathbf{n}_I = \begin{bmatrix} \mathbf{n}_\omega^\top & \mathbf{n}_{b_\omega}^\top & \mathbf{n}_v^\top & \mathbf{n}_{b_v}^\top \end{bmatrix}^\top$ represents the IMU process noise with covariance matrix $\mathbf{Q}$. While the error-state of position and biases can be directly calculated by the difference between the true and estimated values, the error quaternion, $\delta\bar{\mathbf{q}}$, is defined as:

$$ \delta\bar{\mathbf{q}} \simeq \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}^T & 1 \end{bmatrix}^T. \tag{3.21} $$

This is determined by the relation $\bar{\mathbf{q}} = \delta\bar{\mathbf{q}} \otimes \hat{\bar{\mathbf{q}}}$, and since it describes the small rotation, where only $\delta\boldsymbol{\theta}$ is used as a minimal representation. The Jacobians $\mathbf{F}$ and $\mathbf{G}$ are given by

$$ \mathbf{F} = \begin{bmatrix} -{}^{I}\hat{\boldsymbol{\omega}}^\wedge & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\hat{\mathbf{R}}_I^{W\,I}\hat{\mathbf{v}}^\wedge & \mathbf{0}_{3\times3} & -\hat{\mathbf{R}}_I^W & \mathbf{0}_{3\times3} \end{bmatrix}, \tag{3.22} $$

$$ \mathbf{G} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & 0_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\hat{\mathbf{R}}_I^W & \mathbf{0}_{3\times3} \end{bmatrix}. \tag{3.23} $$

To account for discrete time intervals, the differential model needs to be integrated into differences equations. As described in [5], forward Euler integration is used to propagate the motion model. The covariance, $\mathbf{P}_{AA}$, can be propagated as follows:

$$ \mathbf{P}_{AA_{k|k-1}} = \begin{bmatrix} \boldsymbol{\Gamma}_{k-1}\mathbf{P}_{II_{k-1|k-1}}\boldsymbol{\Gamma}_{k-1}^\top + \mathbf{G}\mathbf{Q}_{k-1}\mathbf{G}^\top\Delta t & \boldsymbol{\Gamma}_{k-1}\mathbf{P}_{IC_{k-1|k-1}} \\ \mathbf{P}_{IC_{k-1|k-1}}^\top\boldsymbol{\Gamma}_{k-1}^\top & \mathbf{P}_{CC_{k-1|k-1}} \end{bmatrix}. \tag{3.24} $$

Here, $\mathbf{P}_{II}$ represents the covariance matrix of the evolving IMU state, $\mathbf{P}_{CC}$ is the covariance matrix of the camera pose estimates in sliding window, and $\mathbf{P}_{IC}$ denotes the correlation between the IMU state and the states in the sliding window. The state transition matrix, $\mathbf{\Gamma}_{k-1}$, is given by:

$$\mathbf{\Gamma}_{k-1} = \mathbf{I} + \mathbf{F}\Delta t, \tag{3.25}$$

where Δt is the IMU sampling period. Subsequently, the covariance matrix for the entire state vector in Comp-MSCKF is expressed as:

$$\mathbf{P}_{k|k-1} = \left[\begin{array}{cc|c} \mathbf{P}_{AA_{k|k-1}} & \mathbf{\Gamma}_{k-1}\mathbf{P}_{AS_{L_{k-1|k-1}}} & \\ \mathbf{P}^{\top}_{AS_{L_{k-1|k-1}}}\mathbf{\Gamma}^{\top}_{k-1} & \mathbf{P}_{S_L S_{L_{k-1|k-1}}} & \mathbf{\Gamma}_{k-1}\mathbf{P}_{LG_{k-1|k-1}} \\ \hline \multicolumn{2}{c|}{\mathbf{P}^{\top}_{LG_{k-1|k-1}}\mathbf{\Gamma}^{\top}_{k-1}} & \mathbf{P}_{GG_{k-1|k-1}} \end{array}\right]. \tag{3.26}$$

It is evident that the correlation between the local and global states can be compressed as until the new image is detected:

$$\mathbf{P}_{LG}(k) = \left(\prod_{i=1}^{k} \mathbf{\Gamma}_i\right) \mathbf{P}_{LG}(0). \tag{3.27}$$

3.2.2 Update

In the update step, the observations are first used to update the local state, and the correlation is accumulated. As discussed in Section 3.1.2, following the update step allows achieving Equation (3.6). The measurement Jacobian, $\mathbf{H}'_x$, exhibits sparsity and solely contains values corresponding to the local state, represented as $\mathbf{H}'_x = \left[\begin{array}{cc} \mathbf{H}_{L_x} & \mathbf{0}_{G_x} \end{array}\right]$. Consequently, the residual measurement can be expressed as follows:

$$\mathbf{e}'_f \simeq \mathbf{H}_{L_x}\widetilde{\mathcal{X}}_{L_{k|k-1}} + \mathbf{n}'_{f_k}. \tag{3.28}$$

Utilizing this model, the update process of the state estimate unfolds as in Equation (2.15):

$$\hat{\mathcal{X}}_{L_{k|k}} = \hat{\mathcal{X}}_{L_{k|k-1}} + \mathbf{K}_{L_k}\mathbf{e}'_f, \tag{3.29}$$

$$\hat{\mathcal{X}}_{G_{k|k}} = \hat{\mathcal{X}}_{G_{k|k-1}} + \mathbf{K}_{G_k}\mathbf{e}'_f, \tag{3.30}$$

where the Kalman gain, $\mathbf{K}$, can be computed as:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}'_x\mathbf{S}_k^{-1} = \begin{bmatrix} \mathbf{P}_{LL_{k|k-1}}\mathbf{H}_{L_x}^{\top}\mathbf{S}_k^{-1} \\ \mathbf{P}_{GL_{k|k-1}}\mathbf{H}_{L_x}^{\top}\mathbf{S}_k^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{L_k} \\ \mathbf{K}_{G_k} \end{bmatrix}. \tag{3.31}$$

Here, $\mathbf{S}_k = \mathbf{H}'_x\mathbf{P}_{k|k-1}\mathbf{H}'^{\top}_x + \mathbf{R}'_k = \mathbf{H}_{L_x}\mathbf{P}_{LL_{k|k-1}}\mathbf{H}_{L_x}^{\top} + \mathbf{R}'_k$, and therefore the form of the updated covariance matrix is represented as:

$$\begin{aligned}
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^T \\
&= \mathbf{P}_{k|k-1} - \begin{bmatrix} \mathbf{P}_{LL_{k|k-1}}\mathbf{H}_{L_x}^{T}\mathbf{S}_k^{-1} \\ \mathbf{P}_{GL_{k|k-1}}\mathbf{H}_{L_x}^{T}\mathbf{S}_k^{-1} \end{bmatrix} \mathbf{S}_k \begin{bmatrix} \mathbf{P}_{LL_{k|k-1}}\mathbf{H}_{L_x}^{T}\mathbf{S}_k^{-1} \\ \mathbf{P}_{GL_{k|k-1}}\mathbf{H}_{L_x}^{T}\mathbf{S}_k^{-1} \end{bmatrix}^T \\
&= \mathbf{P}_{k|k-1} -
\end{aligned}$$

$$\begin{bmatrix} \mathbf{P}_{LL_{k|k-1}}\left(\mathbf{H}_{L_x}^{T}\mathbf{S}_k^{-1}\mathbf{H}_{L_x}\right)\mathbf{P}_{LL_{k|k-1}} & \underbrace{(\mathbf{P}_{LL_{k|k-1}}\left(\mathbf{H}_{L_x}^{T}\mathbf{S}_k^{-1}\mathbf{H}_{L_x}\right))}_{\boldsymbol{\Upsilon}}\mathbf{P}_{LG_{k|k-1}} \\ \left(\left(\mathbf{P}_{LL_{k|k-1}}\left(\mathbf{H}_{L_x}^{T}\mathbf{S}_k^{-1}\mathbf{H}_{L_x}\right)\mathbf{P}_{LG_{k|k-1}}\right)^T & \mathbf{P}_{GL_{k|k-1}}\underbrace{(\mathbf{H}_{L_x}^{T}\mathbf{S}_k^{-1}\mathbf{H}_{L_x})}_{\boldsymbol{\Psi}}\mathbf{P}_{LG_{k|k-1}} \end{bmatrix}. \tag{3.32}$$

Until the new local boundary is defined, the calculation of $\hat{\mathcal{X}}_G$, $\mathbf{P}_{LG}$, and $\mathbf{P}_{GG}$ adopts a compressed approach to efficiently manage computational complexity while continuously updating $\hat{\mathcal{X}}_L$ and $\mathbf{P}_{LL}$. The reduction of correlation and global state terms can be re-written as follows:

$$\mathbf{P}_{LG_{k|k}} = \mathbf{P}_{LG_{k|k-1}} - \boldsymbol{\Upsilon}\mathbf{P}_{LG_{k|k-1}} = \boldsymbol{\Phi}\mathbf{P}_{LG_{k|k-1}}, \tag{3.33}$$

$$\mathbf{P}_{GG_{k|k}} = \mathbf{P}_{GG_{k|k-1}} - \left(\mathbf{P}_{GL_{k|k-1}}\boldsymbol{\Psi}\mathbf{P}_{LG_{k|k-1}}\right), \tag{3.34}$$

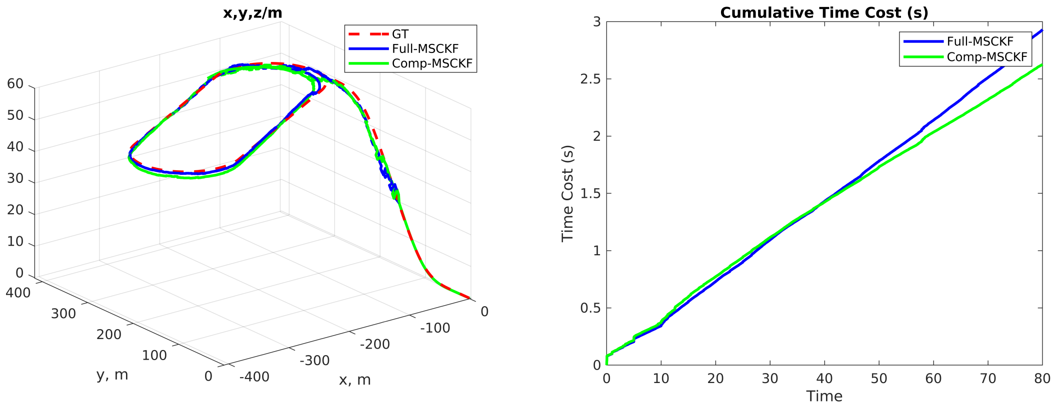
$$\hat{\mathcal{X}}_{G_{k|k}} = \hat{\mathcal{X}}_{G_{k|k-1}} + \left(\mathbf{P}_{GL_{k|k-1}}\mathbf{H}_{L_x}^{T}\mathbf{S}_k^{-1}\mathbf{e}'_f\right). \tag{3.35}$$

As a result, the accumulated form of the estimated global state $\hat{\mathcal{X}}_G$, correlation term $\mathbf{P}_{LG}$, and $\mathbf{P}_{GG}$ can be expressed as follows:

$$\mathbf{P}_{LG}(k) = \left(\prod \boldsymbol{\Phi}\right)\mathbf{P}_{LG}(0) = \boldsymbol{\Phi}(k,0)\mathbf{P}_{LG}(0), \tag{3.36}$$

$$\mathbf{P}_{GG}(k) = \mathbf{P}_{GG}(0) - \mathbf{P}_{GL}(0)\left(\sum \boldsymbol{\Phi}(k,0)^T\boldsymbol{\Psi}\boldsymbol{\Phi}(k,0)\right)\mathbf{P}_{LG}(0), \tag{3.37}$$

$$\hat{\mathcal{X}}_G(k) = \hat{\mathcal{X}}_G(0) + \mathbf{P}_{GL}(0)\left(\sum \boldsymbol{\Phi}(k,0)^T\mathbf{H}_L^{T}\mathbf{S}^{-1}\mathbf{e}'_f\right). \tag{3.38}$$



(a) (b)

FIGURE 3.5: The result of Comp-MSCKF in MATLAB simulator comparing with Full-MSCKF (Full-MSCKF involves loop-closure and updates entire state vector and covariance matrix upon each incoming data): (a) Final trajectory (b) Cumulative time cost.

Using this compressed correlation term, the global map and covariance can be recovered at a much lower rate whenever the local map boundary changes.

3.3 Performance Evaluations of Comp-MSCKF

This section assesses the performance of Comp-MSCKF through both simulation and real-world datasets. In the simulation, the convergence of key-frame states is evaluated, focusing on the uncertainty of these states within the context of Comp-MSCKF. Additionally, experiments are conducted using the "Starry Night" and "KITTI" datasets, comparing the performance of Comp-MSCKF with that of standard MSCKF [4] and Schmidt-MSCKF [6]. All experiments are performed in MATLAB.

3.3.1 Simulation

For the experimental validation of the proposed method, a high-fidelity MATLAB simulator [14] was used, as illustrated in Figure 3.1. This simulator, named CP-SLAM, is designed for comprehensive all-source navigation. To assess Comp-MSCKF, only the visual-inertial sensors were utilized. The generation of sensor data, encompassing IMU readings at a rate of 100Hz and vision data at 20Hz, accurately follows a simulated trajectory employing realistic sensor models to replicate real-world conditions. Detailed IMU

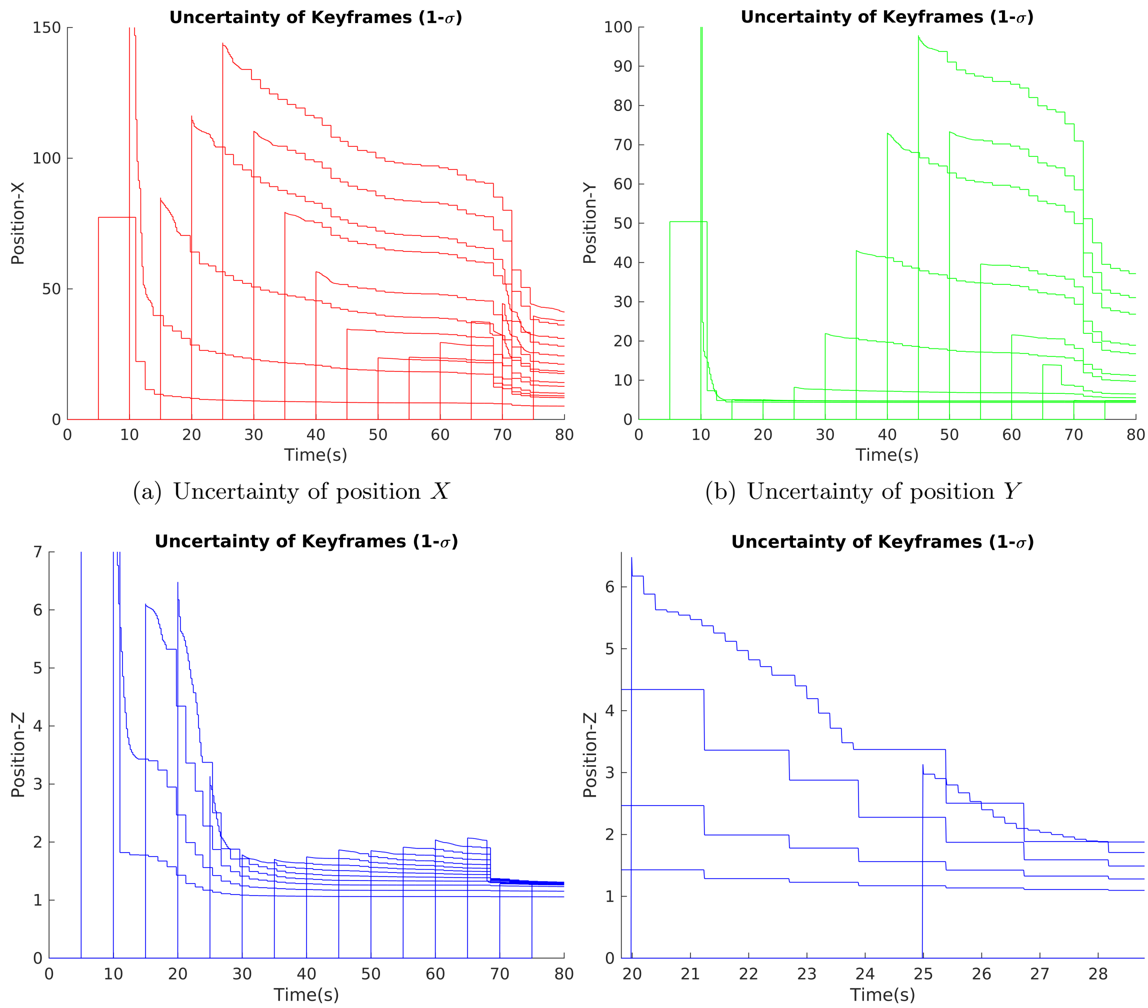
Table 3.1: Simulation setup parameters for the IMU and Camera [14].

Sensor	Type	Unit	Specification
IMU	Sampling rate	Hz	100
	Accel bias	mg	2
	Gyro bias	°/h	100
	Accel bias stability	g, 1σ	0.02
	Gyro bias stability	°/h, 1σ	100
	Accel bias correlation time	s	300
	Gyro bias correlation time	s	300
Camera	Frame rate	Hz	20
	FOV angle	°	30
	Range noise	m	1
	Bearing noise	°	1
	Elevation noise	°	1.5

and camera parameters used in the simulation are provided in Table3.1. To demonstrate the effectiveness of Comp-MSCKF, the simulator is adapted to enable the integration of key-frames into the state vector. The frequency of this integration is linked to the camera's field-of-view and the vehicle's speed. For simplicity, a 5-second interval is employed in this work to achieve a balanced coverage of the trajectory.

The outcomes of the proposed method, Comp-MSCKF, are presented in Figure 3.5(a), showcasing the trajectory results in comparison to Full-MSCKF. The Full-MSCKF updates the entire state vector along with its covariance matrix upon each data input, incorporating loop-closure without compressing the data. The Root Mean Square Error (RMSE) of Comp-MSCKF is 8.655m, higher than that of Full-MSCKF, which is 4.519m. However, as illustrated in Figure 3.5(b), Comp-MSCKF provides a temporal perspective on computational efficiency over Full-MSCKF.

Figure 3.6 provides insights into the uncertainty evolution of keyframes, featuring a total of 14 registered keyframes. The compressed update strategy is strategically applied as the vehicle approaches the local map boundary, leading to the re-centering of the local map at the current vehicle location. In the simulation, the local map is re-centered at approximately $1.5 - 2$ seconds. A notable loop-closure event occurs around 70 seconds, during which the uncertainty of keyframes significantly decreases. Figure 3.6(d) provides an enhanced view of key-frame number 4, illustrating the effects of both sensor updates and compressed updates. Around 20 seconds, the fourth key-frame is incorporated into the state vector and consistently updated based on vision information. By approximately 24 seconds, it transitions to a global state as it surpasses the local boundary. From that point, it compresses all incoming information, and at around 25 seconds, with the boundary change, all the compressed information related to the fourth key-frame is updated.



(a) Uncertainty of position X

(b) Uncertainty of position Y

(c) Uncertainty of position Z

(d) Uncertainty of 4^{th} key-frame

FIGURE 3.6: The evolution of uncertainty in key-frames: A comparison between XYZ and an enhanced view of 4^{th} key-frame, highlighting compressed updates during the simulation.

Subsequently, only global updates occur until the loop-closure event. This comprehensive set of results provides a detailed understanding of the performance, accuracy, and computational efficiency of the Comp-MSCKF.

3.3.2 "Starry Night" Dataset

The "Starry Night" dataset [5] consists of stereo vision and pre-processed IMU readings within an environment featuring static landmarks, as illustrated in Figure 3.7 and Table 3.2. Only the monocular camera data from the left is utilized for this experiment, and

(a) (b)

FIGURE 3.7: Dataset environment of "Starry Night" [5]: (a) The hand-held sensor head used in experiments. (b) Overview of the data collection environment.

TABLE 3.2: Sensor parameters used in "Starry Night" dataset [5].

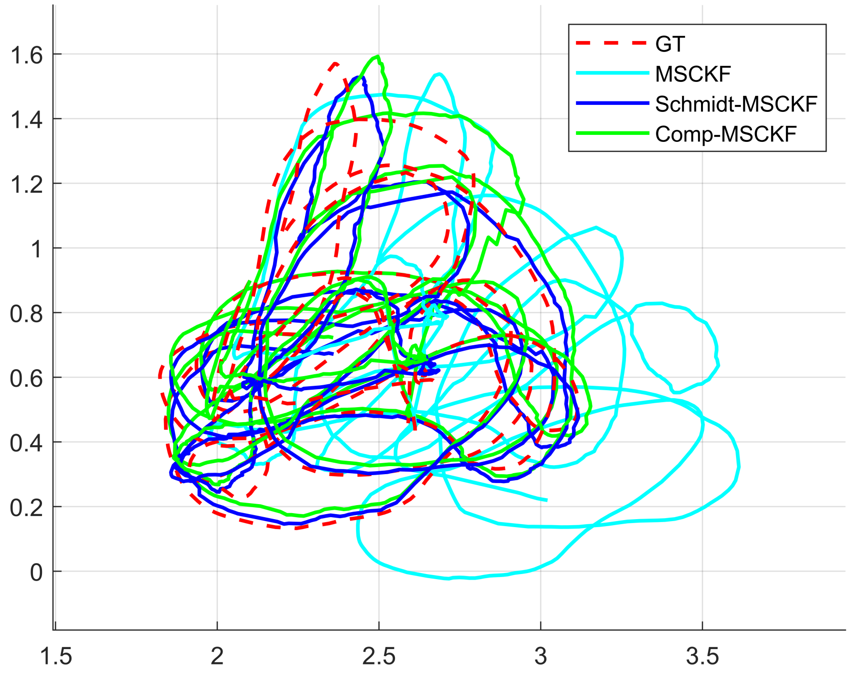
Sensor	Type	Unit	Specification
IMU	Sampling rate	Hz	10
	Gyro measurement noise	$rad/s/\sqrt{s}$	0.2
	Velocity measurement noise	$m/s/\sqrt{s}$	0.2
	Gyro bias random work noise	$rad/s^2/\sqrt{s}$	0.001
	Velocity bias random walk noise	$m/s^2/\sqrt{s}$	0.001
Camera	Frame rate	Hz	10
	Horizontal focal length	pixels	484.4998
	Vertical focal length	pixels	484.4998
	Horizontal optical center	pixels	321.6805
	Vertical optical center	pixels	247.4814

the right camera's information is disregarded. The Vicon motion capture system records sensor head motion and feature positions, serving as the ground-truth.

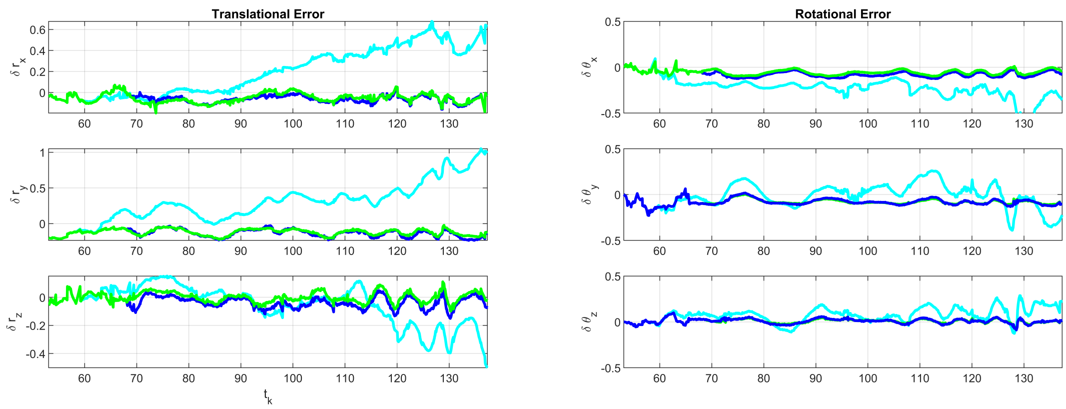
The original dataset observes a set of 20 features, but it has been enhanced by introducing synthetic features distributed with larger spatial extents, resulting in a maximum of 500 features. The modified dataset preserves the IMU data from the original set and introduces zero-mean Gaussian noise to corrupt the synthetic camera measurements [5].

For the experiments, instead of marginalizing cloned camera poses from the sliding window of the active state, specific clones are retained as key-frames in the state vector for loop-closure as similar to [6]. The selection of key-frames can be based on various heuristics. In this work, new key-frames are simply added at fixed time intervals, and all feature IDs are pre-defined for identifying loop-closure candidates in key-frame-based loop closing.

The trajectory estimation results of the proposed Comp-MSCKF are visually presented



(a) The Estimated Trajectories



(b) Translation and Rotational Error

FIGURE 3.8: The trajectory estimations for the "Starry Night" dataset are provided for the proposed Comp-MSCKF, Schmidt-MSCKF [6], and standard MSCKF [4]. Examining the translation error, it is evident that the error of the standard MSCKF increases over time. In contrast, both Comp-MSCKF and Schmidt-MSCKF effectively maintain bounded error throughout the duration.

in Figure 3.8(a), offering a comparative analysis alongside Schmidt-MSCKF and standard MSCKF. Notably, both Comp-MSCKF and Schmidt-MSCKF demonstrate a reasonable degree of proximity to the ground-truth trajectory, highlighting their efficacy in

Table 3.3: Comparison of RMSE values for translation and rotation, alongside the final translation error: Standard MSCKF, Schmidt-MSCKF, and Comp-MSCKF on the "Starry Night" Dataset.

	MSCKF (no loop-closure)	**Schmidt - MSCKF**	**Comp - MSCKF**
Trans.RMSE (m)	0.248	0.091	**0.082**
Rot.RMSE (deg)	0.161	0.070	**0.062**
Final Trans Err (m)	1.279	0.155	**0.123**

accurately estimating the pose by including the key-frames for loop-closure. In contrast, standard MSCKF exhibits a divergence from the ground-truth trajectory, suggesting a susceptibility to cumulative error over time, as evident in Figure 3.8(b). These results affirm the superior accuracy and precision of Comp-MSCKF in trajectory estimation, particularly when compared to standard MSCKF without loop-closure. The inclusion of key-frames for loop-closure proves instrumental in mitigating cumulative errors.

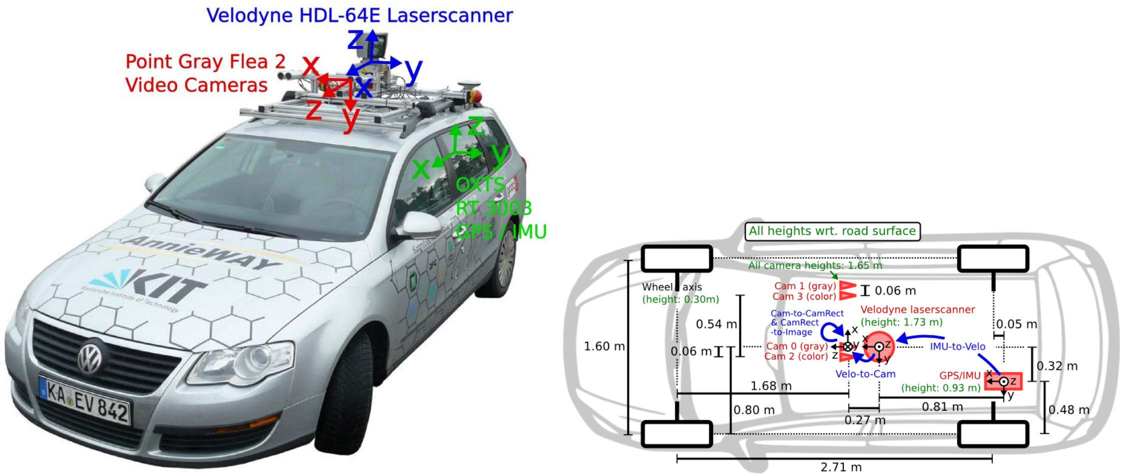
For a more detailed evaluation, Table 3.3 comprehensively compares translational and rotational accuracy metrics among the three methodologies—MSCKF, Comp-MSCKF, and Schmidt-MSCKF. The standard MSCKF, operating without loop-closure, demonstrates the highest errors in translation RMSE, rotational RMSE, and final translation error, with recorded values of 0.248m, 0.161°, and 1.279m, respectively.

In contrast, Comp-MSCKF emerges as the standout performer, showcasing the lowest errors in translation RMSE at 0.082m, rotational RMSE at 0.062°, and final translation error at 0.123m. Schmidt-MSCKF also exhibits favourable performance, with errors slightly higher than Comp-MSCKF, recording values of 0.091m, 0.070°, and 0.155m for translation RMSE, rotational RMSE, and final translation error, respectively.

The integration of key-frames for loop-closure proves instrumental in mitigating cumulative errors, underscoring the superior accuracy and precision of Comp-MSCKF in trajectory estimation when compared to both Schmidt-MSCKF and standard MSCKF. However, it is essential to consider the computational complexity as well. As highlighted in Table 3.4, the computational time for Comp-MSCKF is 328.86 seconds, which is marginally higher than Schmidt-MSCKF at 320.92 seconds. Nevertheless, this difference in computational time can be perceived as reasonable in comparison to the Full-MSCKF, which reports a computational time of 338.90 seconds. The efficiency of Comp-MSCKF in terms of both accuracy and computational time makes it a compelling choice for applications demanding a balance between precision and computational efficiency.

Table 3.4: Total computational time for each method on the “Starry Night” dataset.

	MSCKF (no loop-closure)	Schmidt - MSCKF	Comp - MSCKF	Full - MSCKF
Compute time (sec)	124.566	320.921	328.862	338.897



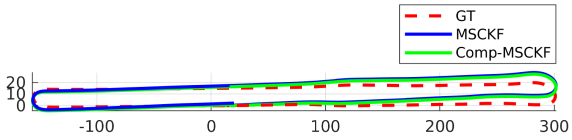
(a) Hardware Platform (b) Sensor Extrinsic setup

Figure 3.9: The sensor setup for collecting the “KITTI” dataset [7].

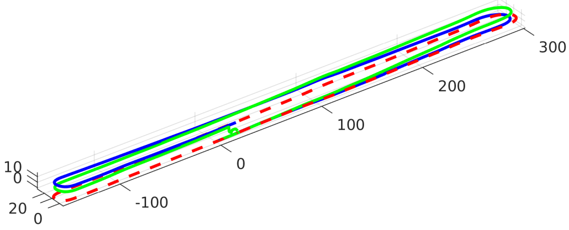
3.3.3 “KITTI” Dataset

The publicly accessible dataset mentioned here was collected by moving platforms, as illustrated in Figure 3.9(a). It stands as a widely used benchmark dataset in the fields of computer vision and robotics. The “KITTI” dataset provides a rich set of data, including camera images, laser scans, high-precision GPS measurements, and IMU. The GPS/IMU system is combined, and the extrinsic setup is depicted in Figure 3.9(b). For each frame, 30 different GPS/IMU values are provided, encompassing geographic coordinates such as altitude, global orientation, velocities, accelerations, angular rates, accuracies, and satellite information. The dataset offers ground-truth trajectory information obtained from high-precision GPS measurements to assess the accuracy and reliability of SLAM algorithms.

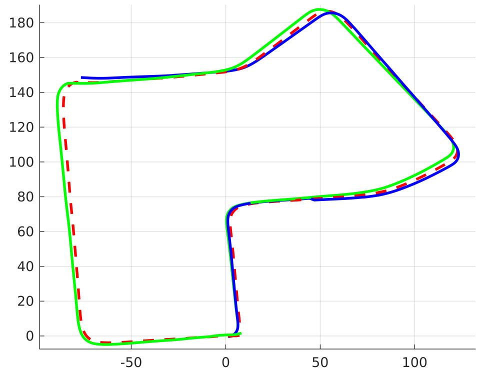
As outlined in Section 3.2.1, the IMU measurement incorporates a pre-processed linear velocity instead of raw linear acceleration. Image processing for feature extraction is conducted using ORB-SLAM3 [12], a state-of-the-art SLAM algorithm. Each extracted feature is assigned a predefined ID, enhancing the understanding and tracking of visual features throughout the experiment.



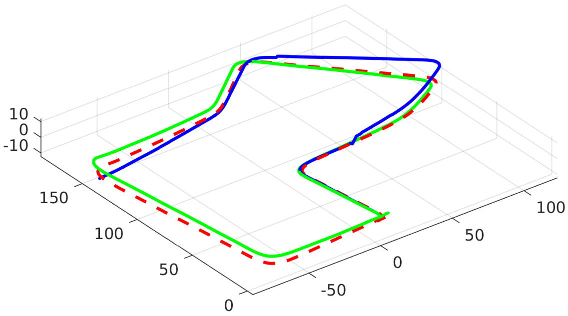
(a) (KITTI-06) Top View



(b) (KITTI-06) Isometric View



(c) (KITTI-07) Top View



(d) (KITTI-07) Isometric View

FIGURE 3.10: Comparative Trajectories of Comp-MSCKF and standard MSCKF on "KITTI" datasets (Both methods failed to close the loop, demonstrating divergence. Notably, MSCKF in sequence 07 exhibited divergence, even in the middle of the trajectory. Trajectories are plotted only until the point of divergence).

The results of the Comp-MSCKF estimation are depicted in Figure 3.10. Schmidt-MSCKF yields very similar outcomes to Comp-MSCKF; hence, only the results of Comp-MSCKF are presented in comparison with the standard MSCKF. Unfortunately, both Schmidt and Comp-MSCKF encounter difficulties in closing the loop. The presented figures extend only until the divergence point in sequences 06 and 07 of the "KITTI" dataset. In the case of the standard MSCKF, it exhibits drift over time and fails to complete sequence 07 of the "KITTI" dataset, diverging in the middle of the trajectory.

To offer a more comprehensive assessment of the proposed method's effectiveness in loop-closure with key-frame states, a dataset that provides the advantage of observing the same features multiple times is necessary. This facilitates more frequent loop-closure updates, which can reveal and address significant drifts in the trajectory. In contrast, the "KITTI" dataset involves collinear motion within a large environment, requiring an extended duration to observe loop-closure features. The extended duration in the "KITTI" dataset poses a challenge when performing loop-closure updates using key-frame states. Over time, the trajectory estimate accumulates drift, and uncertainties may collapse after a loop-closure. Moreover, in the feature extraction process using ORB-SLAM3 [12], features with minimal parallax are deliberately removed. While this is done to improve computational efficiency, it may potentially impact the effectiveness of the proposed Comp-MSCKF, particularly in scenarios where minimal parallax features could contribute to loop-closure.

3.4 Summary

The application of the compressed filtering framework to an MSCKF (Comp-MSCKF) has the advantage of retaining pose key-frames in the state while effectively limiting computational complexity to $\mathcal{O}(N_L^2)$, where N_L represents the number of local key-frames. This approach demonstrates improved performance when compared to both the standard MSCKF and Schmidt-MSCKF.

However, challenges arise when simultaneously dealing with the information of states marginalized within the sliding window and compressed within the local area. This concurrent processing introduces the potential for information loss, significantly impacting the overall accuracy of the system. Distinguishing between local and global information becomes particularly challenging, especially in the context of a monocular camera. Therefore, developing a suitable strategy for compressing data is essential while preserving the fundamental MSCKF framework.

Furthermore, in scenarios where loop closing fails, this consideration becomes crucial, especially given the demonstrated effectiveness of optimization-based methods over filtering-based methods. Optimization methods stand out for their ability to propagate loop-closure data backward along the trajectory estimate. It's noteworthy that, while the Comp-MSCKF offers advantages in terms of computational complexity, it tends to be more sensitive to tuning parameters than optimization-based methods.

Additionally, as observed in the "KITTI" dataset, characterized by larger and longer trajectories, addressing the handling of features with minimal parallax emerges as a critical aspect requiring thoughtful consideration and tailored solutions. In the next chapter (Chapter 4), I present PVI-SLAM, a novel approach founded on PBA, aimed at addressing and overcoming the challenges presented in this chapter.

Chapter 4

Parallax Bundle Adjustment with Inertial Measurement Unit

As highlighted in Chapter 3, filtering-based approaches encountered difficulties when closing loops, especially in larger and longer trajectories. These difficulties were due to the accumulated drift over time, even with the incorporation of key-frames in the state vector for loop-closure. Addressing these challenges, the current chapter shifts its attention toward optimization-based methods. Nonlinear optimization techniques employed in these methods have demonstrated the potential for achieving superior accuracy compared to their filtering-based counterparts. This shift in focus is fueled by the advancements in computer technology.

In this chapter, a novel solution named **PVI-SLAM** is proposed. The method aims to deliver robust, tightly-coupled, optimization-based VI-SLAM by leveraging parallax angle for feature parametrization and pre-integrating IMU measurements in continuous-time.

BA plays a crucial role in the back-end process of modern VI-SLAM systems. The chapter begins by introducing the parallax feature parameterization method for BA. This approach proves effective in addressing challenges related to features observed at minimal parallax angles, particularly in scenarios involving collinear motion.

Building upon the PBA foundation [89], the chapter proposes the integration of IMU data to enhance the accuracy of state estimation and overcome the challenge of recovering the correct metric scale in monocular vision-only systems. The IMU measurements undergo pre-integration using UGPM, and a comparative analysis is conducted with Standard Preintegrated Measurement (PM). In the case of UGPM, the GP method is utilized, providing continuous and non-parametric representations of the system's dynamics. This integration introduces a dynamic dimension to the system, enhancing its robustness and enabling a more accurate representation of the state.

Following this, a novel VI-SLAM system that employs parallax parametrization in the manifold domain (PVI-SLAM) is presented. A compatible error function utilizing the observation ray is implemented to further enhance the robustness of the system. This approach aims to improve the accuracy and reliability of the system, particularly in scenarios with challenging visual conditions or complex motion patterns.

The subsequent sections conduct a robustness analysis of the proposed method through evaluations on publicly available datasets, including "MALAGA", "Starry Night", "EuRoC", and "KITTI". The proposed system's robustness is demonstrated and compared with state-of-the-art methods, highlighting its efficacy in addressing the aforementioned challenges and providing a comprehensive understanding of its performance across diverse scenarios.

4.1 Parallax Feature Parametrization

The modern BA algorithm commonly uses Euclidean XYZ coordinates to represent the locations of features, $\mathbf{f}_j$, in 3D [94–96], as illustrated in Figure 4.1(a):

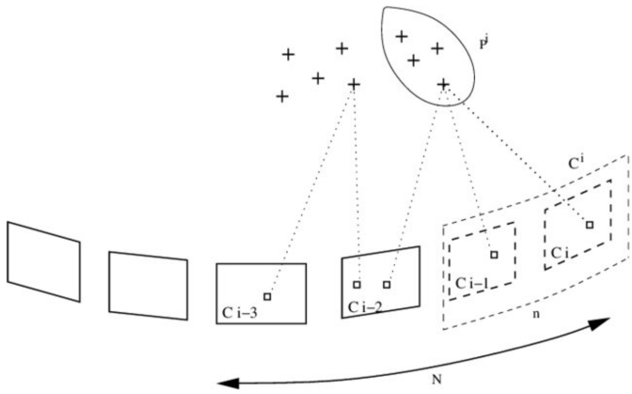
$$\mathbf{f}_j^{XYZ} = [X_j, Y_j, Z_j]^\top . \tag{4.1}$$

An alternative method for parametrizing feature position is the IDP proposed by Civera et al. [93], as depicted in Figure 4.1(b). It was proposed that the inverse depth of the feature can be used in monocular SLAM. IDP is defined relative to the first camera pose that observed the feature as:

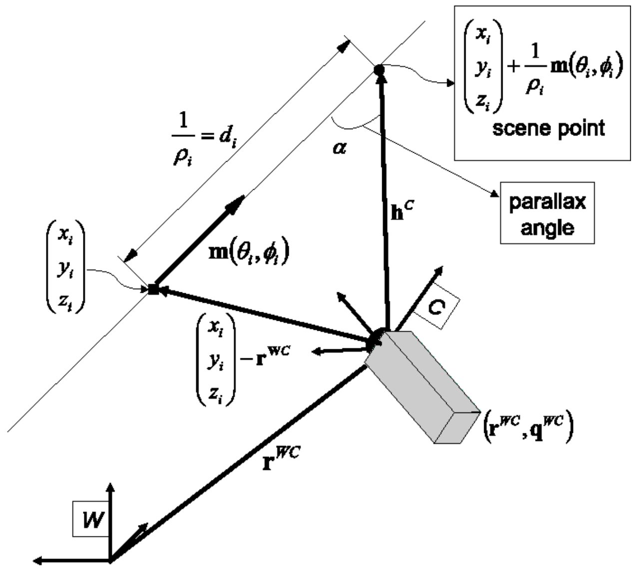
$$\mathbf{f}_j^{IDP} = [x_j, y_j, z_j, \psi_j, \theta_j, \rho_j]^\top , \tag{4.2}$$

where x_j, y_j, and z_j are the camera pose in the first observation of feature $\mathbf{f}_j$, and ψ_j and θ_j represent azimuth and elevation. The point's depth along the ray d_i is encoded by its inverse $\rho_j = 1/d_i$.

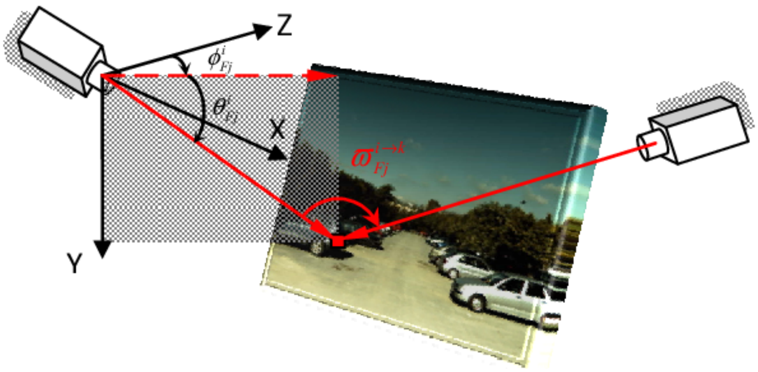
Both the XYZ feature parametrization and IDP prove effective when dealing with features situated at a considerable distance with sufficient parallax angles, as demonstrated in Figure 4.2(a). However, the advantages of IDP become particularly pronounced when



(a) Euclidean XYZ parametrization [92]

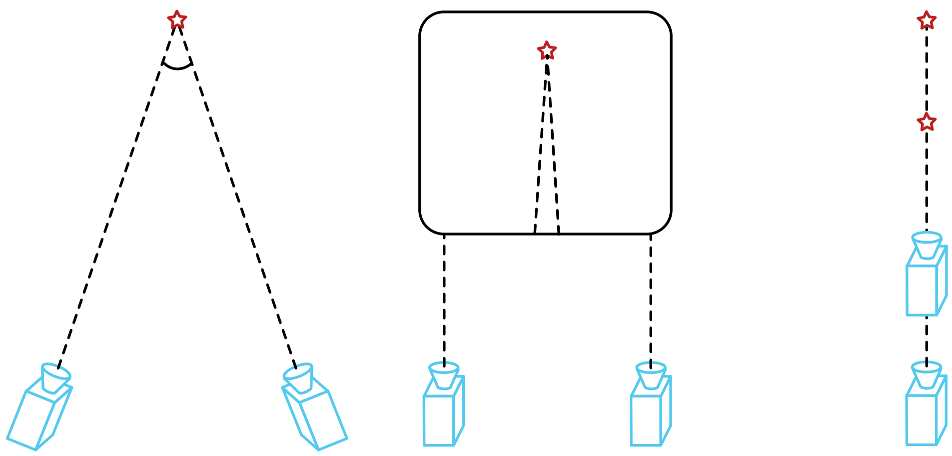


(b) Inverse depth feature parametrization [93]



(c) Parallax feature parametrization [89]

FIGURE 4.1: Different feature parametrization methods.



(a) Depth from parallax (b) Infinity depth (c) No depth information

FIGURE 4.2: Different case of feature observations.

features are at a long distance, as depicted in Figure 4.2(b). In such scenarios, the XYZ feature parametrization tends to be less effective due to the high uncertainty in-depth estimates for distant features and the elevated position uncertainty for features with low parallax. Conversely, neither the XYZ feature parametrization nor IDP provides adequate information when feature observations are close and aligned with the two cameras, as shown in Figure 4.2(c).

Various strategies have been employed to address the challenges posed by problematic features in the context of VI-SLAM, as discussed in [97]. RANdom SAmple Consensus (RANSAC)[98] is a common choice for feature selection and elimination of features with small parallax, as seen in many modern SLAM approaches [10, 12, 30]. However, RANSAC is essentially a randomized method lacking awareness of the frame's structure, including motion state and feature reliability [46]. As highlighted in [93], an approach proposed by [99] introduces a hybrid method. This selectively utilizes problematic features for rotation estimation, aiming to maintain consistency and enhance accuracy by combining them with reliable nearby features, which inherently have lower uncertainty. Another approach, proposed by [100], involves using inertial measurements to determine weights for each feature. This assigns lower weight to problematic features, mitigating their impact on the overall estimation.

However, the challenge persists in the detection and categorization of features into non-problematic and problematic categories, underscoring the complexity of this mechanism and its potential impact on the entire system [89].

To overcome this challenge in the proposed PVI-SLAM, the parallax parametrization method proposed by Zhao et al. [89] is integrated, which introduces the parallax angle into the state vector as:

$$\mathbf{f}_j = [\psi_j, \theta_j, \omega_j]^\top, \tag{4.3}$$

where ψ_j, θ_j, and ω_j are the azimuth, elevation angle, and parallax angle, respectively. These parametrization parameters are determined by way of selecting main and associate anchors. The main anchor corresponds to the pose at which the observation of $\mathbf{f}_j$ is initially recorded. Subsequently, the pose that observes the same feature for the second time becomes the associate anchor. When the feature is observed more than twice, the main and associate anchors can be substituted with either the maximum or parallax angle exceeding a predefined threshold.

The inclusion of parallax parameters has demonstrated superior accuracy, efficiency, and convergence properties compared to other BA parametrization methods [89].

4.2 IMU pre-integration

Different from the content covered in Chapter 3, this chapter centers on a 6-DoF IMU, which integrates measurements from a 3-axis gyroscope and a 3-axis accelerometer. These measurements result in noisy and biased data for the linear acceleration, represented as $\mathbf{a}_m$, and the angular velocity, denoted as $\boldsymbol{\omega}_m$, at time t in the inertial frame $\{I\}$, expressed as:

$$^{I_t}\mathbf{a}_m(t) = \mathbf{R}_{I_t}^{W}(t)^\top \left({}^{W}\mathbf{a}(t) - {}^{W}\mathbf{g}\right) + \mathbf{b}_a(t) + \boldsymbol{\eta}_a(t), \tag{4.4}$$

$$^{I_t}\boldsymbol{\omega}_m(t) = {}^{I_t}\boldsymbol{\omega}(t) + \mathbf{b}_\omega(t) + \boldsymbol{\eta}_\omega(t), \tag{4.5}$$

where $\mathbf{R}_{I_t}^{W} \in \mathrm{SO}(3)$ represents the IMU rotation matrix at time t. $\boldsymbol{\omega}$ is the true instantaneous angular velocity of the $\{I\}$ relative to global frame $\{W\}$, true linear acceleration, $\mathbf{a}$, and the gravity vector, $\mathbf{g}$, are specified in $\{W\}$. $\mathbf{b}_a$ and $\mathbf{b}_\omega$ refer to slowly varying sensor biases. The terms $\boldsymbol{\eta}_a$ and $\boldsymbol{\eta}_\omega$ represent zero-mean Gaussian noises associated with linear acceleration and angular velocity, respectively, with variances $\boldsymbol{\sigma}_a^2$ and $\boldsymbol{\sigma}_\omega^2$.

The kinematic model is expressed through the following equations:

$$\dot{\mathbf{R}}_{I_t}^{W}(t) = \mathbf{R}_{I_t}^{W}(t)^{I_t}\boldsymbol{\omega}(t)^{\wedge}, \tag{4.6}$$

$$^{W}\dot{\mathbf{v}}_{I_t}(t) = {}^{W}\mathbf{a}(t), \tag{4.7}$$

$$^{W}\dot{\mathbf{t}}_{I_t}(t) = {}^{W}\mathbf{v}_{I_t}(t), \tag{4.8}$$

where $\dot{}$ represents the differentiation operator with respect to time t. ${}^{W}\mathbf{v}_{I_t}$ and ${}^{W}\mathbf{t}_{I_t}$ are the position and velocity of the IMU at time t in the global frame W, respectively. The computation of the pose and velocity at time t_2 based on the known initial conditions at time t_1 is expressed as follows:

$$\mathbf{R}_{I_2}^{W} = \mathbf{R}_{I_1}^{W}\left(\prod_{t_1}^{t_2}\operatorname{Exp}\left({}^{I_t}\boldsymbol{\omega}(t)\right)^{dt}\right) \tag{4.9}$$

$$^{W}\mathbf{v}_{I_2} = {}^{W}\mathbf{v}_{I_1} + \int_{t_1}^{t_2}{}^{W}\mathbf{a}(t)dt \tag{4.10}$$

$$^{W}\mathbf{t}_{I_2} = {}^{W}\mathbf{t}_{I_1} + {}^{W}\mathbf{v}_{I_2}\Delta t + \int_{t_1}^{t_2}\int_{t_1}^{t}{}^{W}\mathbf{a}(s)dsdt \tag{4.11}$$

Using Equation (4.4) and Equation (4.5), the above equations can be expressed as a function of the IMU measurements $\mathbf{a}_m$ and $\boldsymbol{\omega}_m$:

$$\mathbf{R}_{I_2}^{W} = \mathbf{R}_{I_1}^{W}\left(\prod_{t_1}^{t_2}\operatorname{Exp}\left({}^{I_t}\boldsymbol{\omega}_m(t) - \mathbf{b}_{\omega}(t)\right)^{dt}\right) \tag{4.12}$$

$$^{W}\mathbf{v}_{I_2} = {}^{W}\mathbf{v}_{I_1} + \mathbf{g}\Delta(t) + \int_{t_1}^{t_2}\mathbf{R}_{I_t}^{W}(t)\left({}^{I_t}\boldsymbol{a}_m(t) - \mathbf{b}_{\omega}(t)\right)dt \tag{4.13}$$

$$\begin{aligned} {}^{W}\mathbf{t}_{I_2} &= {}^{W}\mathbf{t}_{I_1} + {}^{W}\mathbf{v}_{I_1}\Delta t + \frac{1}{2}{}^{W}\mathbf{g}\Delta t^2 \\ &+ \int_{t_1}^{t_2}\int_{t_1}^{t}\mathbf{R}_{I_s}^{W}(s)\left({}^{I_s}\boldsymbol{a}_m(s) - \mathbf{b}_{\omega}(s)\right)dsdt \end{aligned} \tag{4.14}$$

These equations, while suitable for factor graph optimization, possess the limitation of requiring recomputation whenever the linearization point at time t_i changes. To circumvent

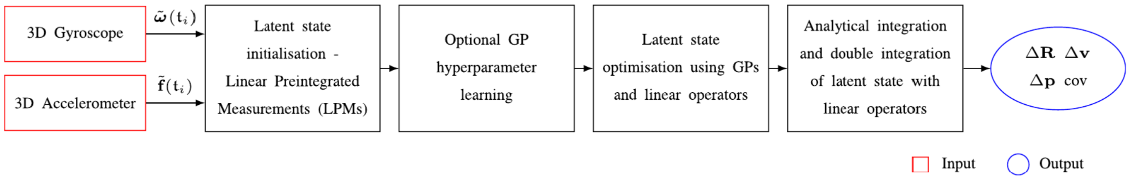


FIGURE 4.3: Overview of UGPM utilizing continuous pre-integration with GP [8].

this issue, the relative motion between t_1 and t_2 can be pre-integrated, offering independence from pose and velocity. This pre-integration is expressed as follows:

$$\Delta\mathbf{R}_{t_2}^{t_1} \doteq (\mathbf{R}_{I_1}^W)^\top \mathbf{R}_{I_2}^W = \prod_{t_1}^{t_2} \mathrm{Exp}\left({}^{I_t}\boldsymbol{\omega}_m(t) - \mathbf{b}_\omega(t)\right)^{dt} \tag{4.15}$$

$$\begin{aligned}\Delta\mathbf{v}_{t_2}^{t_1} &\doteq (\mathbf{R}_{I_1}^W)^\top \left({}^W\mathbf{v}_{I_2} - {}^W\mathbf{v}_{I_{t_1}} - {}^W\mathbf{g}\Delta t\right) \\ &= \int_{t_1}^{t_2} \mathbf{R}_{I_t}^{I_1}(t)\left({}^{I_t}\boldsymbol{a}_m(t) - \mathbf{b}_\omega(t)\right) dt\end{aligned} \tag{4.16}$$

$$\begin{aligned}\Delta\mathbf{t}_{t_2}^{t_1} &\doteq (\mathbf{R}_{I_1}^W)^\top \left({}^W\mathbf{t}_{I_2} - {}^W\mathbf{t}_{I_1} - {}^W\mathbf{v}_{I_1}\Delta t - \frac{1}{2}{}^W\mathbf{g}\Delta t^2\right) \\ &= \int_{t_1}^{t_2}\int_{t_1}^{t} \mathbf{R}_{I_s}^{I_1}(s)\left({}^{I_t}\boldsymbol{a}_m(s) - \mathbf{b}_\omega(s)\right) dsdt\end{aligned} \tag{4.17}$$

Unlike $\Delta\mathbf{R}_{t_2}^{t_1}$, neither $\Delta\mathbf{v}_{t_2}^{t_1}$ nor $\Delta\mathbf{t}_{t_2}^{t_1}$ represent the true physical change in velocity and position. Instead, they are defined to ensure the right-hand side of equations remains independent of the state at time t_i and gravitational effects.

In the proposed work, PVI-SLAM, GP is employed for continuous pre-integration, a technique introduced by Le Gentil et al. [8], as illustrated in the overview presented in Figure 4.3. This approach differs from PM, as in [25] and [1], where Equation (4.15) – Equation (4.17) were numerically integrated using the rectangle rule with discrete IMU measurements. Conventional numerical integration treats acceleration and angular velocity between two consecutive IMU timestamps as constant, potentially impacting the accuracy of the system. The use of GP for continuous pre-integration provides a more refined and continuous model of the IMU measurements, allowing for improved accuracy in the integration process.

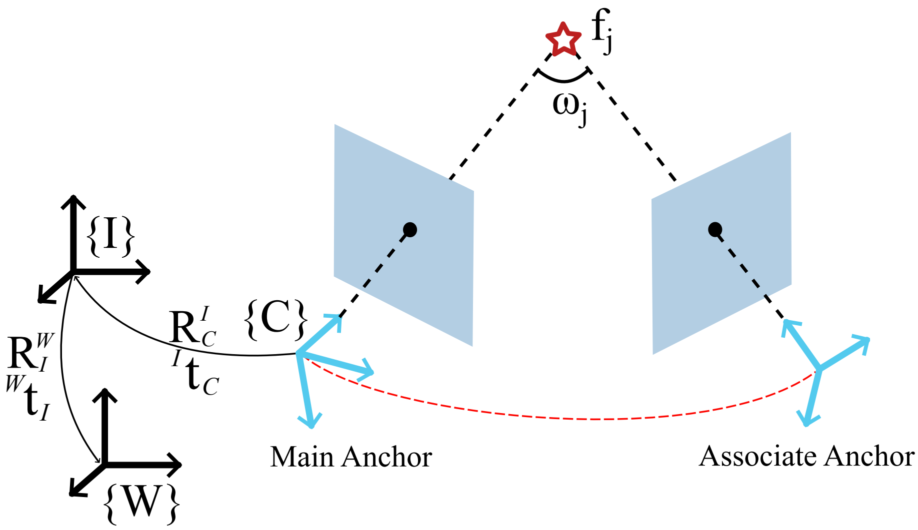


FIGURE 4.4: The illustration of PVI-SLAM system.

4.3 Parallax Visual-Inertial SLAM

In this chapter, the primary objective within the VI-SLAM framework is to simultaneously track the state of the system and map landmarks. These systems are equipped with an IMU and a monocular camera. **To achieve this goal, a PVI-SLAM [101] system is proposed, making use of both PBA and UGPM**.

4.3.1 Problem Statement

The state of PVI-SLAM can be represented as:

$$\mathcal{X} = \{\mathcal{P}_{I_1}, \cdots, \mathcal{P}_{I_N}, \mathbf{f}_1, \cdots, \mathbf{f}_M\}, \tag{4.18}$$

where $\mathbf{f}_j$ represents the j^{th} feature position in PBA parametrization as in Equation (4.3) and the IMU state, $\mathcal{P}_I$, at time i can be written as:

$$\mathcal{P}_{I_i} = \left\{\mathbf{R}^W_{I_i}, {}^W\mathbf{t}_{I_i}, {}^W\mathbf{v}_{I_i}, \mathbf{b}_{\omega_i}, \mathbf{b}_{a_i}\right\}. \tag{4.19}$$

Here, $\mathbf{R}^W_{I_i} \in \mathrm{SO}(3)$ is the rotation matrix of IMU at time i, $\{I_i\}$, in the global frame, $\{W\}$. ${}^W\mathbf{v}_{I_i} \in \mathbb{R}^3$ and ${}^W\mathbf{t}_{I_i} \in \mathbb{R}^3$ are the velocity and position of the IMU in $\{W\}$ at time i. $\mathbf{b}_{a_i}$ and $\mathbf{b}_{\omega_i}$ are slowly varying sensor biases from the IMU's accelerometer and gyroscope, treated as constant between two state timestamps. Camera pose can be obtained using the known extrinsic matrix, $\left(\mathbf{R}^I_C, {}^I\mathbf{t}_C\right)$, as shown in Equation (4.20). A detailed illustration of the reference frames is presented in Figure 4.4.

$$\mathbf{R}_C^W = \mathbf{R}_I^W \mathbf{R}_C^I, \quad {}^W\mathbf{t}_C = {}^W\mathbf{t}_I + \mathbf{R}_I^W {}^I\mathbf{t}_C. \tag{4.20}$$

The estimation of these states employs MLE as Eqaution (2.19). The objective function $J(\mathcal{X})$ integrates information from various sensor measurements relevant to state estimation. In PVI-SLAM, the objective function of the optimization problem tightly couples the measurements from the IMU and the monocular camera, allowing joint estimation of all states [86] as mentioned in Equation (2.20). The formulation of this objective function is as follows:

$$J(\mathcal{X}) := \underbrace{\sum_{i=1}^{N} \sum_{j \in \mathcal{J}(i)} \mathbf{e}_r^{i,j\top} \mathbf{W}_r^i \mathbf{e}_r^{i,j}}_{\text{visual}} + \underbrace{\sum_{i=1}^{N-1} \mathbf{e}_s^{i\top} \mathbf{W}_s^i \mathbf{e}_s^i}_{\text{inertial}}, \tag{4.21}$$

where i and j identify the IMU frame and feature index. $\mathcal{J}(i)$ includes all visible features in IMU frame at time i. $\mathbf{e}_r^{i,j}$ is the reprojection error, $\mathbf{e}_s^i$ is the inertial error, and $\mathbf{W}_s^i$ represents the inverse covariance of the IMU residual at time i. As the uncertainty of the image coordinates for all features is assumed to be independent and identical, the weight matrix $\mathbf{W}_r^i$ is considered as an identity matrix.

4.3.2 Parallax-Based Reprojection Error

The reprojection error, $\mathbf{e}_r^{i,j}$, is computed as the disparity between the observed value, $\mathbf{u}_j^i$, and the estimated value, $\hat{\mathbf{u}}_j^i$:

$$\mathbf{e}_r^{i,j} = \mathbf{u}_j^i - \hat{\mathbf{u}}_j^i \in \mathbb{R}^2. \tag{4.22}$$

As discussed in Section 4.1, the computation of the reprojection error depends on the selection of anchors. Here, the position of main anchor (m) in the camera frame is denoted as ${}^W\mathbf{t}_{C_m}$, the position of associate anchor (a) in the camera frame is referred to as ${}^W\mathbf{t}_{C_a}$, and all other camera positions are defined as ${}^W\mathbf{t}_{C_i}$. Then, the projection model based on parallax angle parametrization can be presented as:

$$\mathbf{u}_j^i = \begin{bmatrix} u_j^i \\ v_j^i \end{bmatrix} = \pi(\mathbf{K}\ (\mathbf{R}_{C_i}^W)^\top\ \mathbf{x}_j^i), \tag{4.23}$$

where the function $\pi(\cdot)$ is defined as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \pi\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} x/z \\ y/z \end{bmatrix}. \tag{4.24}$$

Here, $\mathbf{K}$ is the camera intrinsic matrix, $\mathbf{R}_{C_i}^{W}$ is rotation matrix of camera pose i, $\mathbf{u}_j^i$ is the reprojected image point from feature $\mathbf{f}_j$ to image i, and:

$$\mathbf{x}_j^i = \begin{cases} \mathbf{x}_j^m, & \text{if } i = m, \text{ else} \\ \sin\left(\omega_j + \varphi_j\right) \left\| {}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m} \right\| \mathbf{x}_j^m - \sin\omega_j \left({}^W\mathbf{t}_{C_i} - {}^W\mathbf{t}_{C_m}\right). \end{cases} \tag{4.25}$$

$\mathbf{x}_j^m$ is the unit vector from ${}^W\mathbf{t}_{C_m}$ to $\mathbf{f}_j$:

$$\mathbf{x}_j^m = \begin{bmatrix} \sin\psi_j \cos\theta_j \\ \sin\theta_j \\ \cos\psi_j \cos\theta_j \end{bmatrix}, \tag{4.26}$$

and φ_j is the angle between the vector $\left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)$ and vector $\mathbf{x}_j^m$:

$$\varphi_j = \arccos\left(\mathbf{x}_j^m \cdot \frac{{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}}{\left\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right\|}\right). \tag{4.27}$$

The values of ψ_j and θ_j can be computed using the following equations:

$$\psi_j = \operatorname{atan2}\left(x_j^m, z_j^m\right), \tag{4.28}$$

$$\theta_j = \operatorname{atan2}\left(y_j^m, \sqrt{\left(x_j^m\right)^2 + \left(z_j^m\right)^2}\right), \tag{4.29}$$

where $\mathbf{x}_j^m = \begin{bmatrix} x_j^m & y_j^m & zj^m \end{bmatrix}^\top$. Additionally, ω_j can be determined using the equation:

$$\omega_j = \arccos\left(\frac{\hat{\mathbf{x}}_j^m \cdot \hat{\mathbf{x}}_j^a}{\left\|\hat{\mathbf{x}}_j^m\right\| \left\|\hat{\mathbf{x}}_j^a\right\|}\right). \tag{4.30}$$

4.3.3 Observation Ray Based Error Function

The reprojection error, as mentioned in Equation (4.22), plays a crucial role in VI-SLAM by quantifying the disparity between observed image points and the corresponding projections of 3D points in the image space. The reprojection error is highly sensitive to pixel-level noise in the images, which highlights the need for careful consideration during the optimization process.

Moreover, challenges may arise during the initialization phase, especially when dealing with features positioned behind the camera. This scenario has the potential to impede the convergence of the optimization process to a valid solution. The reprojection error is known to be influenced by the initial guess, emphasizing the significance of robust initialization strategies in mitigating such challenges.

The observation ray objective function introduces geometric constraints based on the directions of observation rays. This proves advantageous in handling features and their corresponding 3D positions, explicitly considering occluded or unseen points. In situations where features are intermittently located behind the camera, this method becomes instrumental in preventing the optimization process from erroneously projecting them onto the image plane. The utilization of observation rays gains particular relevance in resolving depth ambiguities associated with points situated behind the camera, depending on the specific geometry and camera setup [97].

By incorporating these geometric constraints, the observation ray objective function significantly enhances the overall robustness and accuracy of the system. This contribution is particularly valuable in addressing challenges related to feature initialization and mitigating potential depth ambiguities, ultimately fortifying the system's performance in various scenarios [97].

In PBA framework, the observation ray error function, integral to the work in this chapter, is computed using the ray derived in Equation (4.25):

$$\mathbf{e}_r^{i,j} = \mathbf{v}_j^i - \hat{\mathbf{v}}_j^i \in \mathbb{R}^3, \tag{4.31}$$

where

$$\mathbf{v}_j^i = \xi \left(\mathbf{K}^{-1} \begin{bmatrix} u_j^i \\ v_j^i \\ 1 \end{bmatrix} \right), \qquad \hat{\mathbf{v}}_j^i = \xi \left((\mathbf{R}_{C_i}^W)^\top \, \mathbf{x}_j^i \right). \tag{4.32}$$

Here, $\xi(\cdot) = \frac{\cdot}{|\cdot|}$ represents the normalization operation. This error function plays a crucial role in the system and helps enhance its accuracy by considering the direction of the observation ray.

4.3.4 Inertial and Bias Error

To perform GP as in [8], the modelling process commences by defining the following equations and representing ${}^{I_1}\dot{\mathbf{r}}_{I_t}(t)$ and ${}^{I_t}\mathbf{a}_m(t)$ as six independent GP:

$$\mathbf{J}_r\left({}^{I_1}\mathbf{r}_{I_t}(t)\right){}^{I_1}\dot{\mathbf{r}}_{I_t}(t) = {}^{I_t}\boldsymbol{\omega}(t) \tag{4.33}$$

$${}^{I_1}\mathbf{a}_m(t) = \Delta\mathbf{R}_t^{t_1}(t){}^{I_t}\mathbf{a}_m(t). \tag{4.34}$$

Here, the rotation vector in $\{I\}$ at time t_1 is obtained using the logarithm map, ${}^{I_1}\mathbf{r}_{I_t}(t) = \mathrm{Log}\left(\mathbf{R}_{I_t}^{I_1}(t)\right)$, and ${}^{I_1}\mathbf{a}_m(t)$ represents the accelerometer measurements reprojected to $\{I\}$ at time t_1. The inducing values of these GP are learned by formulating a nonlinear optimization problem based on the actual IMU measurements, $\boldsymbol{a}_m(t)$ and $\boldsymbol{\omega}_m(t)$. After learning, it is possible to infer the pre-integrated measurements, $\Delta\mathbf{R}_{t_2}^{t_1}$, $\Delta\mathbf{v}_{t_2}^{t_1}$, and $\Delta\mathbf{t}_{t_2}^{t_1}$ at any timestamp by analytically integrating and double integrating the continuous signals ${}^{I_1}\dot{\mathbf{r}}_{I_t}$ and ${}^{I_1}\mathbf{a}_m(t)$ [8].

In [8], the update of biases is incorporated using the first-order expansion, following a similar approach to [1, 25]:

$$\begin{aligned}
\Delta\mathbf{R}_{t_2}^{t_1}(\mathbf{b}_\omega) &\approx \Delta\mathbf{R}_{t_2}^{t_1}(\overline{\mathbf{b}}_\omega)\,\mathrm{Exp}\left(\frac{\partial\Delta\mathbf{R}_{t_2}^{t_1}}{\partial\mathbf{b}_\omega}\delta\mathbf{b}_\omega\right) \\
\Delta\mathbf{v}_{t_2}^{t_1}(\mathbf{b}_a,\mathbf{b}_\omega) &\approx \Delta\mathbf{v}_{t_2}^{t_1}(\overline{\mathbf{b}}_a,\overline{\mathbf{b}}_\omega) + \frac{\partial\Delta\mathbf{v}_{t_2}^{t_1}}{\partial\mathbf{b}_a}\delta\mathbf{b}_a + \frac{\partial\Delta\mathbf{v}_{t_2}^{t_1}}{\partial\mathbf{b}_\omega}\delta\mathbf{b}_\omega, \\
\Delta\mathbf{p}_{t_2}^{t_1}(\mathbf{b}_a,\mathbf{b}_\omega) &\approx \Delta\mathbf{p}_{t_2}^{t_1}(\overline{\mathbf{b}}_a,\overline{\mathbf{b}}_\omega) + \frac{\partial\Delta\mathbf{p}_{t_2}^{t_1}}{\partial\mathbf{b}_a}\delta\mathbf{b}_a + \frac{\partial\Delta\mathbf{p}_{t_2}^{t_1}}{\partial\mathbf{b}_\omega}\delta\mathbf{b}_\omega.
\end{aligned} \tag{4.35}$$

The corrected bias vector is denoted as $\mathbf{b} = \overline{\mathbf{b}} + \delta\mathbf{b}$, where $\bar{\cdot}$ represents the prior knowledge at the time of pre-integration.

Utilizing Equation (4.35), the IMU residual, $\mathbf{e}_s^i$, between the two consecutive frames at time i and $i+1$, can be written as follows:

$$\mathbf{e}_s^i = \begin{bmatrix} \mathrm{Log}\left(\left(\Delta\mathbf{R}_{i+1}^i\right)^\top \Delta\hat{\mathbf{R}}_{i+1}^i\right) \\ \Delta\hat{\mathbf{v}}_{i+1}^i - \Delta\mathbf{v}_{i+1}^i \\ \Delta\hat{\mathbf{t}}_{i+1}^i - \Delta\mathbf{t}_{i+1}^i \\ \mathbf{b}_{\omega_{(i+1)}} - \mathbf{b}_{\omega_i} \\ \mathbf{b}_{a_{(i+1)}} - \mathbf{b}_{a_i}, \end{bmatrix} \in \mathbb{R}^{15}, \tag{4.36}$$

where $\Delta\mathbf{R}^i_{i+1}$, $\Delta\mathbf{v}^i_{i+1}$, and $\Delta\mathbf{t}^i_{i+1}$ stand for the pre-integrated values of rotation, velocity, and position as from Equation (4.35). These contrast with $\Delta\hat{\mathbf{R}}^i_{i+1}$, $\Delta\hat{\mathbf{v}}^i_{i+1}$, and $\Delta\hat{\mathbf{t}}^i_{i+1}$, which represent estimated relative motion changes and are independent of the pose and velocity at time i.

4.3.5 Nonlinear Least Squares Optimization

The optimization process on the manifold follows the "lift-solve-retract" scheme, as detailed in Section 2.2.3. Initially, it involves lifting the cost function (Equation (4.21)) to the Euclidean space, followed by the application of retraction to PVI-SLAM:

$$\begin{aligned}
\mathbf{R}^W_{I_i} &\leftarrow \mathbf{R}^W_{I_i}\ \mathrm{Exp}\left(\delta\boldsymbol{\phi}_i\right), & \delta\boldsymbol{\phi}_i &\in \mathbb{R}^3 \\
{}^W\mathbf{t}_{I_i} &\leftarrow {}^W\mathbf{t}_{I_i} + \mathbf{R}^W_{I_i}\ \delta\mathbf{t}_i, & \delta\mathbf{t}_i &\in \mathbb{R}^3 \\
{}^W\mathbf{v}_{I_i} &\leftarrow {}^W\mathbf{v}_{I_i} + \delta\mathbf{v}_i, & \delta\mathbf{v}_i &\in \mathbb{R}^3 \\
\delta\mathbf{b}_{\omega_i} &\leftarrow \delta\mathbf{b}_{\omega_i} + \tilde{\delta}\mathbf{b}_{\omega_i}, & \delta\mathbf{b}_{\omega_i} &\in \mathbb{R}^3 \\
\delta\mathbf{b}_{a_i} &\leftarrow \delta\mathbf{b}_{a_i} + \tilde{\delta}\mathbf{b}_{a_i}, & \delta\mathbf{b}_{a_i} &\in \mathbb{R}^3
\end{aligned} \tag{4.37}$$

In the solving step, GN and LM algorithms (Section 2.2.2) are leveraged for the lifted cost function to refine the $\delta\boldsymbol{\phi}_i$, $\delta\mathbf{t}_i$, $\delta\mathbf{v}_i$, $\tilde{\delta}\mathbf{b}_{\omega_i}$, and $\tilde{\delta}\mathbf{b}_{a_i}$ in all the timestamps. In the retracting step, the refined solution is lifted back to the manifold, as shown in (4.37). With the updated estimate, the optimization process can repeat the subsequent steps. Detailed information on the Jacobian calculation of the residual in Equation (4.22) and Equation (4.36) can be found in Appendix A and Appendix B, respectively.

4.4 Performance Evaluations of PVI-SLAM

This section undertakes a comprehensive quantitative evaluation of the proposed methodology, PVI-SLAM. In Section 4.4.1, the evaluation begins with a comparative analysis of the pure-vision performance between PBA and SBA, employing the parallax and XYZ parametrization methods, respectively. Two distinct datasets, namely the "MALAGA PARKING-6L" dataset [9] and the "Starry Night" Dataset [5], are utilized for the assessment, incorporating a variety of initialization strategies.

To explore the impact of IMU integration on PBA, the IMU measurements from Chapter 3 (Equation (3.18)) are incorporated. This analysis provides valuable insights into the simplicity and effectiveness of PBA when combined with IMU data.

FIGURE 4.5: (Left) Trajectory (Right) Sample frame from “MALAGA” Dataset [9]

Subsequently, the performance of PVI-SLAM is evaluated using the “EuRoC” dataset in Section 4.4.2. The system incorporates the suitable objective function, as elucidated in Section 4.3, and operates within a manifold framework, thereby serving as a benchmark against state-of-the-art methodologies.

The investigation then shifts its focus to the “KITTI” dataset, highlighting the advantages of employing the parallax parametrization method in Section 4.4.3. In Section 4.4.3.2, Robustness assessments are carried out by integrating UGPM and comparing its performance with the PM proposed in [1].

Initialization relies on feature observations and poses extracted through ORB-SLAM3 [12]. In Section 4.4.3.3, the system’s robustness is further examined by deploying Visual Odometry (VO) without closing the loop, and a comparative analysis is carried out using both the observation ray objective function and the reprojection error. This comprehensive evaluation aims to provide a nuanced understanding of the proposed methodology across various datasets and scenarios, ensuring a robust assessment of its performance.

4.4.1 “MALAGA” and “Starry Night” Dataset

The publicly accessible “MALAGA PARKING-6L” dataset was collected by an electric vehicle equipped with a camera providing a reliable ground-truth with estimated uncertainty bounds [9]. Images collected during the 250m close-loop trajectory, called the PARKING-6L dataset, are chosen for evaluation as in Figure 4.5. To make the dataset to be suitable for monocular BA, only images from the right-side camera are used. The information of features from those images has been extracted using SIFT [20], RANSAC [98], and the eight-point algorithm [102] as described in [103]. The number of images has been reduced from 508 to 170 as the key-frame for this loop, now containing 170 poses, $58,404$ features, and $167,285$ projections [89]. The "Starry Night" dataset aligns with the data previously utilized in Chapter 3.

4.4.1.1 Comparison Criteria

For an accurate comparison between the two implementations of BA, identical initial input of camera poses and observations are required. Due to the different methods of feature parameterization, the input parameters for features need to be converted to suit the respective BAs. The output of the comparison includes assessments of the initial and final reprojection errors, along with the number of iterations required. Calculation of reprojection error can be done by averaging the squared reprojection error (Equation (4.22)), which is stacked with all the related features from all the camera poses. The RMSE values for poses and features are also compared. In the case of PVI-SLAM, the RMSE of camera pose and feature position has been compared with the results of PBA to assess the improvement achieved through the utilization of IMU.

4.4.1.2 Comparison Result

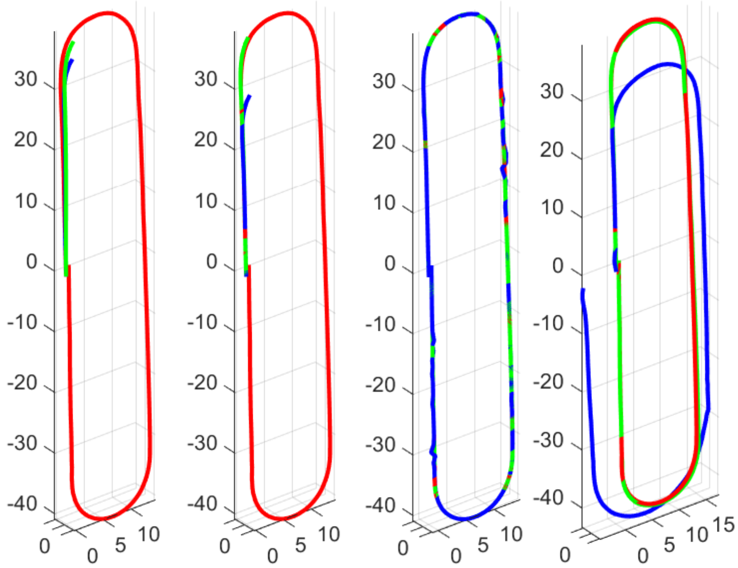
Since the "MALAGA" dataset does not provide the ground-truth feature position, two different initial inputs for both BAs are used, which are *Initialization 1* and *Initialization 2*.

- *Initialization 1*: Ground-truth poses, and observations in (u, v) value are used to compute the initial feature values for both BAs. For the PBA, the estimated parallax parameter for features can be computed with the given poses and the observation as mentioned in Section 4.1. The same initial value needs to be used for both BAs to allow a fair comparison. The estimated feature position in the XYZ parameter for SBA can be calculated from the PBA parameters using the ground-truth anchor poses.

- *Initialization 2*: Estimated poses obtained from VO and observations are used to compute the initial values of poses. In addition, estimated feature positions in parallax and XYZ parameters are used as initial values of features, which were computed in the same way as *Initialization 1*.

TABLE 4.1: Comparison result of "MALAGA" from 30 and 170 images with two different initialization methods.

		Init 1		**Init 2**	
		30	**170**	**30**	**170**
PBA	Trans.RMSE (m)	0.2940	0.000003	0.0286	0.0718
	Rot.RMSE (deg)	0.0087	0.00029	0.0293	0.0728
	Iteration	9	104	8	51
	Initial Cost	580.2133	498896	8.2879	462.998
	Final Cost	**0.1415**	**212.6789**	**0.1415**	**0.1092**
SBA	Trans.RMSE (m)	0.6472	0.0075	1.9635	1.1164
	Rot.RMSE (deg)	0.0086	0.00081	0.0305	0.1291
	Iteration	30	79	13	10
	Initial Cost	580.2133	498896	8.2879	462.998
	Final Cost	0.9938	7897.9	0.9925	282.764



(a) 30 images: Initialization 1 (b) 30 images: Initialization 2 (c) 170 images: Initialization 1 (d) 170 images: Initialization 2

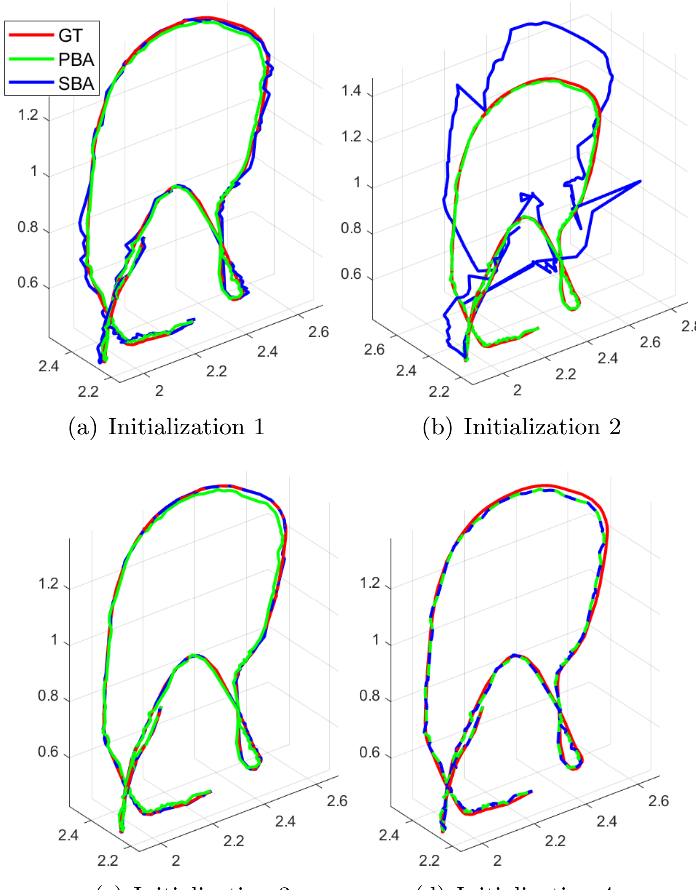
FIGURE 4.6: The result of PBA and SBA from 30 and 170 images in "MALAGA" dataset with two different initialization methods (Red Trajectory: Ground-Truth, Green Trajectory: PBA, Blue Trajectory: SBA).

Whereas the "Starry Night" dataset provides ground-truth for both poses and feature positions, it allows for testing with more variety of initial inputs, including *Initialization 1* and *Initialization 2*, shown as follows:

- *Initialization 3*: Ground-truth of poses is used as the initial value. Ground-truth feature positions in the parallax parameter can be computed using the ground-truth

Table 4.2: Comparison result of "Starry Night" (500 features) from 200 images with four different initialization methods.

		Init 1	**Init 2**	**Init 3**	**Init 4**
		200	**200**	**200**	**200**
PBA	Trans.RMSE (m)	0.0047	0.0047	0.0047	0.0047
	Rot.RMSE (deg)	0.0004	0.2936	0.0004	0.2936
	Feature.RMSE (m)	0.2405	0.1997	0.1997	0.1997
	Iteration	101	22	11	22
	Initial Cost	4.5201	64152	1.9843	4.8201
	Final Cost	2.1122	1.8374	1.8374	1.8374
SBA	Trans.RMSE (m)	0.0020	0.1070	0.00004	0.0047
	Rot.RMSE (deg)	0.0006	0.3126	0.00001	0.00041
	Feature.RMSE (m)	1.7506	4.5561	0.00016	2.0701
	Iteration	27	90	28	1
	Initial Cost	4.8201	64152.1	1.9843	1.8374
	Final Cost	2.8219	2895.4	1.9539	1.8374



(a) Initialization 1 (b) Initialization 2

(c) Initialization 3 (d) Initialization 4

Figure 4.7: The result of PBA and SBA from 200 images (500 features) in "Starry Night" dataset with four different initialization methods.

poses and ground-truth feature positions instead of using the observation data to calculate the parallax parameter. Ground-truth feature positions in the XYZ parameter provided from the data are directly used as the initial value of SBA.

- *Initialization 4*: For PBA, estimated poses and observations are used to compute the initial feature values. Estimated poses are achieved with the IMU measurements and extrinsic matrix from the "Starry Night" dataset. Estimated feature positions in the parallax parameter can be computed using the estimated poses and observations. The output poses and feature positions from PBA are used as the initial value of SBA. This is to check whether the result obtained from PBA is a minimum for SBA or not.

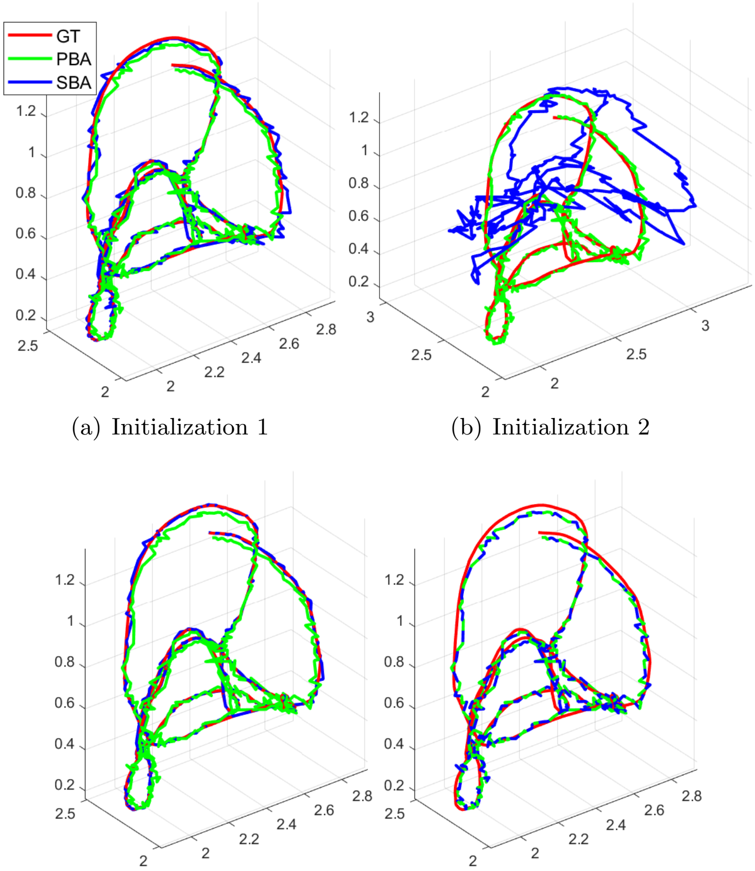
"MALAGA" dataset with 30 images. The results of PBA and SBA on the "MALAGA" dataset with 30 images are presented in Table 4.1 and Figure 4.6. For the *Initialization 1* and *Initialization 2*, PBA converges to a lower final reprojection error in fewer iterations. Also, in the case of PBA, the RMSE of translation and rotation are smaller than SBA in both cases. The larger reprojection error, in both initial and final, can be seen in *Initialization 1* compared to *Initialization 2*, which is due to the absence of ground-truth feature parameters for initial value in both BAs.

"MALAGA" dataset with 170 images. As the loop is closed, PBA stably converged close to the ground-truth with a final reprojection error of 0.1092 using *Initialization 2*, as indicated in Table 4.1, and Figure 4.6(d). In contrast, loop-closure did not perform well with SBA, resulting in a final reprojection error of 282.764 with *Initialization 2*. In the case of *Initialization 1*, where the ground-truth feature is not provided, both methods exhibit significant initial and final costs. Despite PBA converging to a lower final cost of 212.679 compared to SBA's convergence to 7897.9, both methods seem to be trapped in a local minimum.

"Starry Night" dataset (500 Features) with 200 images. In *Initialization 4*, when refined poses and feature positions from the PBA are used as an initial value to SBA, the same final reprojection error is obtained as shown in Table 4.2. The SBA result presented in *Initialization 2* did not converge close enough to ground-truth poses compared to PBA, as can be easily seen in Figure 4.7(b). In all the cases, PBA converged to a lower final reprojection error and yielded better-refined poses and feature positions (Figure 4.7).

TABLE 4.3: Comparison result of "Starry Night" (80 features) from 500 images with four different initialization methods.

		Init 1	Init 2	Init 3	Init 4
		500	**500**	**500**	**500**
PBA	Trans.RMSE (m)	0.0088	0.0088	0.0088	0.0088
	Rot.RMSE (deg)	0.0017	0.1943	0.0017	0.1943
	Feature.RMSE (m)	0.8883	0.0833	0.0883	0.0883
	Iteration	41	44	41	44
	Initial Cost	4.1902	5642	2.0208	5642
	Final Cost	1.3778	1.3778	1.3778	1.3778
SBA	Trans.RMSE (m)	0.0013	0.1729	0.00008	0.0088
	Rot.RMSE (deg)	0.00065	0.2357	0.00003	0.0039
	Feature.RMSE (m)	0.8883	2.3860	0.00067	0.3952
	Iteration	74	297	66	1
	Initial Cost	4.1902	5642	2.0208	1.3778
	Final Cost	2.6636	77.6693	1.8102	1.3778



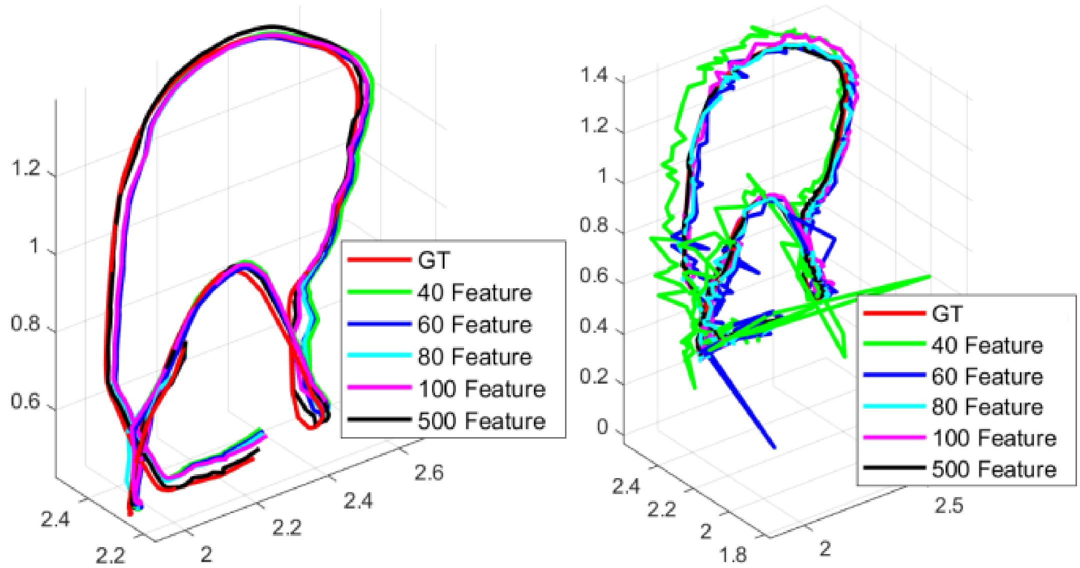
(a) Initialization 1 (b) Initialization 2

(c) Initialization 3 (d) Initialization 4

FIGURE 4.8: The result of PBA and SBA from 500 images (80 features) in "Starry Night" dataset with four different initialization methods.

TABLE 4.4: Comparison result of PBA and PVI-SLAM with 200 images and IMU measurements from "Starry Night" (40, 60, 80, 100, and 500 features).

		Init 2 - 200 Images				
		40	**60**	**80**	**100**	**500**
PBA	Trans.RMSE (m)	0.0148	0.0127	0.0065	0.0069	0.0047
	Rot.RMSE (deg)	0.2944	0.2960	0.2946	0.2957	0.2936
	Feature.RMSE (m)	0.1344	0.8764	0.9576	0.1461	0.1997
PVI-SLAM	Trans.RMSE (m)	0.0150	0.0114	0.0102	0.0142	0.0074
	Rot.RMSE (deg)	0.2964	0.2956	0.2959	0.2955	0.2941
	Feature.RMSE (m)	0.5253	0.2912	0.4471	0.2496	0.3257

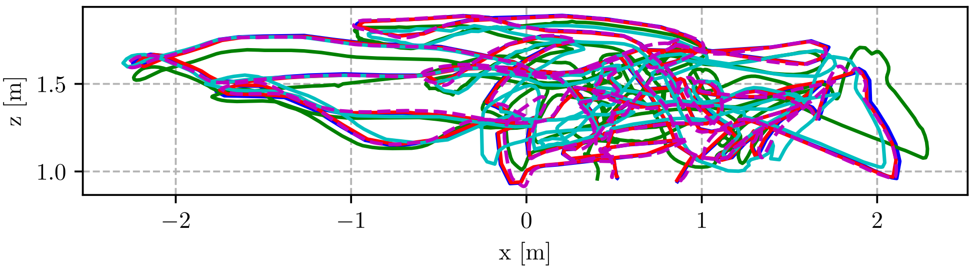


(a) PVI-SLAM (b) Parallax BA

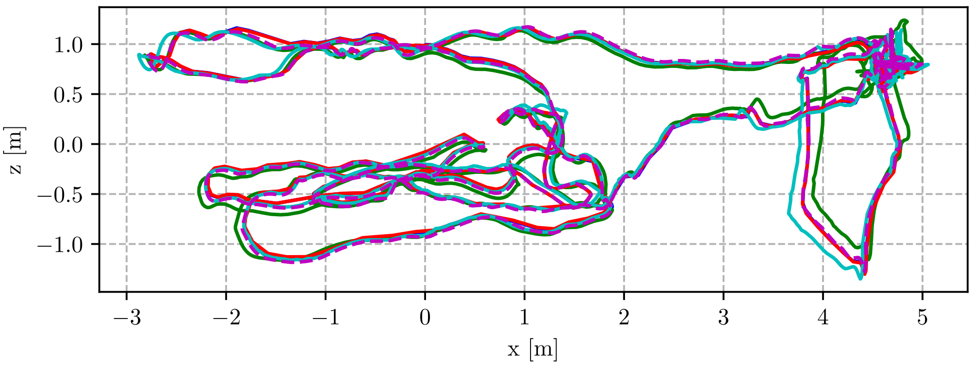
FIGURE 4.9: Comparison between PVI-SLAM and PBA with 200 images (40, 60, 80, 100, 500 features) from "Starry Night" dataset.

"Starry Night" dataset (80 Features) with 500 images. The result of PBA (Table 4.3) shows that the final reprojection error converges to a smaller value, 1.3778, than SBA in *Initialization 1* to *3*. Results using *Initialization 4* are the same, meaning the result of PBA is a minimum of SBA. The results of both BAs are close enough to the ground-truth in all initialization methods except the result of SBA with *initialization 2*, as seen in Figure 4.8(b).

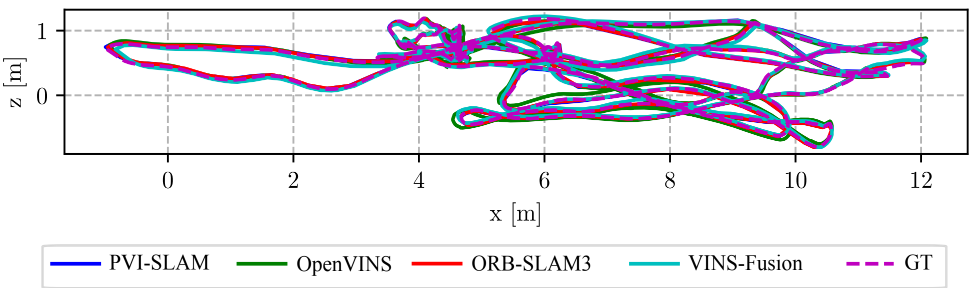
VI-SLAM: "Starry Night" dataset (40, 60, 80, 100, 500 Features) with 200 images. PVI-SLAM and PBA have been compared with different numbers of feature observations during the whole trajectory. As can be seen in Table 4.4, the performance of the pure vision system, PBA, seems to be comparable to PVI-SLAM. However, it cannot



(a) EuRoC-V101



(b) EuRoC-MH01



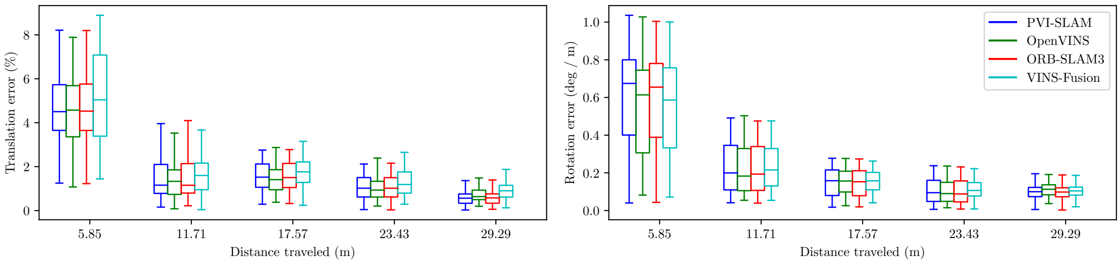
(c) EuRoC-MH03

FIGURE 4.10: The comparison of estimated trajectories between PVI-SLAM, VINS-Fusion [10], OpenVINS [11], and ORB-SLAM3 [12] using the "EuRoC" datasets.

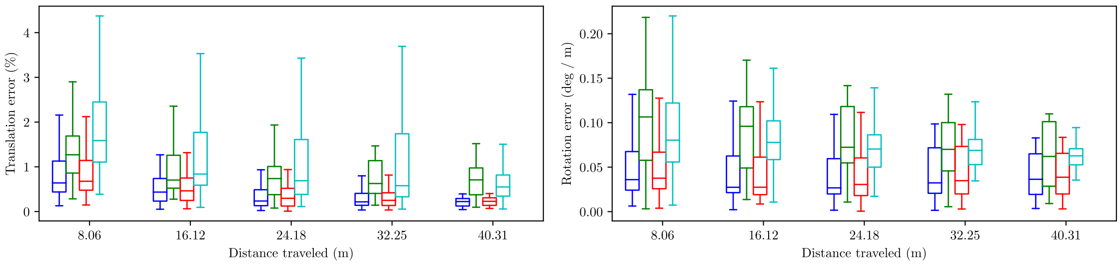
recover the right metric scale without ground-truth while IMU naturally helps to recover the metric scale. Moreover, PVI-SLAM shows more consistence and reliable performance than PBA, even with fewer feature observations.

4.4.2 "EuRoC" dataset

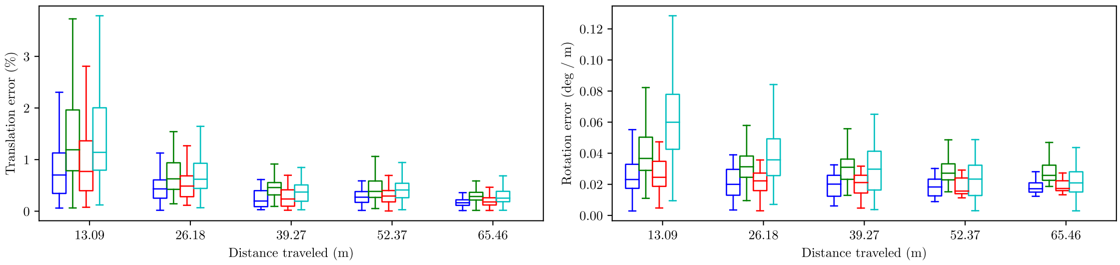
The "EuRoC" Micro Aerial Vehicle (MAV) dataset [104] is collected from two different environments. One setting is a machine hall, providing a challenging industrial environment with diverse conditions. The other environment is a Vicon room designed to evaluate



(a) EuRoC-V101



(b) EuRoC-MH01



(c) EuRoC-MH03

FIGURE 4.11: Comparison of translation error and rotation error between PVI-SLAM, VINS-Fusion [10], OpenVINS [11], and ORB-SLAM3 [12] for each "EuRoC" dataset.

the performance of multi-view reconstruction. The dataset comprises high-frequency IMU measurements, capturing rapid changes in acceleration and angular velocity with precision at rates of 200Hz. Simultaneously, front-down-looking stereo camera images are captured at a lower frequency, typically around 20Hz, facilitating visual feature tracking and mapping. The synchronization of IMU and camera data through precise timestamps ensures accurate temporal alignment, a critical factor for the successful fusion of visual and inertial information. Additionally, ground-truth odometry information obtained from laser tracking systems and Vicon is provided at the same high frequency as the IMU data. This provision serves as a reliable reference for evaluating the performance of VI-SLAM algorithms.

For the evaluation, sequences MH01, MH03 and V101 from the machine hall and the Vicon room, respectively, are utilized. Comparative analyses involve VINS-Fusion [10], OpenVINS [11], and ORB-SLAM3 [12]. In the case of the proposed PVI-SLAM method, estimated poses and image coordinates of feature observations are initialized using ORB-SLAM3. The impact of this initialization on PVI-SLAM is discussed in Section 4.4.3.3.

The trajectories estimated by the proposed method, PVI-SLAM and other state-of-the-art techniques are visually compared in Figure 4.10, while the average errors relative to the distance and angle travelled are depicted in Figure 4.11. The visualizations clearly indicate that PVI-SLAM and ORB-SLAM3 outperform VINS-Fusion and OpenVINS. Particularly in the more challenging MH01 and MH03 dataset, both PVI-SLAM and ORB-SLAM3 stand out prominently compared to other methods.

In the assessment of translation accuracy measured by RMSE, PVI-SLAM consistently outperforms other state-of-the-art methods across different sequences. For MH01, MH03, and V101 sequences, PVI-SLAM achieves translation RMSE values of **0.033m**, **0.030m**, and **0.035m**, respectively. In comparison, ORB-SLAM3 reports translation errors with values of 0.036m, 0.034m, and 0.038m for the corresponding sequences. OpenVINS records the highest translation errors with values of 0.142m, 0.108m, and 0.103m, while VINS-Fusion falls in between with values of 0.077m, 0.078m, and 0.110m.

Moving to the evaluation of rotational RMSE, PVI-SLAM maintains commendable performance across MH01, MH03, and V101 sequences, recording rotational RMSE values of 1.097°, 1.186°, and 5.513°, respectively. Though slightly higher, these values remain comparable to those of other state-of-the-art methods. OpenVINS, ORB-SLAM3, and VINS-Fusion exhibit rotational RMSE values of 1.606°, 1.106°, and 2.501° for MH01; 1.417°, 1.338°, and 1.640° for MH03; and 5.377°, 5.504°, and 6.281° for V101, respectively. The consistent performance of PVI-SLAM underscores its effectiveness in achieving accurate and competitive results in both translation and rotation, establishing it as a robust method in comparison to other leading techniques.

4.4.3 "KITTI" dataset

In contrast to the "EuRoC" dataset, the "KITTI" dataset is known for exhibiting more instances of collinear motion among its sequences. Consequently, the proposed PVI-SLAM approach demonstrates a notable advantage over alternative methods when applied to the "KITTI" dataset. This advantage is more pronounced and evident, showcasing the efficacy

Table 4.5: Data sizes for sequences 06, 07, and 09 from the "KITTI" dataset extracted using ORB-SLAM3 [12].

Dataset	**06**	**07**	**09**
Total Number of Poses	412	412	677
Total Number of Features	28141	41183	56439
Total Number of Observations	151990	236482	299513

Table 4.6: Comparison between parallax angle feature parametrization and XYZ parametrization in BA.

Dataset	**06**				**07**				**09**			
parametrization	*PBA*		*SBA*		*PBA*		*SBA*		*PBA*		*SBA*	
Strategy	*GN*	*LM*	*GN*	*LM*	*GN*	*LM*	*GN*	*LM*	*GN*	*LM*	*GN*	*LM*
Initial Cost	31.994	31.994	-	31.994	207.022	207.022	-	207.021	347.115	347.115	-	347.115
Final Cost	**4.041**	4.041	-	4.066	**4.590**	5.296	-	7.410	**2.918**	2.919	-	3.254
Iteration	20	18	-	16	20	16	-	11	11	27	-	17
Time (sec)	**72.474**	83.328	-	124.916	**100.013**	103.170	-	148.443	**73.479**	91.250	-	249.308

of the proposed approach in addressing and mitigating the challenges posed by collinear motion in the feature-rich environment of the "KITTI" dataset.

For the comparative evaluation between the proposed methodology and ORB-SLAM3 [12], the data of feature observations and poses are extracted using ORB-SLAM3. As ORB-SLAM3 does not inherently support VI-SLAM with the "KITTI" dataset, monocular visual SLAM is executed instead. Specifically, sequences 06, 07, and 09 from the "KITTI" dataset are processed using ORB-SLAM3. The data sizes are summarized in Table 4.5. In the ORB-SLAM3 package, various parameters are adjusted to extract more features from a greater distance and to select more key-frames, thereby improving the information available for pre-integrated IMU data. The raw IMU data, captured at a rate of 100Hz, is utilized as input for the pre-integration method.

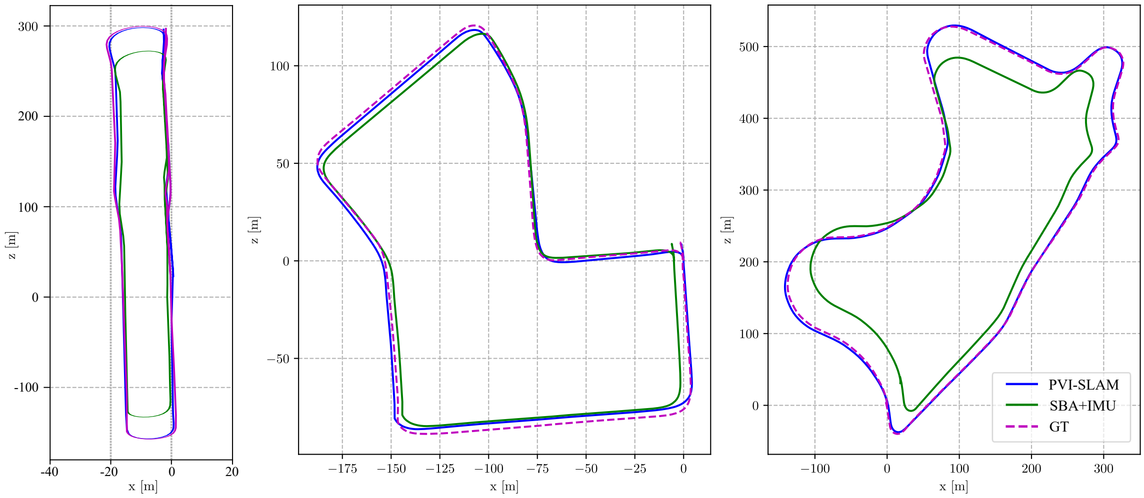
4.4.3.1 Comparison between PVI-SLAM and SBA+IMU

Visual SLAM. First, the V-SLAM results (no IMU data is used) using different feature parameterizations are evaluated. The performance of PBA is compared with SBA, which employs the XYZ parametrization (as utilized in ORB-SLAM3). Given that the outcomes of ORB-SLAM3 show no significant deviation from those of the XYZ parametrization, they are considered equivalent to the XYZ results.

For initialization, only the poses from ORB-SLAM3 are employed. The process of feature initialization relies on the observations of these features, as described in [89]. It is important to highlight that the inclusion of features extracted from greater distances presents a

TABLE 4.7: Comparison of between PVI-SLAM and SBA+IMU.

Dataset	**06**		**07**		**09**	
parametrization	*PVI-SLAM*	*SBA+IMU*	*PVI-SLAM*	*SBA+IMU*	*PVI-SLAM*	*SBA+IMU*
Strategy	*GN*	*LM*	*GN*	*LM*	*GN*	*LM*
Initial Cost	31.909	31.909	206.662	206.662	346.343	346.343
Final Cost	**4.476**	6.298	**8.246**	9.2312	**3.859**	4.648
Iteration	51	14	51	22	51	16
Time (sec)	298.383	64.468	481.897	109.230	614.530	101.337

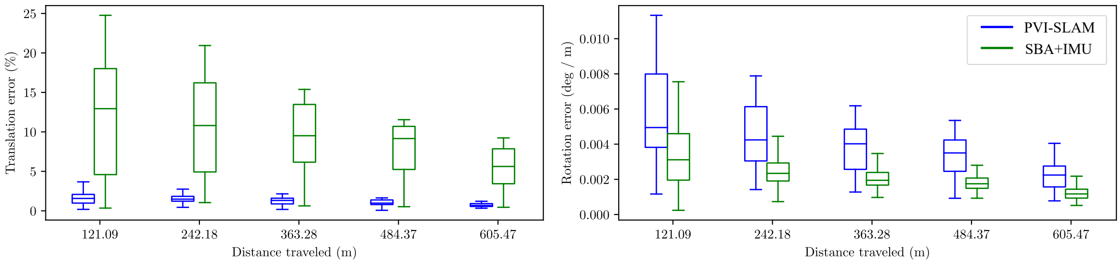


(a) KITTI-06 (b) KITTI-07 (c) KITTI-09

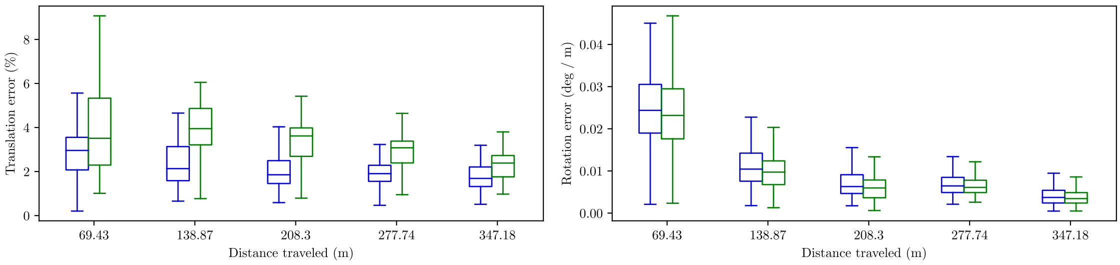
FIGURE 4.12: The comparison of trajectories between PVI-SLAM and SBA+IMU using the “KITTI” datasets.

challenge, as even with optimized poses and feature positions from ORB-SLAM3, convergence cannot be attained using the GN method. As indicated in Table 4.6, it is evident that PBA achieves convergence to a lower final cost compared to SBA across all datasets when using both GN and LM optimization techniques. Notably, SBA encounters singularity issues when applying GN.

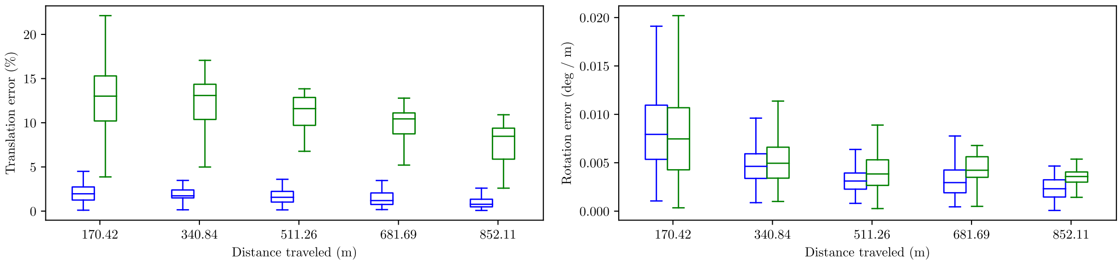
Visual-Inertial SLAM. Since the poses obtained from monocular SLAM using ORB-SLAM3 do not provide the correct metric scale, the scale of the initial pose estimates is adjusted using the provided IMU dataset. This correction aims to improve the initial guesses for the evaluation of VI-SLAM. The IMU data is pre-integrated with the camera image timestamps, following the method outlined in [1]. In addition, integrating IMU data into the system requires an extra step to initialize the state vector. The initial velocity



(a) KITTI-06



(b) KITTI-07



(c) KITTI-09

FIGURE 4.13: Comparison of translation error and rotation error between PVI-SLAM and SBA+IMU for each sequence of "KITTI" dataset.

value is computed from the pre-integrated data by propagating it accordingly. During the initialization process, sensor biases are set to zero.

To address the singularity issues, only the LM algorithm is used for SBA with IMU, while the GN algorithm is employed for PBA with IMU, as indicated in Table 4.7. Across all datasets, the final cost of PVI-SLAM converges to a lower value compared to SBA+IMU, even when starting from the same initial values. As evident in Figure 4.12 and Figure 4.13, the translation and rotation errors of PVI-SLAM are significantly smaller than those of SBA+IMU. Specifically, RMSE for PVI-SLAM across the entire trajectory is **2.405m** for sequence 06, **2.663m** for sequence 07, and **3.256m** for sequence 09. In contrast, for SBA+IMU, these values are significantly higher at 20.415m for sequence 06, 4.4316m for

TABLE 4.8: The comparison results between dead-reckoning using IMU measurement pre-integrated with PM and UGPM.

	KITTI-06		KITTI-07		KITTI-09	
	PM	***UGPM***	***PM***	***UGPM***	***PM***	***UGPM***
IMU Intial Cost	0.2062	**0.2062**	0.2619	**0.2616**	3.3362	**3.3361**
Trans.RMSE (m)	32.612	**32.091**	27.800	34.870	272.534	284.717
Rot.RMSE (deg)	12.397	**10.263**	8.054	**6.519**	27.613	56.913

sequence 07, and 34.100m for sequence 09. Furthermore, the rotation RMSE also tends to be smaller in the case of PVI-SLAM. While it may be slightly larger in the case of sequence 06, it generally remains below 0.005° per meter throughout the trajectory.

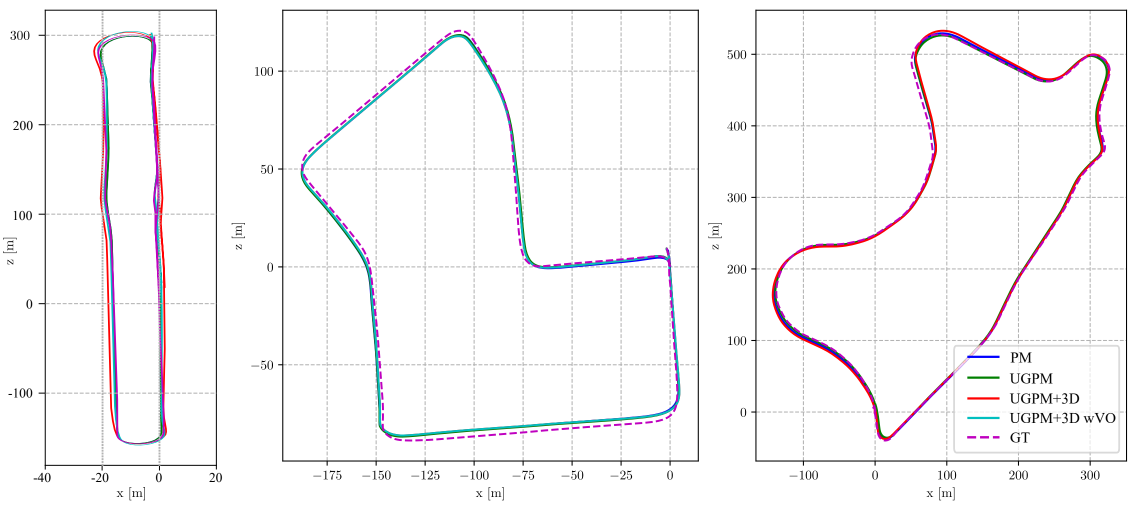
4.4.3.2 Comparison between PM and UGPM

As outlined in Section 4.2, the implementation of UGPM is aimed at enhancing system accuracy by incorporating IMU measurements within a continuous model.

During the evaluation, IMU measurements are pre-integrated using both PM and UGPM. The process of dead-reckoning utilizes these pre-integrated IMU data from both PM and UGPM. The comparative results of this evaluation are summarized in Table 4.8. Consistently, UGPM exhibits lower initial cost values when the system is initialized with the ground-truth pose compared to PM across the majority of datasets. For instance, in sequence 06, UGPM records an initial cost of 0.2062, which is identical to PM. In sequence 07, UGPM exhibits a lower initial cost of 0.2616 in contrast to PM's 0.2619. Similarly, in sequence 09, UGPM displays a lower initial cost of 3.3361 than PM's 3.3362.

Concerning the RMSE for dead-reckoning translation and rotation, sequence 06 demonstrates that UGPM outperforms PM in both aspects, with lower values of 32.091m and 10.263°, respectively, compared to PM's 32.612m and 12.397°. However, in sequence 07, only the RMSE of rotation is lower for UGPM with 6.519° compared to PM's 8.054°.

The outcomes of PVI-SLAM utilizing both PM and UGPM are illustrated in Figure 4.14. A comparison between PM and UGPM in VI-SLAM reveals similar trends, as shown in Figure 4.15. In sequence 6, PVI-SLAM with UGPM displays lower RMSE values for translation and rotation at 2.410m and 0.840°, respectively, while PM yields 2.411m and 0.841°. Additionally, UGPM exhibits a lower rotational error in sequence 09, with a value of 1.296°, compared to PM's 1.398°.



(a) KITTI-06 (b) KITTI-07 (c) KITTI-09

FIGURE 4.14: Comparing PVI-SLAM utilizing PM, UGPM, UGPM with Observation ray objective function (3D), and UGPM with 3D initialized using VO, without loop-closure (While PVI-SLAM with UGPM and 3D successfully converges in KITTI-06 and KITTI-07 when initialized with VO and without loop-closure, it encounters convergence challenges in the KITTI-09 dataset).

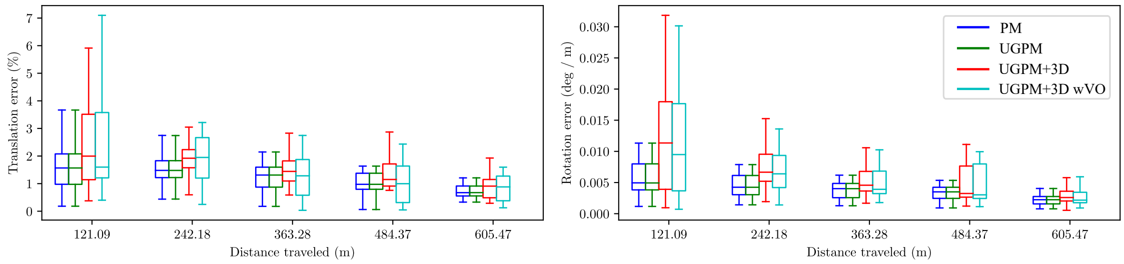
TABLE 4.9: The initial objective function for two different pose initializations— one with loop-closure and the other without loop-closure.

	KITTI-06		**KITTI-07**		**KITTI-09**	
	UV	***IMU***	***UV***	***IMU***	***UV***	***IMU***
w loop-closure	6.1756	0.333	207.021	0.335	347.118	3.255
w/o loop-closure	1466681.480	0.228	1173817.125	0.636	28529.359	3.424

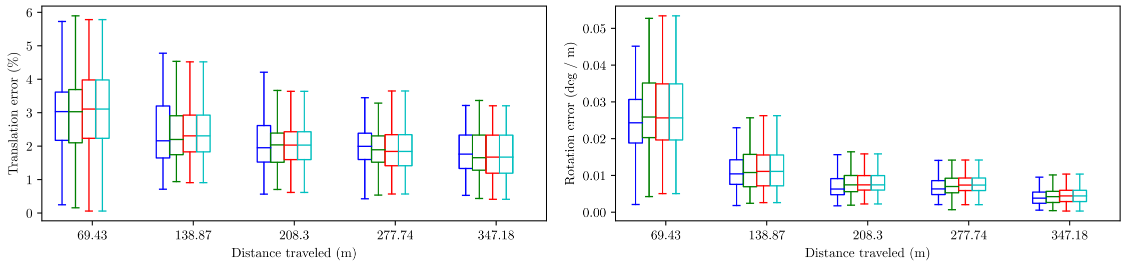
While UGPM does not exhibit significant advantages over PM with the "KITTI" dataset, which predominantly involves static motion, the notable advantage of UGPM becomes evident in more dynamic and fast-motion datasets, as indicated in [8]. As the integration of UGPM into PVI-SLAM produces comparable results to incorporating PM, UGPM is utilized in PVI-SLAM, offering potential advantages for both collinear and dynamic motion in subsequent evaluations with more suitable datasets.

4.4.3.3 Comparison between Observation Ray and Reprojection Error

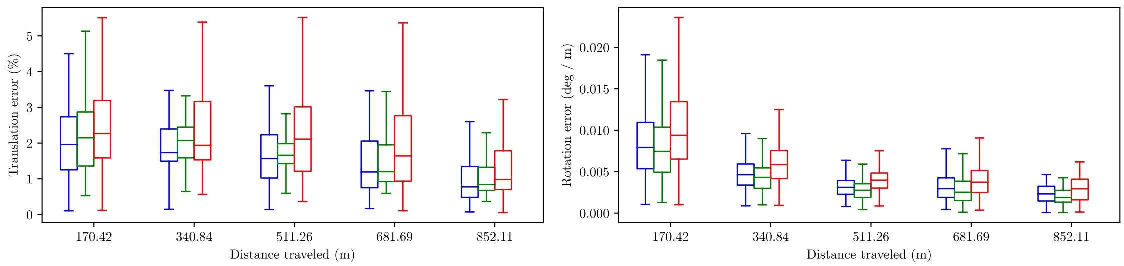
In this section, the implementation of the observation ray is carried out as described in Equation (4.31) and is then compared against the reprojection error given by Equation (4.22). To evaluate the robustness of the two objective functions further, two distinct



(a) KITTI-06



(b) KITTI-07



(c) KITTI-09

FIGURE 4.15: Comparison of translation and rotation error of PVI-SLAM utilizing PM, UGPM, UGPM with observation ray objective function (3D), and UGPM with 3D initialized using VO, without loop-closure (While PVI-SLAM with UGPM and 3D successfully converges in KITTI-06 and KITTI-07 when initialized with VO and without loop-closure, it encounters convergence challenges in the KITTI-09 dataset).

initializations are employed. The first initialization is identical to that used in the previous section (Section 4.4.3.1), obtained from ORB-SLAM3 [12]. The second initialization involves poses obtained from pure VO without closing the loop. The objective function calculated using the observation ray is subsequently converted back to reprojection error only for comparative analysis. The disparities in the initial cost for these two different initializations are presented in Table 4.9. Due to the absence of loop-closure, a substantial difference in image reprojection error is observed.

When utilizing the observation ray as the objective function for the initial pose with loop-closure, it does not demonstrate improvement over using the reprojection error objective function. However, attempting to optimize with the initial pose without loop-closure using the reprojection error objective function results in system optimization failure, leading to singularity in all datasets. In contrast, PVI-SLAM using the observation ray objective function manages to converge with the initial pose without loop-closure to a solution comparable to the one using loop-closed pose initialization, as depicted in Figure 4.14 and Figure 4.15.

For PVI-SLAM utilizing UGPM and observation ray objective function without loop-closure, the achieved translation RMSE is 2.949m and 2.699m in sequence 06 and 07, respectively, with corresponding rotational errors of 1.531° and 1.422°. On the other hand, PVI-SLAM utilizing UGPM and observation ray objective function with loop-closure yields translation errors of 2.632m and 2.699m, along with rotational errors of 1.387° and 1.422° for the same sequences. In the case of sequence 09, the system successfully converges with a good initial value when loop-closure is incorporated. However, it fails to converge even with the utilization of the observation ray objective function when initialized from VO without loop-closure.

4.5 Summary

This chapter introduces and evaluates VI-SLAM based on PBA with pre-integrated IMU data (PVI-SLAM). The incorporation of IMU into pure V-SLAM corrects the unknown scale from the monocular camera, substantially enhancing the reliability and consistency of the system, even with fewer feature observations.

Leveraging the "EuRoC" dataset, PVI-SLAM exhibits superior performance compared to state-of-the-art approaches (VINS-Fusion [10], OpenVINS [11], and ORB-SLAM3 [12]). To underscore the advantages of employing the parallax parameterization, the evaluation extends to the "KITTI" dataset. The primary challenge addressed revolves around the singularity issue encountered when utilizing SBA+IMU as used in ORB-SLAM3, especially with features located at a greater distance. PVI-SLAM is demonstrated to effectively address this challenge. In terms of convergence properties and accuracy, PVI-SLAM outperforms SBA+IMU, both with and without IMU data integration. Additionally, to further enhance the system, UGPM is implemented to harness benefits in both collinear and dynamic motion by handling IMU measurements in a continuous model. Furthermore, the incorporation of the observation ray enhances the system's robustness, enabling convergence even in trajectories without loop-closure from VO when the reprojection error objective function fails to converge.

However, it is crucial to note that the proposed method PVI-SLAM may not always guarantee convergence, especially when dealing with a large number of frames. The convergence of this high-dimensional nonlinear optimization problem is not assured. Additionally, considering the computational complexity is important when handling batch nonlinear optimization for online system implementation. Therefore, the next chapter introduces the linear map joining method to address these challenges.

Chapter 5

Linear Submap Joining using Parallax VI-SLAM

In the field of SLAM, dealing with high-dimensional nonlinear optimization problems is inherently challenging. The previous chapter (Chapter 4) underscores the crucial importance of precise initial values for achieving successful convergence in nonlinear optimization problems. However, even with the provision of accurate initial values, there is no guarantee of converging to the global minimum.

To address these issues, **this chapter introduces a Linear Submap Joining method using the Linear SLAM framework applied to the proposed PVI-SLAM methodology**. Instead of retaining all the data and undergoing full nonlinear optimization, which is often impractical, this method optimizes small parts of the full dataset as local maps, utilizing information relevant to each specific local map. Subsequently, the information from each optimized local map is fused through the map joining process to construct a unified map. This is particularly beneficial in situations where computational resources are limited and helps mitigate issues related to local minima.

An evaluation is performed using publicly available real datasets, such as "EuRoC" and "KITTI". The performance of Linear SLAM, which is built upon local maps optimized using PVI-SLAM, is demonstrated, showcasing close proximity to solutions achievable through a full nonlinear optimization algorithm from an accurate initial guess. Notably, the evaluation emphasizes the effectiveness of the method in addressing challenges related to poor initial values, situations that would typically lead to convergence failure in the context of full nonlinear optimization.

5.1 Integrating PVI-SLAM with Linear Submap Joining

Large-scale maps are effectively managed by combining submaps, as demonstrated in [80–83]. Most of these approaches, such as [84] by Huang et al., avoid marginalizing any states and treat the estimated state of each local map as integrated observations during the map joining process. Another notable work by Zhao et al. [85] presents a map joining algorithm that transforms a nonlinear optimization problem into a combination of LLS optimization and nonlinear coordinate transformation. This algorithm eliminates the need for initial guesses or iterative procedures since LLS problems can be resolved using closed-form formulas.

In this section, **PVI-SLAM with Linear Submap Joining algorithms** is proposed to resolve the problem of high computational cost balancing with estimation accuracy. To perform Linear SLAM framework [85], a structured three-step procedure is required for addressing large-scale VI-SLAM challenges. Firstly, each local map is independently built using local information by solving a small-scale VI-SLAM problem through PVI-SLAM (Section 4.3). Secondly, to integrate into the Linear SLAM framework, it is essential to transform the structure of the state vector, which in turn requires a recalculation of the information matrix for the system. Finally, submap joining can be carried out, primarily through solving LLS and conducting nonlinear coordinate transformations.

5.1.1 Local Visual-Inertial SLAM

As illustrated in Figure 5.1, the Linear Submap Joining process begins by optimizing each of the local maps. For simplicity in this chapter, only two local maps are considered: Local map 1, denoted as $\mathcal{X}_W^{L_1}$, and local map 2, denoted as $\mathcal{X}_W^{L_2}$, are expressed as:

$$\begin{aligned} \mathcal{X}_W^{L_1} &= \left[{}^W\mathcal{P}_{I_1}, \cdots, {}^W\mathcal{P}_{I_p}, \mathcal{F}_1^{L_1}, \mathcal{F}_{12}^{L_1} \right] \\ \mathcal{X}_W^{L_2} &= \left[{}^W\mathcal{P}_{I_p}, \cdots, {}^W\mathcal{P}_{I_q}, \mathcal{F}_2^{L_2}, \mathcal{F}_{12}^{L_2} \right], \end{aligned} \tag{5.1}$$

where $\mathcal{F}_1^{L_1}$ and $\mathcal{F}_2^{L_2}$ represent features unique to each local map, and $\mathcal{F}_{12}^{L_1}$ and $\mathcal{F}_{12}^{L_2}$ denote features that are common between the two local maps. All features are represented in the form of parallax parametrization. Both local maps are in the coordinate frame of the world frame, $\{W\}$, defined by the first pose of the state as the origin. The pose, ${}^W\mathcal{P}_{I_i}$, stay same as in Chapter 4:

$${}^W\mathcal{P}_{I_i} = \left\{ \mathbf{R}_{I_i}^W, {}^W\mathbf{t}_{I_i}, {}^W\mathbf{v}_{I_i}, \mathbf{b}_{\omega_i}, \mathbf{b}_{a_i} \right\}. \tag{5.2}$$

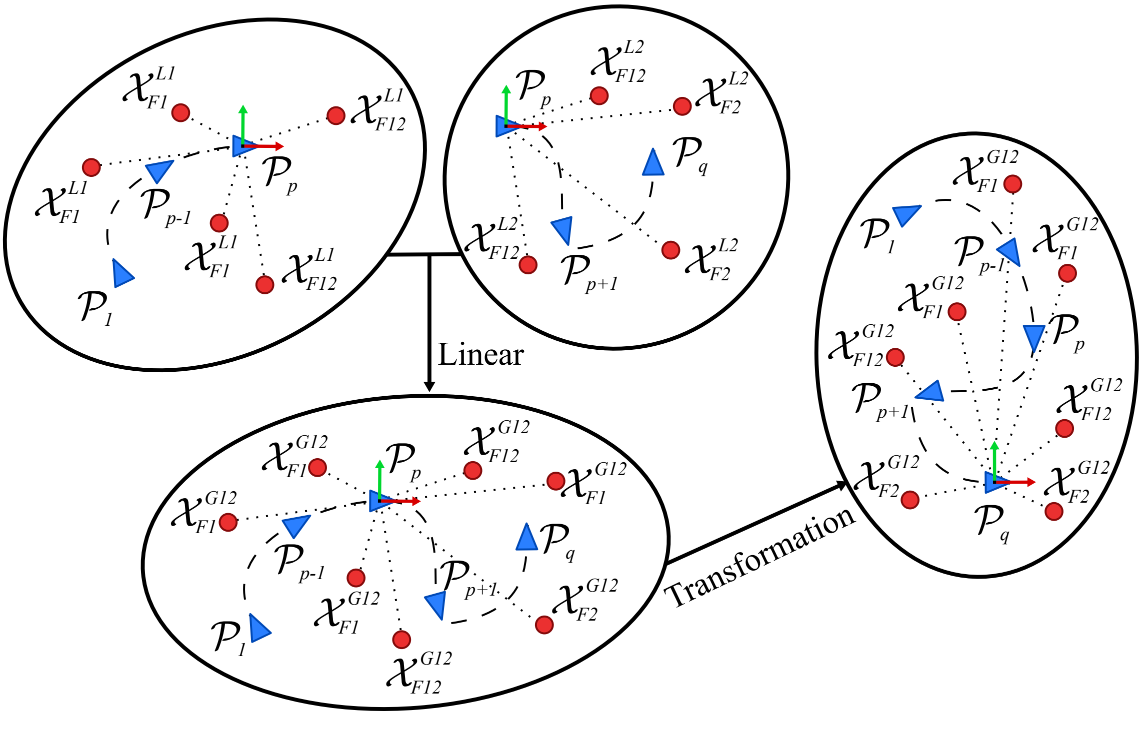


FIGURE 5.1: Linear way of Map Joining.

Subsequently, each local map can be optimized through PVI-SLAM in Section 4.3, leading to the estimated local maps, $\hat{\mathcal{X}}_W^{L_1}$ and $\hat{\mathcal{X}}_W^{L_2}$.

5.1.2 Structural Transformation

After performing PVI-SLAM, the corresponding information matrix is also required as an input for Linear SLAM. However, to align with the requirements of Linear SLAM, adjustments need to be made to the parameters related to poses and feature positions. The information matrix from PVI-SLAM cannot be directly used, necessitating modifications in this process. In this adjustment, ${}^W\hat{\mathbf{v}}_{I_i}$, $\hat{\mathbf{b}}_{\omega_i}$, and $\hat{\mathbf{b}}_{a_i}$ are removed from the state vector. For each pose, the rotation matrix, $\mathbf{R}_{I_i}^W$, is converted into Euler angle, ${}^W\mathbf{r}_{I_i}$. The transformation of feature positions, $\mathbf{f}_j$, into XYZ parameters can be achieved by the following process:

$$\mathbf{f}_j^{XYZ} = d_j \mathbf{x}_j^m + {}^W\mathbf{t}_{C_m}, \tag{5.3}$$

where d_j is the depth of the feature $\mathbf{f}_j$ from the main anchor ${}^W\mathbf{t}_{C_m}$. By utilizing the angles ω and φ as specified in Equation (4.27) and Equation (4.30), respectively, the depth can be calculated as:

$$d_j = \frac{\sin(\omega_j + \varphi_j)}{\sin \omega_j} \left\| {}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m} \right\| . \tag{5.4}$$

Now the state vector of both local maps can be re-written as:

$$\begin{aligned} \hat{\mathcal{X}}_W^{L_1} &= \left[{}^W\hat{\mathcal{P}}_{I_1}, \cdots, {}^W\hat{\mathcal{P}}_{I_p}, \hat{\mathcal{F}}_1^{L_1}, \hat{\mathcal{F}}_{12}^{L_1} \right] \\ \hat{\mathcal{X}}_W^{L_2} &= \left[{}^W\hat{\mathcal{P}}_{I_p}, \cdots, {}^W\hat{\mathcal{P}}_{I_q}, \hat{\mathcal{F}}_2^{L_2}, \hat{\mathcal{F}}_{12}^{L_2} \right], \end{aligned} \tag{5.5}$$

where ${}^W\hat{\mathcal{P}}_{I_i} = \left[{}^W\hat{\mathbf{t}}_{I_i}, {}^W\hat{\mathbf{r}}_{I_i} \right]$ and $\hat{\mathcal{F}}_k^L = \left[\hat{\mathbf{f}}_1^{XYZ}, \cdots, \hat{\mathbf{f}}_M^{XYZ} \right]$ represent all the feature positions in XYZ parametrization. To perform Linear Submap Joining, both local maps are then transformed into the coordinate frame of the start pose of each local map, $\{1\}$ and $\{p\}$, respectively:

$$\hat{\mathcal{X}}_1^{L_1} = \left[{}^{I_1}\hat{\mathbf{t}}_{I_2}, {}^{I_1}\hat{\mathbf{r}}_{I_2}, \cdots, {}^{I_1}\hat{\mathbf{t}}_{I_p}, {}^{I_1}\hat{\mathbf{r}}_{I_p}, \hat{\mathcal{F}}_1^{L_1}, \hat{\mathcal{F}}_{12}^{L_1} \right], \tag{5.6}$$

$$\hat{\mathcal{X}}_p^{L_2} = \left[{}^{I_p}\hat{\mathbf{t}}_{I_{(p+1)}}, {}^{I_p}\hat{\mathbf{r}}_{I_{(p+1)}}, \cdots, {}^{I_p}\hat{\mathbf{t}}_{I_q}, {}^{I_p}\hat{\mathbf{r}}_{I_q}, \hat{\mathcal{F}}_2^{L_2}, \hat{\mathcal{F}}_{12}^{L_2} \right]. \tag{5.7}$$

Subsequently, the information matrices for each local, denoted as $\mathbf{I}_1^{L_1}$ and $\mathbf{I}_p^{L_2}$, are recalculated based on the state vector in Equation (5.6) and Equation (5.7). This process involves utilizing the cost function defined in Equation (4.21), which still requires ${}^W\hat{\mathbf{v}}_{I_i}$, $\hat{\mathbf{b}}_{\omega_i}$, and $\hat{\mathbf{b}}_{a_i}$ as constants for the residual. The final form of the state vector for each local map can be expressed as follows:

$$\begin{aligned} \mathcal{M}^{L_1} &= \left(\hat{\mathcal{X}}_1^{L_1}, \mathbf{I}_1^{L_1} \right), \\ \mathcal{M}^{L_2} &= \left(\hat{\mathcal{X}}_p^{L_2}, \mathbf{I}_p^{L_2} \right). \end{aligned} \tag{5.8}$$

5.1.3 Linear SLAM: Map Joining

When the two local maps are given to perform Linear Submap Joining, the first map, $\hat{\mathcal{X}}_1^{L_1}$, need to be transformed into the coordinate frame of the last pose, $\hat{\mathcal{X}}_p^{L_1}$, as in Figure 5.1:

$$\hat{\mathcal{X}}_p^{L_1} = \left[{}^{I_p}\hat{\mathbf{t}}_{I_1}, {}^{I_p}\hat{\mathbf{r}}_{I_1}, \cdots, {}^{I_p}\hat{\mathbf{t}}_{I_{p-1}}, {}^{I_p}\hat{\mathbf{r}}_{I_{p-1}}, \hat{\mathcal{F}}_1^{L_1}, \hat{\mathcal{F}}_{12}^{L_1} \right]. \tag{5.9}$$

The corresponding information matrix is recalculated through the following process:

$$\mathbf{I}_p^{L_1} = \nabla_p^T \mathbf{I}_1^{L_1} \nabla_p, \tag{5.10}$$

where ∇_p is the Jacobian of $\mathcal{X}_1^{L_1}$ with respect to $\mathcal{X}_p^{L_1}$ evaluated at $\hat{\mathcal{X}}_p^{L_1}$ as:

$$\nabla_q = \left.\frac{\partial \mathcal{X}_1^{L_1}}{\partial \mathcal{X}_p^{L_1}}\right|_{\hat{\mathcal{X}}_p^{L_1}} . \tag{5.11}$$

Then, the following two local maps can be achieved in the coordinated frame of $\{p\}$:

$$\begin{aligned} \hat{\mathcal{X}}_p^{L_1} &= \left[{}^{I_p}\hat{\mathbf{t}}_{I_1}, {}^{I_p}\hat{\mathbf{r}}_{I_1}, \cdots, {}^{I_p}\hat{\mathbf{t}}_{I_{p-1}}, {}^{I_p}\hat{\mathbf{r}}_{I_{p-1}}, \hat{\mathcal{F}}_1^{L_1}, \hat{\mathcal{F}}_{12}^{L_1}\right], \\ \hat{\mathcal{X}}_p^{L_2} &= \left[{}^{I_p}\hat{\mathbf{t}}_{I_{p+1}}, {}^{I_p}\hat{\mathbf{r}}_{I_{p+1}}, \cdots, {}^{I_p}\hat{\mathbf{t}}_{I_q}, {}^{I_p}\hat{\mathbf{r}}_{I_q}, \hat{\mathcal{F}}_2^{L_2}, \hat{\mathcal{F}}_{12}^{L_2}\right]. \end{aligned} \tag{5.12}$$

By combining two local maps (Equation (5.12)), the state vector of the integrated map, $\mathcal{M}^{G_{12}}$, can be obtained in the coordinate frame of $\{q\}$ as:

$$\begin{aligned} \mathcal{X}_q^{G_{12}} &= \left[{}^{I_q}\mathcal{P}_{I_1}, \cdots, {}^{I_q}\mathcal{P}_{I_p}, {}^{I_q}\mathcal{P}_{I_{p+1}}, \cdots, {}^{I_q}\mathcal{P}_{I_{q-1}}, \mathcal{F}_1^G, \mathcal{F}_2^G, \mathcal{F}_{12}^G\right] \\ &= \left[{}^{I_q}\mathbf{t}_{I_1}, {}^{I_q}\mathbf{r}_{I_1}, \cdots, {}^{I_q}\mathbf{t}_{I_p}, {}^{I_q}\mathbf{r}_{I_p}, {}^{I_q}\mathbf{t}_{I_{p+1}}, {}^{I_q}\mathbf{r}_{I_{p+1}}, \cdots, {}^{I_q}\mathbf{t}_{I_{q-1}}, {}^{I_q}\mathbf{r}_{I_{q-1}}, \mathcal{F}_1^G, \mathcal{F}_2^G, \mathcal{F}_{12}^G\right]. \end{aligned} \tag{5.13}$$

This can be optimized directly by minimizing the following objective function, similar to the approach taken by Huang et al. [105]:

$$\begin{aligned} f\left(\mathcal{X}^{G_{12}}\right) &= \|\mathbf{e}_1\|^2_{\mathbf{I}_p^{L_1}} + \|\mathbf{e}_2\|^2_{\mathbf{I}_p^{L_2}} \\ &= \left\| \begin{bmatrix} \mathbf{R}_{I_q}^{I_p}\left({}^{I_q}\mathbf{t}_{I_1} - {}^{I_q}\mathbf{t}_{I_p}\right) - {}^{I_p}\hat{\mathbf{t}}_{I_1} \\ r\left(\mathbf{R}_{I_q}^{I_1}(\mathbf{R}_{I_q}^{I_p})^\top\right) - {}^{I_p}\hat{\mathbf{r}}_{I_1} \\ \vdots \\ \mathbf{R}_{I_q}^{I_p}\left({}^{I_q}\mathbf{t}_{I_{p-1}} - {}^{I_q}\mathbf{t}_{I_p}\right) - {}^{I_p}\hat{\mathbf{t}}_{I_{p-1}} \\ r\left(\mathbf{R}_{I_q}^{I_{p-1}}(\mathbf{R}_{I_q}^{I_p})^\top\right) - {}^{I_p}\hat{\mathbf{r}}_{I_{p-1}} \\ \mathbf{R}_{I_q}^{I_p}\left(\mathcal{F}_1^G - {}^{I_q}\mathbf{t}_{I_p}\right) - \hat{\mathcal{F}}_1^{L_1} \\ \mathbf{R}_{I_q}^{I_p}\left(\mathcal{F}_{12}^G - {}^{I_q}\mathbf{t}_{I_p}\right) - \hat{\mathcal{F}}_{12}^{L_1} \end{bmatrix} \right\|^2_{\mathbf{I}_p^{L_1}} + \left\| \begin{bmatrix} \mathbf{R}_{I_q}^{I_p}\left({}^{I_q}\mathbf{t}_{I_{p+1}} - {}^{I_q}\mathbf{t}_{I_p}\right) - {}^{I_p}\hat{\mathbf{t}}_{I_{p+1}} \\ r\left(\mathbf{R}_{I_q}^{I_{p+1}}(\mathbf{R}_{I_q}^{I_p})^\top\right) - {}^{I_p}\hat{\mathbf{r}}_{I_{p+1}} \\ \vdots \\ -\mathbf{R}_{I_q}^{I_p}{}^{I_q}\mathbf{t}_{I_p} - {}^{I_p}\hat{\mathbf{t}}_{I_q} \\ r\left((\mathbf{R}_{I_q}^{I_p})^\top\right) - {}^{I_p}\hat{\mathbf{r}}_{I_q} \\ \mathbf{R}_{I_q}^{I_p}\left(\mathcal{F}_2^G - {}^{I_q}\mathbf{t}_{I_p}\right) - \hat{\mathcal{F}}_2^{L_2} \\ \mathbf{R}_{I_q}^{I_p}\left(\mathcal{F}_{12}^G - {}^{I_q}\mathbf{t}_{I_p}\right) - \hat{\mathcal{F}}_{12}^{L_2} \end{bmatrix} \right\|^2_{\mathbf{I}_p^{L_2}} \end{aligned} \tag{5.14}$$

where $r(\cdot)$ is the function that converts a rotation matrix to Euler angles. However, given that this involves a NLLS problem, successful application of this approach necessitates a reliable initial guess and iterative optimization to find a solution. Consequently, in the context of Linear SLAM [85], the state vector of $\mathcal{X}_q^{G12}$ undergoes redefinition:

$$
\begin{aligned}
{}^{p}\overline{\mathbf{t}}_1 &= \mathbf{R}_{I_q}^{I_p}\left({}^{I_q}\mathbf{t}_{I_1} - {}^{I_q}\mathbf{t}_{I_p}\right), \quad {}^{p}\overline{\mathbf{r}}_1 = r\left(\mathbf{R}_{I_q}^{I_1}(\mathbf{R}_{I_q}^{I_p})^\top\right), \\
&\vdots \\
{}^{p}\overline{\mathbf{t}}_{p-1} &= \mathbf{R}_{I_q}^{I_p}\left({}^{I_q}\mathbf{t}_{I_{p-1}} - {}^{I_q}\mathbf{t}_{I_p}\right), \quad {}^{p}\overline{\mathbf{r}}_{p-1} = r\left(\mathbf{R}_{I_q}^{I_{p-1}}(\mathbf{R}_{I_q}^{I_p})^\top\right), \\
{}^{p}\overline{\mathbf{t}}_{p+1} &= \mathbf{R}_{I_q}^{I_p}\left({}^{I_q}\mathbf{t}_{I_{p+1}} - {}^{I_q}\mathbf{t}_{I_p}\right), \quad {}^{p}\overline{\mathbf{r}}_{p+1} = r\left(\mathbf{R}_{I_q}^{I_{p+1}}(\mathbf{R}_{I_q}^{I_p})^\top\right), \\
&\vdots \\
{}^{p}\overline{\mathbf{t}}_q &= -\mathbf{R}_{I_q}^{I_p}{}^{I_q}\mathbf{t}_{I_p}, \quad {}^{p}\overline{\mathbf{r}}_q = r\left((\mathbf{R}_{I_q}^{I_p})^\top\right),
\end{aligned} \tag{5.15}
$$

and

$$
\begin{aligned}
\overline{\mathcal{F}}_1^{G_{12}} &= \mathbf{R}_{I_q}^{I_p}\left(\mathcal{F}_1^{L_1} - {}^{I_q}\mathbf{t}_{I_p}\right), \\
\overline{\mathcal{F}}_2^{G_{12}} &= \mathbf{R}_{I_q}^{I_p}\left(\mathcal{F}_1^{L_2} - {}^{I_q}\mathbf{t}_{I_p}\right), \\
\overline{\mathcal{F}}_{12}^{G_{12}} &= \mathbf{R}_{I_q}^{I_p}\left(\mathcal{F}_1^{G_{12}} - {}^{I_q}\mathbf{t}_{I_p}\right).
\end{aligned} \tag{5.16}
$$

Then, the new state vector, $\overline{\mathcal{X}}_p^{G_{12}}$, in the frame of $\{p\}$, as can be seen in Figure 5.1, can be written:

$$
\begin{aligned}
\overline{\mathcal{X}}_p^{G_{12}} &= \left[{}^{p}\overline{\mathbf{t}}_1, {}^{p}\overline{\mathbf{r}}_1, \cdots, {}^{p}\overline{\mathbf{t}}_{p-1}, {}^{p}\overline{\mathbf{r}}_{p-1}, {}^{p}\overline{\mathbf{t}}_{p+1}, {}^{p}\overline{\mathbf{r}}_{p+1}, \cdots, {}^{p}\overline{\mathbf{t}}_q, {}^{p}\overline{\mathbf{r}}_q, \overline{\mathcal{F}}_1^{G_{12}}, \overline{\mathcal{F}}_2^{G_{12}}, \overline{\mathcal{F}}_{12}^{G_{12}}\right] \\
&= g\left(\mathcal{X}_q^{G_{12}}\right),
\end{aligned} \tag{5.17}
$$

where $g(\cdot)$ serves as the transformation function. In this way, instead of facing a NLLS problem directly, Equation (5.14) is transformed into a LLS problem:

$$
\bar{f}\left(\overline{\mathcal{X}}_p^{G_{12}}\right) = \left\| \begin{bmatrix} {}^{p}\overline{\mathbf{t}}_1 - {}^{p}\hat{\mathbf{t}}_1 \\ {}^{p}\overline{\mathbf{r}}_1 - {}^{p}\hat{\mathbf{r}}_1 \\ \vdots \\ {}^{p}\overline{\mathbf{t}}_{(p-1)} - {}^{p}\hat{\mathbf{t}}_{(p-1)} \\ {}^{p}\overline{\mathbf{r}}_{(p-1)} - {}^{p}\hat{\mathbf{r}}_{(p-1)} \\ \overline{\mathcal{F}}_1^{G_{12}} - \hat{\mathcal{F}}_1^{L_1} \\ \overline{\mathcal{F}}_{12}^{G_{12}} - \hat{\mathcal{F}}_{12}^{L_1} \end{bmatrix} \right\|_{\mathbf{I}_p^{L_1}}^2 + \left\| \begin{bmatrix} {}^{p}\overline{\mathbf{t}}_1 - {}^{p}\hat{\mathbf{t}}_{(p+1)} \\ {}^{p}\overline{\mathbf{r}}_{(p+1)} - {}^{p}\hat{\mathbf{r}}_{(p+1)} \\ \vdots \\ {}^{p}\overline{\mathbf{t}}_q - {}^{p}\hat{\mathbf{t}}_q \\ {}^{p}\overline{\mathbf{r}}_q - {}^{p}\hat{\mathbf{r}}_q \\ \overline{\mathcal{F}}_2^{G_{12}} - \hat{\mathcal{F}}_2^{L_2} \\ \overline{\mathcal{F}}_{12}^{G_{12}} - \hat{\mathcal{F}}_{12}^{L_2} \end{bmatrix} \right\|_{\mathbf{I}_p^{L_2}}^2 . \tag{5.18}
$$

The optimal solution for a joined map can be achieved by solving the sparse linear equation as follows:

$$
\text{minimize}\, \bar{f}\left(\overline{\mathcal{X}}_p^{G_{12}}\right) = \left\| \mathbf{A}\overline{\mathcal{X}}_p^{G_{12}} - \mathbf{Z} \right\|_{I_Z}^2 \tag{5.19}
$$

$$
\mathbf{A}^\top \mathbf{I}_Z \mathbf{A} \overline{\mathcal{X}}_p^{\hat{G}_{12}} = \mathbf{A}^\top \mathbf{I}_Z \mathbf{Z} \tag{5.20}
$$

$$\bar{\mathbf{I}}_p^{G_{12}} = \mathbf{A}^\top \mathbf{I}_Z \mathbf{A}, \tag{5.21}$$

where $\mathbf{Z} = \left[\hat{\mathcal{X}}_p^{L_1}, \hat{\mathcal{X}}_p^{L_2}\right]$, $\mathbf{I}_Z = \text{diag}\left(\mathbf{I}_p^{L_1}, \mathbf{I}_p^{L_2}\right)$ and $\mathbf{A}$ is the coefficient matrix of Equation (5.18).

When the optimal solution $\hat{\overline{\mathcal{X}}}_p^{G12}$ is obtained, the nonlinear coordinate transformation can be applied to revert to the form presented in Equation (5.13) by:

$$\hat{\mathcal{X}}_q^{G_{12}} = g^{-1}\left(\hat{\overline{\mathcal{X}}}_p^{G_{12}}\right). \tag{5.22}$$

The corresponding information matrix can also be obtained through the following process:

$$\mathbf{I}_q^{G_{12}} = \nabla_q^T \, \bar{\mathbf{I}}_p^{G_{12}} \, \nabla_q, \tag{5.23}$$

where ∇_q is the Jacobian of $\overline{\mathcal{X}}_p^{G_{12}}$ with respect to $\mathcal{X}_q^{G_{12}}$ evaluated at $\hat{\mathcal{X}}_q^{G_{12}}$ as:

$$\nabla_q = \left.\frac{\partial g\left(\mathcal{X}_q^{G_{12}}\right)}{\partial \mathcal{X}_q^{G_{12}}}\right|_{\hat{\mathcal{X}}_q^{G_{12}}}. \tag{5.24}$$

5.1.4 Sequenced Local Map Joining

When a sequence of local maps is required to be joined, the "Divide and Conquer" strategy proposed by Zhao et al. in [85] can be applied by repeating the same procedure outlined in Section 5.1. In contrast to the traditional approach to map joining, where each local map is initially in the frame of its first pose and remains in that frame even after joining, Linear SLAM [85] maintains that the first local map is always in the frame of its last pose, and the second map is in the frame of its start pose. After map joining, the resulting map is always in the frame of its last pose. This allows for a "Divide and Conquer" process directly, leading to additional computational cost savings.

5.2 Performance Evaluation on Linear Submap Joining utilizing PVI-SLAM

This section evaluates the robustness of the Linear Submap Joining framework within the context of PVI-SLAM. The assessment begins with a comparative analysis of the computational time required for the proposed Map Joining process with PVI-SLAM, considering different numbers of local maps. Subsequently, the estimated trajectory of Linear Submap

Table 5.1: Total computation time for Linear Submap Joining process for "EuRoC" dataset.

Dataset	V101			MH01			MH03		
Num of Local maps	**2**	**4**	**8**	**2**	**4**	**8**	**2**	**4**	**8**
Local Maps	159.536	81.330	31.331	220.537	102.893	38.985	21.013	21.193	19.285
Structure Transformation	2.970	3.241	3.669	3.808	3.858	4.335	2.821	3.095	3.549
Linear Submap Joining	5.297	9.768	19.369	5.866	15.874	34.134	4.582	9.899	17.021
Total Time (sec)	167.803	94.339	**54.369**	230.211	122.625	**77.454**	**28.416**	34.187	39.855

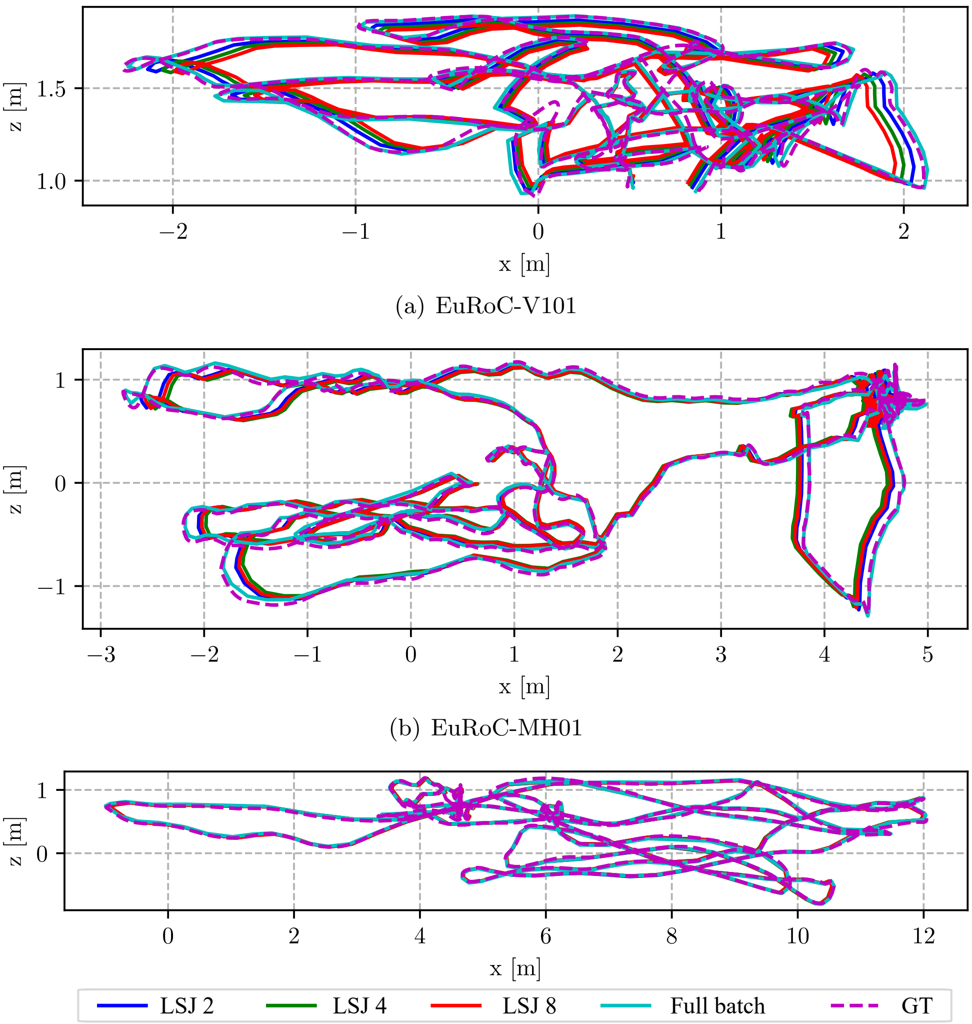
Joining is compared with the full NLLS problem solved by PVI-SLAM. Experiments are conducted using the "EuRoC" and "KITTI" datasets. In the case of the "KITTI" dataset, various initialization strategies, as discussed in Chapter 4, are applied to evaluate further the robustness of PVI-SLAM utilizing Linear Submap Joining.

5.2.1 "EuRoC" Dataset

Table 5.1 displays the overall processing time for Linear Submap Joining, varying with the number of local maps. The time dedicated to optimizing local maps decreases with an increasing number of local maps. However, there is a concurrent rise in the time required for structure transformation and executing Linear Submap Joining. Despite this, notable time savings persist across various scenarios. For example, the V101 dataset, initially taking 167.80 seconds with 2 local maps, significantly reduces to just 54.37 seconds with 8 local maps. Similarly, for the MH01 dataset, the time decreases from 230.21 seconds with 2 local maps to 77.45 seconds with 8 local maps.

The results of the estimated trajectory for Linear Submap Joining are depicted in Figure 5.2 and Figure 5.3. It is evident that Linear Submap Joining can achieve accuracy close to that of full batch optimization. For the V101 dataset, the translation error increases with the growing number of local maps, resulting in RMSE values of 0.086m, 0.143m, and 0.191m, while the rotation error remains relatively consistent across all cases. In the case of MH03, Linear Submap Joining outperforms full batch optimization, yielding RMSE values of 0.025m, 0.033m, and 0.034m with 2, 4, and 8 local maps, respectively. In contrast, full batch optimization achieves a translation error RMSE of 0.030m and the lowest rotational error RMSE at 1.186°, surpassing Linear Submap Joining with rotational errors of 1.229°, 1.215°, and 1.241°.

Furthermore, Linear Submap Joining demonstrates comparable performance to state-of-the-art methods such as OpenVINS [11], ORB-SLAM3 [12], and VINS-Fusion [10], as illustrated in Figure 5.4 and Figure 5.5. In the case of MH03, Linear Submap Joining



(a) EuRoC-V101

(b) EuRoC-MH01

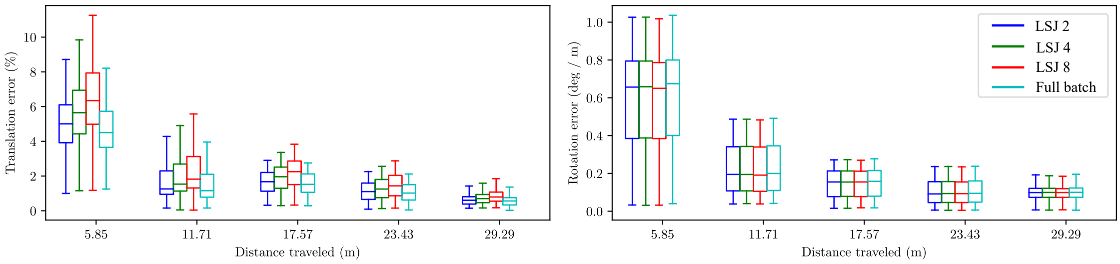
(c) EuRoC-MH03

FIGURE 5.2: Comparing Trajectories: Full Batch PVI-SLAM vs Linear Submap Joining (LSJ) with Varying Numbers (2,4, and 8) of Local Maps in the "EuRoC" Datasets.

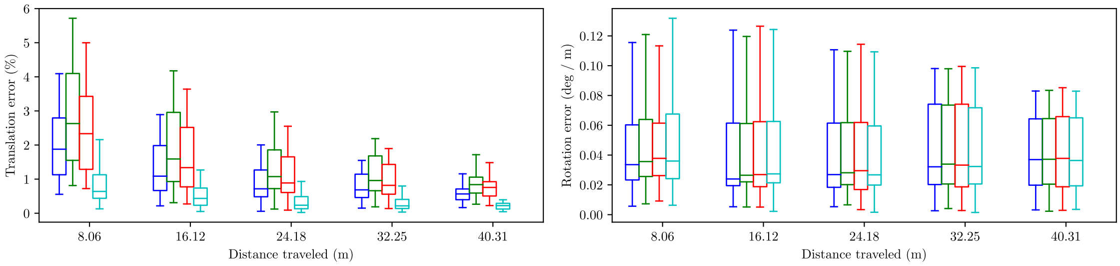
achieves the lowest translation and rotational error RMSE values at **0.025**m and **1.229°**, respectively. In comparison, OpenVINS, ORB-SLAM3, and VINS-Fusion exhibit RMSE values of 0.108m and 1.417°, 0.034m and 1.338°, and 0.078m and 1.640°, respectively.

5.2.2 "KITTI" Dataset

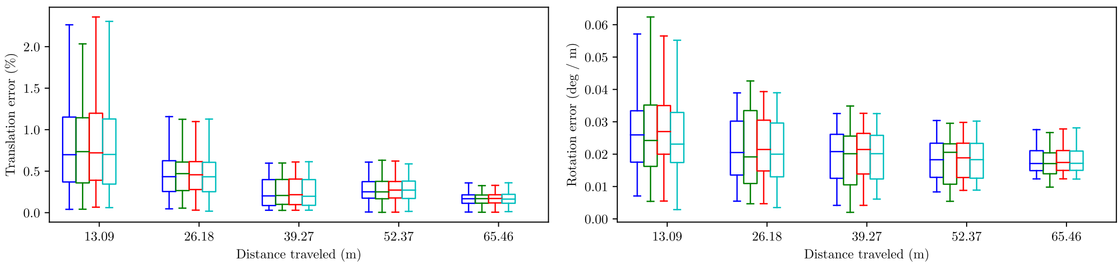
In the case of the "KITTI" dataset, local maps are constructed using PVI-SLAM with two different initializations—one with loop-closure from ORB-SLAM3 [12] and the other with VO without loop-closure.



(a) EuRoC-V101



(b) EuRoC-MH01

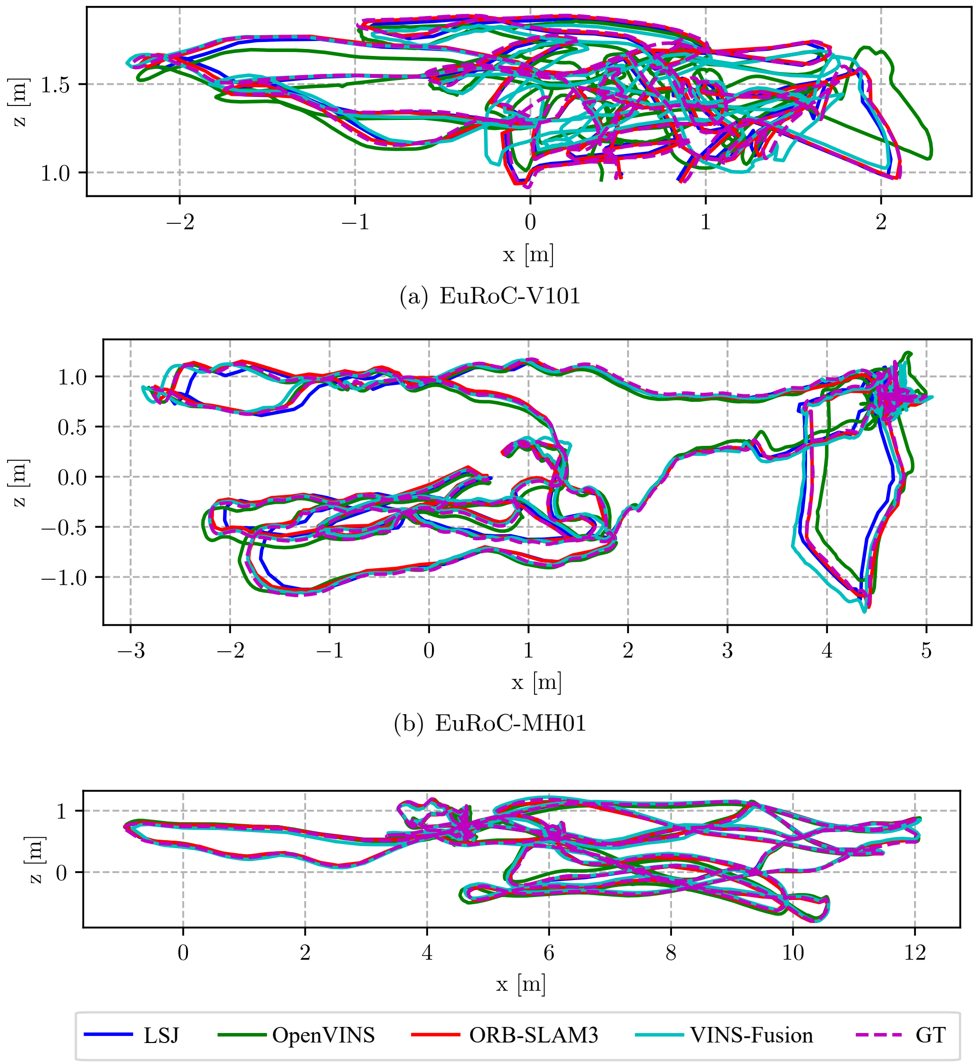


(c) EuRoC-MH03

FIGURE 5.3: Comparative Analysis of Translation and Rotation Errors: Full Batch PVI-SLAM vs. Linear Submap Joining (LSJ) with Different Local Map Configurations (2,4, and 8) across "EuRoC" Datasets.

5.2.2.1 Utilizing Adequate Initial Guess

As depicted in Table 5.2, various numbers of local maps are examined to evaluate their computational time compared to full batch optimization (in Table 4.7). In the case of the sequence 06 dataset, using 2 local maps requires a total processing time of 370.11 seconds. However, by increasing the number of local maps to 8, the processing time is significantly reduced to only 206.81 seconds. Notably, for the sequence 07 dataset, there is a substantial reduction in processing time from 513.86 seconds when using 2 local maps to a total of 257.02 seconds when employing 8 local maps, nearly halving the processing time. This approach yields results that closely resemble those of batch optimization, as demonstrated



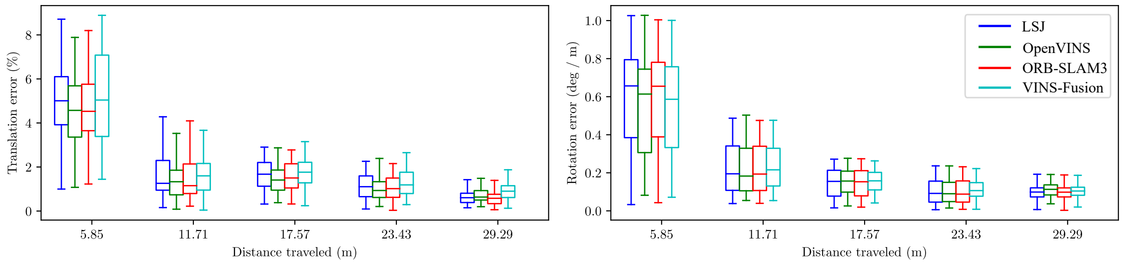
(c) EuRoC-MH03

FIGURE 5.4: Trajectory Comparison: Linear Submap Joining vs State-of-the-Art Methods OpenVINS [11], ORB-SLAM3 [12], and VINS-Fusion [13] Utilizing the “EuRoC” Datasets.

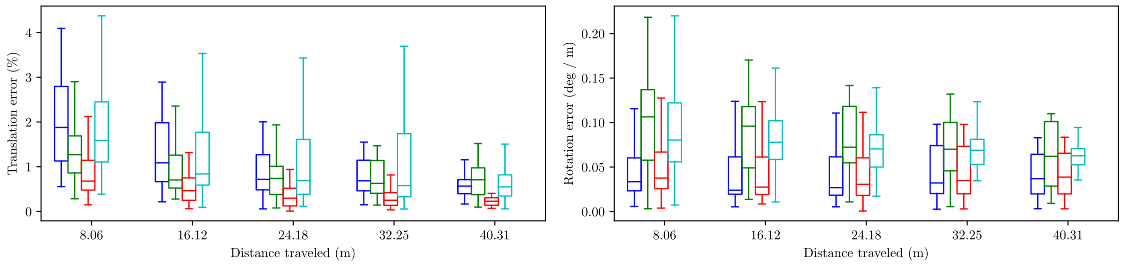
in Figure 5.6 and Figure 5.7. The RMSE for translation is 3.865m for sequence 06, 2.537m for sequence 07, and 5.630m for sequence 09, which closely aligns with the performance of batch optimization.

5.2.2.2 Absence of Favorable Initial Guess

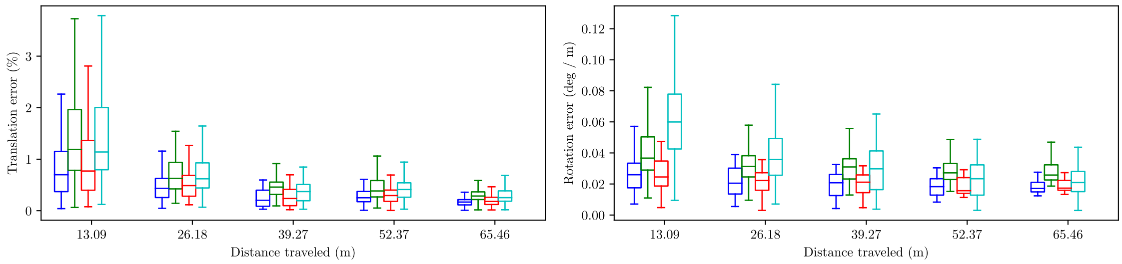
In the previous chapter (Chapter 4), batch optimization using PVI-SLAM did not consistently converge when initialized with poor poses from visual odometry (VO) without



(a) EuRoC-V101



(b) EuRoC-MH01



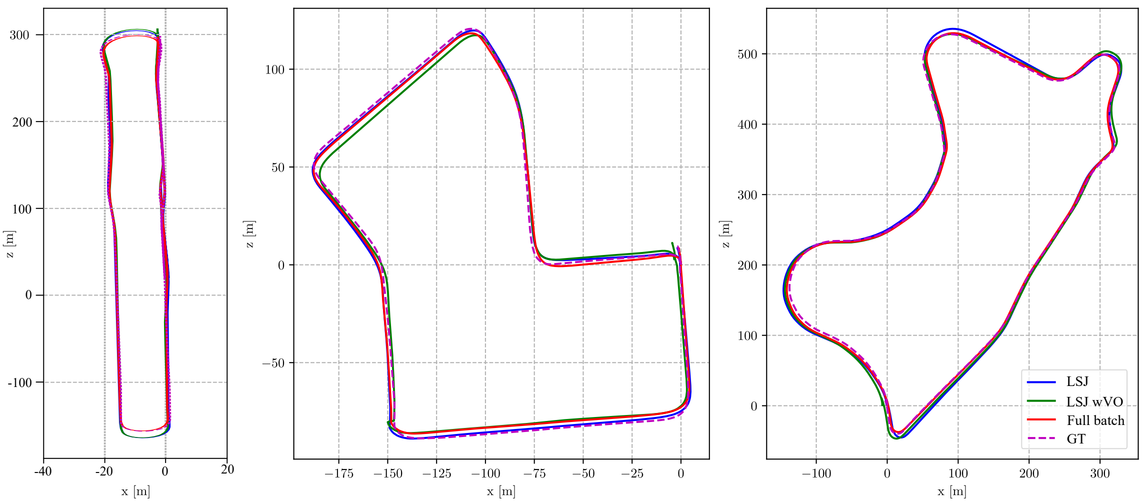
(c) EuRoC-MH03

FIGURE 5.5: Comparative Analysis of Translation and Rotation Errors: Linear Submap Joining vs State-of-the-Art Methods OpenVINS [11], ORB-SLAM3 [12], and VINS-Fusion [13] across Different "EuRoC" Datasets.

loop-closure. However, Linear Submap Joining consistently achieved convergence in all scenarios, effectively overcoming challenges in high-dimensional nonlinear optimization. As depicted in Figure 5.6 and Figure 5.7, in most instances, although Linear Submap Joining with VO demonstrates a higher RMSE compared to batch optimization, it closely approximates the results of batch optimization. For sequence 06, 07, and 09 of the "KITTI" dataset, the translation RMSE of Linear Submap Joining with VO is 6.232m, 3.991m, and 5.659m, while the rotation RMSE is 0.861°, 1.921°, and 1.528°.

TABLE 5.2: Total computation time for Linear Submap Joining for "KITTI" dataset.

Dataset	**06**			**07**			**09**		
Num of Local maps	**2**	**4**	**8**	**2**	**4**	**8**	**2**	**4**	**8**
Local Maps	359.609	263.919	165.807	469.718	306.302	191.099	401.284	358.262	188.058
Structure Transformation	3.974	3.315	3.650	4.518	4.429	4.452	5.829	5.987	6.529
Linear Submap Joining	6.530	9.390	37.354	39.624	67.872	61.469	38.455	80.228	123.270
Total Time (sec)	370.113	276.624	**206.811**	513.861	378.604	**257.020**	445.568	444.477	**317.856**



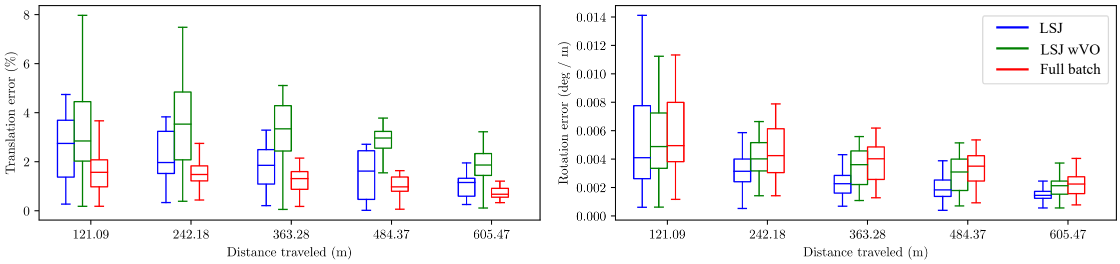
(a) KITTI-06 (b) KITTI-07 (c) KITTI-09

FIGURE 5.6: The comparison of trajectories between Batch PVI-SLAM, Linear Submap Joining, and Linear Submap Joining using VO without loop-closure using the "KITTI" datasets.

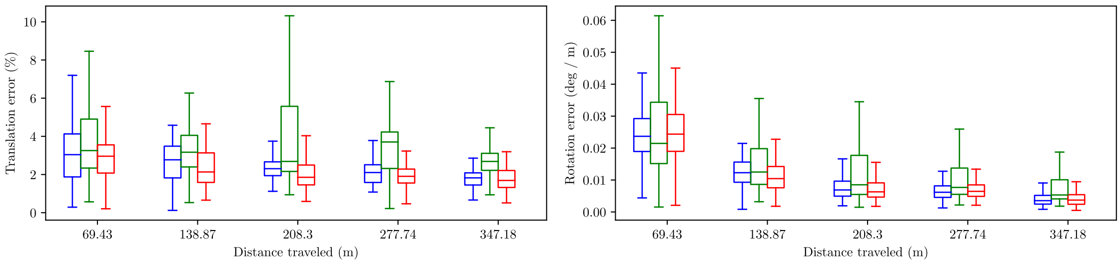
5.3 Summary

Addressing high-dimensional nonlinear optimization problems poses challenges to system robustness, as demonstrated in the previous chapter, particularly when confronted with sub-optimal initialization. In response to this challenge, the integration of the Linear SLAM framework into the PVI-SLAM system enables effective handling of high-dimensional nonlinear optimization challenges. This strategic incorporation alleviates concerns such as susceptibility to local minima traps.

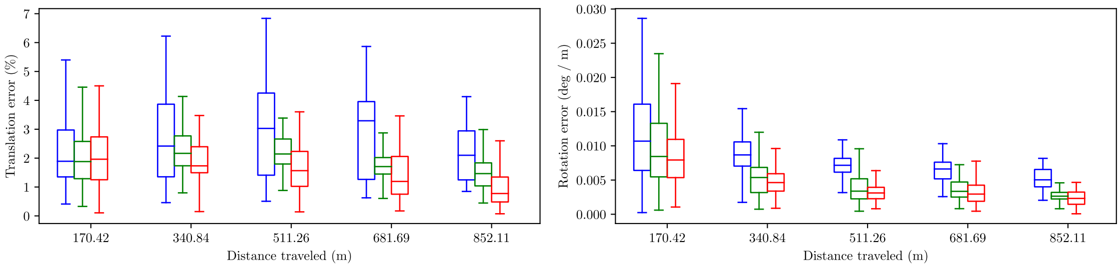
The incorporation of Linear SLAM into PVI-SLAM necessitates additional processing time for state transformation. However, this overhead proves worthwhile, as it results in substantial time savings when compared to the execution of a full nonlinear optimization procedure. The notable efficiency of Linear SLAM is attributed to its streamlined two-step approach: solving LLS problem and implementing specific coordinate changes. A



(a) KITTI-06



(b) KITTI-07



(c) KITTI-09

FIGURE 5.7: Comparison of translation error and rotation error between Batch PVI-SLAM, Linear Submap Joining, and Linear Submap Joining using VO without loop-closure for each "KITTI" dataset.

distinguishing feature of this method is its independence from initial guesses and its ability to avoid local minima issues, making it particularly advantageous when a robust solution is crucial, even in challenging initial conditions.

Consequently, this integrated system exhibits enhanced robustness in the face of challenging scenarios, showcasing improved convergence and efficiency. This innovative approach addresses the limitations associated with high-dimensional nonlinear optimization, thereby fortifying the reliability and performance of the PVI-SLAM system.

Chapter 6

Conclusion

This thesis has contributed a comprehensive framework for enhancing the robustness of VI-SLAM by incorporating advancements in both filtering-based and optimization-based approaches. The primary emphasis throughout the research has been on achieving the right balance between computational complexity and accuracy within the system. This pursuit has been driven by a recognition of the inherent challenges and limitations associated with existing methods, necessitating a refined and innovative approach to address these issues.

Addressing computational challenges in filtering-based approaches, particularly loop-closure issues, the thesis explores and implements a compressed framework on MSCKF. This innovative approach aims to achieve efficient computational complexity without compromising accuracy. Furthermore, to overcome challenges in filtering-based methods and enhance accuracy, VI-SLAM integrates PBA to handle problematic features observed in collinear motion. Lastly, to tackle the high-dimensional nonlinear optimization problem and manage computational complexity, a Linear SLAM framework is employed for joining the local map constructed by PVI-SLAM.

6.1 Summary of Contributions

6.1.1 Balancing Efficiency and Accuracy in Monocular VI-SLAM: Introducing Compressed-MSCKF with Loop-Closure

Operating VI-SLAM in a small-scale system necessitates careful management of computational complexity to ensure the robustness of the system. However, when examining filtering-based approaches, the computational cost tends to increase as the observed map grows larger. While sliding windows that marginalize past information can alleviate some computational burden, they often come at the cost of sacrificing accuracy. Consequently, incorporating loop-closure becomes essential, yet this introduces the challenge of handling the size of the state vector as keyframes are continuously included as loop-closure constraints.

To address this challenge, the Comp-MSCKF with loop-closure is introduced in Chapter 3. The MSCKF framework retains specific past camera poses as keyframes instead of all observed features in the state vector, utilizing multiple visual feature measurements to provide localization information. This approach enables linear computational complexity with respect to the number of features, a significant reduction compared to feature-based SLAM. When incorporating loop-closure constraints into the state vector, a compressed framework is applied by dividing the state into local and global components. This strategy limits the computational complexity to the quadratic order of the number of local maps, which is typically much smaller than considering the entire map.

In the experimental evaluations, the Comp-MSCKF demonstrated superior accuracy compared to both the standard MSCKF and Schmidt-MSCKF. Significantly, the Comp-MSCKF emerged as a compelling choice, even considering computational complexity. However, within the MSCKF framework and compression method, there is a potential for information loss during simultaneous marginalization and compression. The crucial task of determining the appropriate strategy for dividing the state into local and global components remains. Furthermore, for larger and longer trajectories without frequent loop-closure, significant drift can impede loop-closure in filtering-based methods, especially when problematic features are present.

6.1.2 Advancing VI-SLAM: PVI-SLAM with PBA and UGPM

The shift in focus has moved from a filtering-based method to an optimization-based approach to tackle issues stemming from significant drift, especially when loop-closure is infrequent, thereby providing higher accuracy. Additionally, advancements in computer technology have empowered the real-time implementation of optimization-based methods.

In Chapter 4, PVI-SLAM is introduced, incorporating PBA to effectively address problematic features arising from collinear motion. While conventional features, such as XYZ parametrization and IDP, perform well in certain scenarios, they face challenges in low parallax angles. The degeneracy in these situations arises from the high uncertainty in direct depth calculation, resulting in singularity and leading to divergence or local minima in the optimization problem.

Furthermore, the approach incorporates pre-integrated IMU measurements, contributing to improved accuracy and the recovery of correct metric scale in monocular VI-SLAM. Unlike the PM method that involves numerical integration, UGPM addresses IMU data in continuous-time using GP. This approach enhances accuracy, particularly in dynamic motion scenarios.

To enhance the robustness of the PVI-SLAM system, an observation ray is utilized in the objective function, departing from the conventional BA error function that relies on 2D image pixels. The observation ray objective function introduces geometric constraints based on the directions of observation rays. This incorporation of geometric constraints significantly enhances the overall robustness and accuracy of the system. Such improvements are particularly valuable in addressing challenges related to feature initialization and mitigating potential depth ambiguities.

The experimental results demonstrate that the integration of IMU significantly enhances the reliability and consistency of the system, even with fewer feature observations. In contrast, SBA+IMU encounters challenges in achieving convergence and determining appropriate stopping criteria, requiring the use of the LM optimization method. Conversely, PVI-SLAM achieves convergence using the GN method. While utilizing UGPM can reduce IMU initial error, it may not exhibit significant improvement in collinear motion. The incorporation of the observation ray objective function imparts notable robustness to the system, enabling convergence even with poorly initialized state vectors.

However, it is important to note that the convergence of high-dimensional nonlinear optimization problems is not guaranteed to reach the global minimum, and the proposed system may not assure convergence. Moreover, batch optimizing the problem incurs a high computational cost.

6.1.3 Efficient Nonlinear Optimization in VI-SLAM with Linear Submap Joining

In Chapter 5, a Linear Submap Joining method using the Linear SLAM framework was introduced. This technique is applied to the proposed PVI-SLAM methodology to address challenges associated with high-dimensional nonlinear optimization and computational complexity. Unlike the conventional submap joining approach, Linear SLAM eliminates the need for initial guesses or iterative processes for optimization by treating it as a LLS problem and employing nonlinear coordinate transformation. To integrate Linear SLAM with PVI-SLAM, an additional step is required where the state vector optimized from PVI-SLAM is transformed to be suitable for Linear SLAM. Despite the additional time required, this proves to be worthwhile, consuming a very small fraction of the total time. Moreover, it results in substantial time savings compared to the execution of a full nonlinear optimization procedure. Most importantly, the integrated system exhibits heightened robustness in challenging scenarios of bad initialization, demonstrating improved convergence and overall enhanced efficiency.

6.2 Future Research

The methodology presented in this thesis has showcased notable improvements in both accuracy and efficiency when compared to existing state-of-the-art approaches throughout the conducted experiments. Although these findings hold promise, there are compelling prospects for additional research and development endeavours to bolster and amplify the influence of the proposed methodology.

6.2.1 Observability Analysis

While the evaluation has provided valuable insights into the robustness of the system's performance, a more comprehensive exploration is essential, especially when considering the multifaceted nature of SLAM. The current analysis has been constrained to a specific depth, primarily examining convergence and accuracy. However, a notable gap exists in terms of a detailed comparative study, particularly with observability, a critical factor in SLAM systems. Observability, encompassing the system's ability to effectively estimate the robot's pose and map features, is especially pertinent in SLAM scenarios where environmental dynamics and sensor characteristics play pivotal roles. To address this gap and further advance the understanding of the proposed system, future work will focus on an expanded and more nuanced evaluation. By extending the scope of the evaluation, a comprehensive understanding is aimed at regarding how the proposed system compares to alternative methods within the dynamic landscape of VI-SLAM.

(a) QUT-SERF (b) Victoria Park (c) QUT-DVP

FIGURE 6.1: A comprehensive exploration of self-collecting datasets for advancements in the proposed method.

6.2.2 Assessing Performance on Individually Collected Datasets

In the present thesis, the evaluation of the proposed method has provided valuable insights into its performance through the analysis of selected real-world datasets. However, recognizing the need for a more comprehensive understanding of the method's capabilities and robustness, further testing with a diverse set of datasets is considered essential.

Furthermore, as articulated in the initial project plan, the hardware configuration, detailed in Appendix C, has enable the collection of visual-inertial data in both outdoor and indoor settings at Queensland University of Technology (QUT) (Samford Ecological Research Facility (SERF) and Da Vinci Precinct (DVP) Hangar facilities), as depicted in Figure 6.1. Leveraging this dataset, there is a strategic opportunity to expand the evaluation by subjecting the proposed method to a variety of motion scenarios. This extension to diverse testing environments is intended to facilitate a thorough examination of the system's adaptability and effectiveness across a spectrum of conditions.

6.2.3 Extension Work on Multi-Drone Systems

The robustness exhibited by PVI-SLAM in various scenarios underscores its efficacy in real-world applications, particularly in the domain of VI-SLAM. The integration of the Linear SLAM framework has further demonstrated the system's ability to manage computational complexity effectively and address high-dimensional nonlinear problems inherent in VI-SLAM.

In light of these achievements, future work will focus on extending the application of PVI-SLAM to multi-drone systems. The adaptability showcased in handling diverse scenarios positions the system as a promising solution for collaborative mapping and localization, especially in environments involving multiple drones. The results obtained through Linear Submap Joining suggest the potential to implement this approach in a multi-drone context, aiming to achieve results close to full optimization.

6.2.4 Real-time Implementation

The current work presented in this thesis is confined to MATLAB code, limiting its applicability to real-time systems. As part of future work, the goal is to transition the implementation of the PVI-SLAM algorithm into a real-time system. Unlike the XYZ parametrization and IDP, the parallax parametrization dynamically changes with improvements in parallax angles during feature observations, introducing challenges in optimizing feature parameters.

To address this issue, an approach similar to the one proposed by Mendes et al. [106] is planned to be adopted. Their work introduces a parametrization strategy within an incremental graph-based SLAM framework, providing a viable solution to handle changing parallax angles. Implementing a proper method inspired by Mendes et al.'s work will be crucial for ensuring the robust optimization of feature parameters in the context of PVI-SLAM system.

Appendix A

Jacobian for Reprojection Error

This appendix shows the derivation of the Jacobian matrix of the reprojection error (Equation (4.22)) with respect to the state vector (Equation (4.18)), considering the retraction mapping specified in Equation (4.37). The purpose is to enable optimization of the cost function given in Equation (4.21) within the manifold domain as explained in Section 2.2.3.

To compute the Jacobian, $\mathbf{U}_i$ is first defined as:

$$\mathbf{U}_i = \begin{bmatrix} U_{i_1} \\ U_{i_2} \\ U_{i_3} \end{bmatrix} = \mathbf{K}\ (\mathbf{R}_{C_i}^{W})^{\top}\ \mathbf{x}_j^i, \tag{A.1}$$

where the estimated reprojected observation matrix can be achieved as Equation (4.23):

$$\mathbf{u}_j^i = \begin{bmatrix} u_j^i \\ v_j^i \end{bmatrix} = \begin{bmatrix} U_{i_1}/U_{i_3} \\ U_{i_2}/U_{i_3} \end{bmatrix}. \tag{A.2}$$

Then, the Jacobian of $\mathbf{u}_j^i$ with respect to $\mathbf{U}_i$ can be calculated as:

$$\frac{\partial \mathbf{u}_j^i}{\partial \mathbf{U}_i} = \begin{bmatrix} 1/U_{i_3} & 0 & -U_{i_1}/(U_{i_3})^2 \\ 0 & 1/U_{i_3} & -U_{i_2}/(U_{i_3})^2 \end{bmatrix}. \tag{A.3}$$

In the context of PBA, the observation is subject to variation based on the anchor, determining the vector from the anchor to feature $\boldsymbol{f}_j$ as detailed in Equation (4.25). For the optimization on the manifold, the cost function is lifted using the approach outlined in Equation (4.37).

A.1 Jacobian for Observation from Main Anchor, u_j^m

In the case of the reprojected observation from the main anchor to feature j written as:

$$\mathbf{u}_j^m = \begin{bmatrix} u_j^m \\ v_j^m \end{bmatrix} = \pi(\mathbf{K}\ (\mathbf{R}_{C_m}^W)^\top\ \mathbf{x}_j^m). \tag{A.4}$$

where it is only related to the rotation of the main anchor, $\mathbf{R}_{I_m}^W$, and the feature parameter, $\mathbf{f}_j$, in the state vector. The chain rule is used to calculate the Jacobian.

A.1.1 Calculation of $\partial u_j^m / \partial \delta \phi_m$

Using the chain rule, $\frac{\partial \mathbf{u}_j^m}{\partial \delta \boldsymbol{\phi}_m}$ can be written as:

$$\frac{\partial \mathbf{u}_j^m}{\partial \delta \boldsymbol{\phi}_m} = \frac{\partial \mathbf{u}_j^m}{\partial \mathbf{U}_m} \frac{\partial \mathbf{U}_m}{\partial \delta \boldsymbol{\phi}_m}. \tag{A.5}$$

With the lifted rotation matrix, U_m can be re-written as:

$$\begin{aligned} \mathbf{U}_m \left(R_{I_m}^W \operatorname{Exp}(\delta \boldsymbol{\phi}_m) \right) &= \mathbf{K}\ (\mathbf{R}_C^I)^\top\ (\mathbf{R}_{I_m}^W\ \operatorname{Exp}(\boldsymbol{\phi_m}))^\top\ \mathbf{x}_j^m \\ &= \mathbf{K}\ (\mathbf{R}_C^I)^\top\ (1 - \delta \boldsymbol{\phi}_m^\wedge)\ (\mathbf{R}_{I_m}^W)^\top\ \mathbf{x}_j^m \\ &= \mathbf{K}\ (\mathbf{R}_C^I)^\top\ ((\mathbf{R}_{I_m}^W)^\top\ \mathbf{x}_j^m)^\wedge\ \delta \boldsymbol{\phi}_m. \end{aligned} \tag{A.6}$$

Then, the Jacobian can be calculated as:

$$\frac{\partial \mathbf{U}_m}{\partial \delta \boldsymbol{\phi}_m} = \mathbf{K}\ (\mathbf{R}_C^I)^\top\ ((\mathbf{R}_{I_m}^W)^\top\ \mathbf{x}_j^m)^\wedge \tag{A.7}$$

A.1.2 Calculation of $\partial u_j^m / \partial \delta f_j$

$$\frac{\partial \mathbf{u}_j^m}{\partial \boldsymbol{f}_j} = \frac{\partial \mathbf{u}_j^m}{\partial \mathbf{U}_m} \frac{\partial \mathbf{U}_m}{\partial \mathbf{x}_j^m} \frac{\partial \mathbf{x}_j^m}{\partial \boldsymbol{f}_j}. \tag{A.8}$$

where

$$\frac{\partial \mathbf{U}_m}{\partial \mathbf{x}_j^m} = \mathbf{K}\ (\mathbf{R}_{C_m}^W)^\top, \qquad \frac{\partial \mathbf{x}_j^m}{\partial \boldsymbol{f}_j} = \begin{bmatrix} \cos\psi_j \cos\theta_j & -\sin\psi_j \sin\theta_j & 0 \\ 0 & \cos\theta_j & 0 \\ -\sin\psi_j \cos\theta_j & -\cos\psi_j \sin\theta_j & 0 \end{bmatrix}. \tag{A.9}$$

A.2 Jacobian for Observation from Associate Anchor, u_j^a

The reprojected observation from the associate anchor to feature j written as:

$$\mathbf{u}_j^a = \begin{bmatrix} u_j^a \\ v_j^a \end{bmatrix} = \pi(\mathbf{K}\ (\mathbf{R}_{C_a}^W)^\top\ \mathbf{x}_j^a). \tag{A.10}$$

In this case, it is related to $\mathbf{R}_{I_a}^W$, ${}^W\mathbf{t}_{I_m}$, ${}^W\mathbf{t}_{I_a}$, and $\mathbf{f}_j$ in the state vector.

A.2.1 Calculation of $\partial u_j^a / \partial \delta \phi_a$

$$\frac{\partial \mathbf{u}_j^a}{\partial \delta \boldsymbol{\phi}_a} = \frac{\partial \mathbf{u}_j^a}{\partial \mathbf{U}_a}\ \frac{\partial \mathbf{U}_a}{\partial \delta \boldsymbol{\phi}_a}. \tag{A.11}$$

With the lifted rotation matrix, U_a can be re-written as:

$$\mathbf{U}_m \left(R_{I_a}^W \operatorname{Exp}(\delta \boldsymbol{\phi}_i)\right) = \mathbf{K}\ (\mathbf{R}_C^I)^\top\ ((\mathbf{R}_{I_a}^W)^\top\ \mathbf{x}_j^a)^\wedge\ \delta \boldsymbol{\phi}_a, \tag{A.12}$$

and the Jacobian of $\boldsymbol{U}$ respect to $\delta \boldsymbol{\phi}_a$ cna be written as:

$$\frac{\partial \mathbf{U}_a}{\partial \delta \boldsymbol{\phi}_a} = \mathbf{K}\ (\mathbf{R}_C^I)^\top\ ((\mathbf{R}_{I_a}^W)^\top\ \mathbf{x}_j^a)^\wedge. \tag{A.13}$$

A.2.2 Calculation of $\partial u_j^a / \partial \delta t_m$

$$\frac{\partial \mathbf{u}_j^a}{\partial \delta \boldsymbol{t}_m} = \frac{\partial \mathbf{u}_j^a}{\partial \mathbf{U}_a}\ \frac{\partial \mathbf{U}_a}{\partial \mathbf{x}_j^a}\ \frac{\partial \mathbf{x}_j^a}{\partial \delta \boldsymbol{t}_m}, \tag{A.14}$$

where

$$\frac{\partial \mathbf{U}_a}{\partial \mathbf{x}_j^a} = \mathbf{K}\ (\mathbf{R}_{C_a}^W)^\top, \tag{A.15}$$

and

$$\begin{aligned} \frac{\partial \mathbf{x}_j^a}{\partial \delta \boldsymbol{t}_m} =& \mathbf{x}_j^m \left(\frac{\partial \sin(\omega_j + \varphi_j)}{\partial \delta \boldsymbol{t}_m}\ \left\| {}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m} \right\| + \frac{\partial \left(\left\| {}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m} \right\| \right)}{\partial \delta \boldsymbol{t}_m}\ \sin(\omega_j + \varphi_j) \right) \\ & - \sin \omega_j\ \frac{\partial \left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m} \right)}{\partial \delta \boldsymbol{t}_m}. \end{aligned} \tag{A.16}$$

Here,

$$\frac{\partial \sin(\omega_j + \varphi_j)}{\partial \delta \boldsymbol{t}_m} = \frac{\partial \sin(\omega_j + \varphi_j)}{\partial \varphi_j}\ \frac{\partial \varphi_j}{\partial \cos \varphi_j}\ \frac{\partial \cos \varphi_j}{\partial \delta \boldsymbol{t}_m}, \tag{A.17}$$

$$\frac{\partial \sin\left(\omega_j+\varphi_j\right)}{\partial \varphi_j}=\cos\left(\omega_j+\varphi_j\right), \tag{A.18}$$

$$\frac{\partial \varphi_j}{\partial \cos\varphi_j}=-\frac{1}{\sqrt{1-\left(\frac{\mathbf{x}_j^{m}\ \ \left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)}{\left\|{}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right\|}\right)^2}}, \tag{A.19}$$

$$\begin{aligned}\frac{\partial \cos\varphi_j}{\partial \delta \boldsymbol{t}_m}=&\frac{\partial\left(\mathbf{x}_j^{m}\ \ \left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)\right)}{\partial \delta \boldsymbol{t}_m}\ \frac{1}{\left\|{}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right\|}\\&-\frac{\partial\left(\left\|{}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right\|\right)}{\partial \delta \boldsymbol{t}_m}\left(\mathbf{x}_j^{m}\ \ \left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)\right)\ \frac{1}{\left\|{}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right\|^2},\end{aligned} \tag{A.20}$$

$$\frac{\partial\left(\mathbf{x}_j^{m}\ \ \left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)\right)}{\partial \delta \boldsymbol{t}_m}=\frac{\partial\left(\mathbf{x}_j^{m}\ \ \left({}^{W}\mathbf{t}_{C_a}-\left({}^{W}\mathbf{t}_{I_m}+\mathbf{R}_{I_m}^{W}\delta\mathbf{t}_m\right)\right)\right)}{\partial \delta \boldsymbol{t}_m}=-\mathbf{x}_j^{m}\ \ \mathbf{R}_{I_m}^{W}, \tag{A.21}$$

$$\frac{\partial\left(\left\|{}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right\|\right)}{\partial \delta \boldsymbol{t}_m}=\frac{\partial\left(\left\|{}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right\|\right)}{\partial\left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)}\ \frac{\partial\left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)}{\partial \delta \boldsymbol{t}_m}, \tag{A.22}$$

where

$$\frac{\partial\left(\left\|{}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right\|\right)}{\partial\left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)}=-\frac{\left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)}{\left\|\left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)\right\|}, \tag{A.23}$$

$$\frac{\partial\left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)}{\partial \delta \boldsymbol{t}_m}=\frac{\partial\left({}^{W}\mathbf{t}_{C_a}-\left({}^{W}\mathbf{t}_{I_m}+\mathbf{R}_{I_m}^{W}\delta\mathbf{t}_m\right)\right)}{\partial \delta \boldsymbol{t}_m}=-\mathbf{R}_{I_m}^{W}. \tag{A.24}$$

A.2.3 Calculation of $\partial u_j^a/\partial \delta t_a$

$$\frac{\partial \mathbf{u}_j^a}{\partial \delta \boldsymbol{t}_a}=\frac{\partial \mathbf{u}_j^a}{\partial \mathbf{U}_a}\ \frac{\partial \mathbf{U}_a}{\partial \mathbf{x}_j^a}\ \frac{\partial \mathbf{x}_j^a}{\partial \delta \boldsymbol{t}_a}, \tag{A.25}$$

where

$$\begin{aligned}\frac{\partial \mathbf{x}_j^a}{\partial \delta \boldsymbol{t}_a}=&\mathbf{x}_j^{m}\left(\frac{\partial \sin\left(\omega_j+\varphi_j\right)}{\partial \delta \boldsymbol{t}_a}\ \ \left\|{}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right\|+\frac{\partial\left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)}{\partial \delta \boldsymbol{t}_a}\ \ \sin\left(\omega_j+\varphi_j\right)\right)\\&-\sin\omega_j\ \ \frac{\partial\left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)}{\partial \delta \boldsymbol{t}_a},\end{aligned} \tag{A.26}$$

$$\frac{\partial \sin\left(\omega_j+\varphi_j\right)}{\partial \delta \boldsymbol{t}_a}=\frac{\partial \sin\left(\omega_j+\varphi_j\right)}{\partial \varphi_j}\ \frac{\partial \varphi_j}{\partial \cos\varphi_j}\ \frac{\partial \cos\varphi_j}{\partial \delta \boldsymbol{t}_a}, \tag{A.27}$$

$$\begin{aligned}\frac{\partial \cos\varphi_j}{\partial \delta \boldsymbol{t}_a}=&\frac{\partial\left(\mathbf{x}_j^{m}\ \ \left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)\right)}{\partial \delta \boldsymbol{t}_a}\ \frac{1}{\left\|{}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right\|}\\&-\frac{\partial\left(\left\|{}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right\|\right)}{\partial \delta \boldsymbol{t}_a}\left(\mathbf{x}_j^{m}\ \ \left({}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right)\right)\ \frac{1}{\left\|{}^{W}\mathbf{t}_{C_a}-{}^{W}\mathbf{t}_{C_m}\right\|^2},\end{aligned} \tag{A.28}$$

$$\begin{aligned}\frac{\partial\left(\mathbf{x}_j^m \quad \left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)\right)}{\partial\delta\boldsymbol{t}_a} &= \frac{\partial\left(\mathbf{x}_j^m \quad \left({}^W\mathbf{t}_{C_a} - \left(\left({}^W\mathbf{t}_{I_m} + \mathbf{R}_{I_m}^W\delta\mathbf{t}_a\right) + \mathbf{R}_I^W\right)\right)\right)}{\partial\delta\boldsymbol{t}_m} \\ &= \mathbf{x}_j^m \quad \mathbf{R}_{I_a}^W,\end{aligned} \tag{A.29}$$

$$\frac{\partial\left(\left\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right\|\right)}{\partial\delta\boldsymbol{t}_a} = \frac{\partial\left(\left\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right\|\right)}{\partial\left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)} \quad \frac{\partial\left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)}{\partial\delta\boldsymbol{t}_a}, \tag{A.30}$$

$$\frac{\partial\left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)}{\partial\delta\boldsymbol{t}_a} = \frac{\partial\left({}^W\mathbf{t}_{C_a} - \left({}^W\mathbf{t}_{I_m} + \mathbf{R}_{I_m}^W\delta\mathbf{t}_m\right)\right)}{\partial\delta\boldsymbol{t}_a} = \mathbf{R}_{I_a}^W. \tag{A.31}$$

A.2.4 Calculation of $\partial u_j^a / f_j$

$$\frac{\partial\mathbf{u}_j^a}{\partial\boldsymbol{f}_j} = \frac{\partial\mathbf{u}_j^a}{\partial\mathbf{U}_a} \quad \frac{\partial\mathbf{U}_a}{\partial\mathbf{x}_j^a} \quad \frac{\partial\mathbf{x}_j^a}{\partial\boldsymbol{f}_j}. \tag{A.32}$$

In this case, the Jacobian with respect to azimuth and elevation angles is obtained, denoted as $f_{j12} = \begin{bmatrix} \psi_j & \theta_j \end{bmatrix}^\top$, and then compute it with respect to the parallax angle, ω_j. Firstly, $\frac{\partial\mathbf{x}_j^a}{\partial\boldsymbol{f}_{j12}}$ can be written as:

$$\frac{\partial\mathbf{x}_j^a}{\partial\boldsymbol{f}_{j12}} = \left(\left\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right\|\right)\left(\mathbf{x}_j^m \quad \frac{\partial\sin\left(\omega_j + \varphi_j\right)}{\partial\boldsymbol{f}_{j12}} + \sin\left(\omega_j + \varphi_j\right) \quad \frac{\partial\mathbf{x}_j^m}{\partial\boldsymbol{f}_{j12}}\right), \tag{A.33}$$

where

$$\frac{\partial\sin\left(\omega_j + \varphi_j\right)}{\partial\boldsymbol{f}_{j12}} = \frac{\partial\sin\left(\omega_j + \varphi_j\right)}{\partial\varphi_j} \quad \frac{\partial\varphi_j}{\partial\cos\varphi_j} \quad \frac{\partial\cos\varphi_j}{\partial\mathbf{x}_j^m} \quad \frac{\partial\mathbf{x}_j^m}{\partial\boldsymbol{f}_{j12}}, \tag{A.34}$$

$$\frac{\partial\cos\varphi_j}{\partial\mathbf{x}_j^m} = \frac{{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}}{\left\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right\|}, \tag{A.35}$$

$$\frac{\partial\mathbf{x}_j^m}{\partial\boldsymbol{f}_{j12}} = \frac{\partial\mathbf{x}_j^m}{\partial\boldsymbol{f}_j} = \begin{bmatrix} \cos\psi_j\cos\theta_j & -\sin\psi_j\sin\theta_j \\ 0 & \cos\theta_j \\ -\sin\psi_j\cos\theta_j & -\cos\psi_j\sin\theta_j \end{bmatrix}. \tag{A.36}$$

Then, the Jacobian respect to ω_j is computed as:

$$\frac{\partial\mathbf{x}_j^a}{\partial\omega_j} = \frac{\partial\sin\left(\omega_j + \varphi_j\right)}{\partial\omega_j} \quad \left\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right\| \quad \mathbf{x}_j^m - \cos\left(\omega_j\right) \quad \left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right). \tag{A.37}$$

A.3 Jacobian for Observation from Camera Position (Excluding Main and Associate Anchors), u_j^i

When calculating the reprojected observation from the camera position that is neither the main anchor nor the associate anchor:

$$\mathbf{u}_j^i = \begin{bmatrix} u_j^i \\ v_j^i \end{bmatrix} = \pi(\mathbf{K}\ (\mathbf{R}_{C_i}^W)^\top\ \mathbf{x}_j^i), \tag{A.38}$$

the Jacobian calculation can be performed as outlined in this section. Here, it is related to $\mathbf{R}_{I_i}^W$, ${}^W\mathbf{t}_{I_m}$, ${}^W\mathbf{t}_{I_a}$, ${}^W\mathbf{t}_{I_i}$ and $\mathbf{f}_j$ in the state vector.

A.3.1 Calculation of $\partial u_j^i / \partial \delta \phi_i$

$$\frac{\partial \mathbf{u}_j^i}{\partial \delta \boldsymbol{\phi}_i} = \frac{\partial \mathbf{u}_j^i}{\partial \mathbf{U}_i}\ \frac{\partial \mathbf{U}_i}{\partial \delta \boldsymbol{\phi}_i}, \tag{A.39}$$

where

$$\frac{\partial \mathbf{U}_i}{\partial \delta \boldsymbol{\phi}_i} = \mathbf{K}\ (\mathbf{R}_C^I)^\top\ ((\mathbf{R}_{I_i}^W)^\top\ \mathbf{x}_j^i)^\wedge. \tag{A.40}$$

A.3.2 Calculation of $\partial u_j^i / \partial \delta t_m$

$$\frac{\partial \mathbf{u}_j^i}{\partial \delta \boldsymbol{t}_m} = \frac{\partial \mathbf{u}_j^i}{\partial \mathbf{U}_i}\ \frac{\partial \mathbf{U}_i}{\partial \mathbf{x}_j^i}\ \frac{\partial \mathbf{x}_j^i}{\partial \delta \boldsymbol{t}_m}. \tag{A.41}$$

where

$$\frac{\partial \mathbf{U}_i}{\partial \mathbf{x}_j^i} = \mathbf{K}\ (\mathbf{R}_{C_i}^W)^\top, \tag{A.42}$$

$$\begin{aligned}\frac{\partial \mathbf{x}_j^i}{\partial \delta \boldsymbol{t}_m} =& \mathbf{x}_j^m \left(\frac{\partial \sin(\omega_j + \varphi_j)}{\partial \delta \boldsymbol{t}_m}\ \left\| {}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m} \right\| + \frac{\partial \left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m} \right)}{\partial \delta \boldsymbol{t}_m}\ \sin(\omega_j + \varphi_j) \right) \\ & - \sin\omega_j\ \frac{\partial \left({}^W\mathbf{t}_{C_i} - {}^W\mathbf{t}_{C_m} \right)}{\partial \delta \boldsymbol{t}_m},\end{aligned} \tag{A.43}$$

$$\frac{\partial \sin(\omega_j + \varphi_j)}{\partial \delta \boldsymbol{t}_m} = \frac{\partial \sin(\omega_j + \varphi_j)}{\partial \varphi_j}\ \frac{\partial \varphi_j}{\partial \cos\varphi_j}\ \frac{\partial \cos\varphi_j}{\partial \delta \boldsymbol{t}_m}, \tag{A.44}$$

$$\frac{\partial \sin(\omega_j + \varphi_j)}{\partial \varphi_j} = \cos(\omega_j + \varphi_j), \tag{A.45}$$

$$\frac{\partial \varphi_j}{\partial \cos\varphi_j} = -\frac{1}{\sqrt{1 - \left(\frac{\mathbf{x}_j^m\ ({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m})}{\| {}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m} \|} \right)^2}}, \tag{A.46}$$

$$\begin{aligned}\frac{\partial \cos\varphi_j}{\partial \delta \boldsymbol{t}_m} =& \frac{\partial \left(\mathbf{x}_j^m \quad \left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)\right)}{\partial \delta \boldsymbol{t}_m} \quad \frac{1}{\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\|} \\ &- \frac{\partial \left(\left\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right\|\right)}{\partial \delta \boldsymbol{t}_m} \left(\mathbf{x}_j^m \quad \left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)\right) \quad \frac{1}{\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\|^2},\end{aligned} \tag{A.47}$$

$$\begin{aligned}\frac{\partial \left(\mathbf{x}_j^m \quad \left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)\right)}{\partial \delta \boldsymbol{t}_m} =& \frac{\partial \left(\mathbf{x}_j^m \quad \left({}^W\mathbf{t}_{C_a} - \left(\left({}^W\mathbf{t}_{I_m} + \mathbf{R}_{I_m}^W \delta \mathbf{t}_m\right) + \mathbf{R}_I^W\right)\right)\right)}{\partial \delta \boldsymbol{t}_m} \\ &= -\mathbf{x}_j^m \quad \mathbf{R}_{I_m}^W,\end{aligned} \tag{A.48}$$

$$\frac{\partial \left(\left\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right\|\right)}{\partial \delta \boldsymbol{t}_m} = \frac{\partial \left(\left\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right\|\right)}{\partial \left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)} \quad \frac{\partial \left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)}{\partial \delta \boldsymbol{t}_m}, \tag{A.49}$$

$$\frac{\partial \left(\left\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right\|\right)}{\partial \left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)} = -\frac{\left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)}{\|\left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)\|}, \tag{A.50}$$

$$\frac{\partial \left({}^W\mathbf{t}_{C_i} - {}^W\mathbf{t}_{C_m}\right)}{\partial \delta \boldsymbol{t}_m} = \frac{\partial \left({}^W\mathbf{t}_{C_i} - \left({}^W\mathbf{t}_{I_m} + \mathbf{R}_{I_m}^W \delta \mathbf{t}_m\right)\right)}{\partial \delta \boldsymbol{t}_m} = -\mathbf{R}_{I_m}^W. \tag{A.51}$$

A.3.3 Calculation of $\partial u_j^i / \partial \delta t_a$

$$\frac{\partial \mathbf{u}_j^i}{\partial \delta \boldsymbol{t}_a} = \frac{\partial \mathbf{u}_j^i}{\partial \mathbf{U}_i} \quad \frac{\partial \mathbf{U}_i}{\partial \mathbf{x}_j^i} \quad \frac{\partial \mathbf{x}_j^i}{\partial \delta \boldsymbol{t}_a}, \tag{A.52}$$

$$\frac{\partial \mathbf{x}_j^i}{\partial \delta \boldsymbol{t}_a} = \mathbf{x}_j^m \left(\frac{\partial \sin\left(\omega_j + \varphi_j\right)}{\partial \delta \boldsymbol{t}_a} \quad \left\|{}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right\| + \frac{\partial \left({}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m}\right)}{\partial \delta \boldsymbol{t}_a} \quad \sin\left(\omega_j + \varphi_j\right)\right). \tag{A.53}$$

A.3.4 Calculation of $\partial u_j^i / \partial \delta t_i$

$$\frac{\partial \mathbf{u}_j^i}{\partial \delta \boldsymbol{t}_i} = \frac{\partial \mathbf{u}_j^i}{\partial \mathbf{U}_i} \quad \frac{\partial \mathbf{U}_i}{\partial \mathbf{x}_j^i} \quad \frac{\partial \mathbf{x}_j^i}{\partial \delta \boldsymbol{t}_i}, \tag{A.54}$$

$$\frac{\partial \mathbf{x}_j^i}{\partial \delta \boldsymbol{t}_i} = -\sin\left(\omega_j\right) \quad \frac{\partial \left({}^W\mathbf{t}_{C_i} - {}^W\mathbf{t}_{C_m}\right)}{\partial \delta \boldsymbol{t}_i}, \tag{A.55}$$

$$\frac{\partial \left({}^W\mathbf{t}_{C_i} - {}^W\mathbf{t}_{C_m}\right)}{\partial \delta \boldsymbol{t}_i} = \frac{\partial \left(\left({}^W\mathbf{t}_{I_i} + \mathbf{R}_{I_i}^W \delta \mathbf{t}_i\right) - {}^W\mathbf{t}_{C_m}\right)}{\partial \delta \boldsymbol{t}_i} = \mathbf{R}_{I_i}^W \tag{A.56}$$

A.3.5 Calculation of $\partial u_j^i / \partial f_j$

$$\frac{\partial \mathbf{u}_j^i}{\partial \boldsymbol{f}_j} = \frac{\partial \mathbf{u}_j^i}{\partial \mathbf{U}_i} \quad \frac{\partial \mathbf{U}_i}{\partial \mathbf{x}_j^i} \quad \frac{\partial \mathbf{x}_j^i}{\partial \boldsymbol{f}_j}. \tag{A.57}$$

In this case, similar to the Jacobian calculation in Section A.2.4, the computation of the Jacobian is divided into two parameters, $f_{j_{12}}$ and ω_j as:

$$\frac{\partial \mathbf{x}_j^i}{\partial \boldsymbol{f}_{j_{12}}} = \frac{\partial \mathbf{x}_j^i}{\partial \boldsymbol{f}_j} = \begin{bmatrix} \cos\psi_j \cos\theta_j & -\sin\psi_j \sin\theta_j \\ 0 & \cos\theta_j \\ -\sin\psi_j \cos\theta_j & -\cos\psi_j \sin\theta_j \end{bmatrix}, \tag{A.58}$$

$$\frac{\partial \mathbf{x}_j^i}{\partial \omega_j} = \frac{\partial \sin(\omega_j + \varphi_j)}{\partial \omega_j} \;\left\| {}^W\mathbf{t}_{C_a} - {}^W\mathbf{t}_{C_m} \right\| \; \mathbf{x}_j^m - \cos(\omega_j) \; \left({}^W\mathbf{t}_{C_i} - {}^W\mathbf{t}_{C_m} \right). \tag{A.59}$$

Appendix B

Jacobian for IMU

This appendix provides the derivation of the Jacobian matrix of the pre-integrated IMU (Equation (4.36)) with respect to the state vector (Equation (4.18)), considering the retraction mapping specified in Equation (4.37). The objective is to facilitate the optimization of the cost function given in Equation (4.21) within the manifold domain, as elucidated in Section 2.2.3.

B.1 Jacobian for Rotation Residual, $e_{\Delta R_{ij}}$

The residual of rotation can be derived as follows:

$$\mathbf{e}_{\Delta R_{ij}} \doteq \mathrm{Log}\left(\left(\Delta\mathbf{R}_j^i\left(\overline{\mathbf{b}}_{\omega_i}\right)\mathrm{Exp}\left(\frac{\partial\Delta\mathbf{R}_j^i}{\partial\mathbf{b}_{\omega_i}}\delta\mathbf{b}_{\omega_i}\right)\right)^\top \mathbf{R}_{I_i}^{W\top}\mathbf{R}_{I_j}^{W}\right), \tag{B.1}$$

where its Jacobian respect to the lifted state vector is composed as:

$$\mathbf{J}_R = \frac{\partial\mathbf{e}_{\Delta\boldsymbol{R}_{ij}}}{\partial\delta\mathbf{x}} = \begin{bmatrix} 0, & \cdots, & \frac{\partial\mathbf{e}_{\Delta\boldsymbol{R}_{ij}}}{\partial\delta\boldsymbol{\phi}_i}, & 0, & 0, & \frac{\partial\mathbf{e}_{\Delta\boldsymbol{R}_{ij}}}{\partial\delta\tilde{\boldsymbol{b}}_{\omega_i}}, & 0, & \frac{\partial\mathbf{e}_{\Delta\boldsymbol{R}_{ij}}}{\partial\delta\boldsymbol{\phi}_j}, & 0, & 0, & 0, & 0, & \cdots, & 0\end{bmatrix}. \tag{B.2}$$

B.1.1 Calculation of $\partial e_{\Delta R_{ij}} / \partial \delta \phi_i$

$$
\begin{aligned}
\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\boldsymbol{R}_{I_i}^{W} \operatorname{Exp}\left(\delta \boldsymbol{\phi}_i\right)\right) &= \operatorname{Log}\left(\left(\Delta \boldsymbol{R}_j^i\left(\overline{\boldsymbol{b}}_{\omega_i}\right) \boldsymbol{E}\right)^{\top}\left(\boldsymbol{R}_{I_i}^{W} \operatorname{Exp}\left(\delta \boldsymbol{\phi}_i\right)\right)^{\top} \boldsymbol{R}_{I_j}^{W}\right) \\
&= \operatorname{Log}\left(\left(\Delta \boldsymbol{R}_j^i\left(\overline{\boldsymbol{b}}_{\omega_i}\right) \boldsymbol{E}\right)^{\top} \operatorname{Exp}\left(-\delta \boldsymbol{\phi}_i\right) \boldsymbol{R}_{I_i}^{W\top} \boldsymbol{R}_{I_j}^{W}\right) \\
&= \operatorname{Log}\left(\left(\Delta \boldsymbol{R}_j^i\left(\overline{\boldsymbol{b}}_{\omega_i}\right) \boldsymbol{E}\right)^{\top} \boldsymbol{R}_{I_i}^{W\top} \boldsymbol{R}_{I_j}^{W} \operatorname{Exp}\left(-\boldsymbol{R}_{I_j}^{W\top} \boldsymbol{R}_{I_i}^{W} \delta \boldsymbol{\phi}_i\right)\right) \\
&\simeq \boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\boldsymbol{R}_{I_i}^{W}\right) - J_r^{-1}\left(\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\boldsymbol{R}_{I_i}^{W}\right)\right) \boldsymbol{R}_{I_j}^{W\top} \boldsymbol{R}_{I_i}^{W} \delta \boldsymbol{\phi}_i,
\end{aligned}
\tag{B.3}
$$

where

$$
\mathbf{E} = \operatorname{Exp}\left(\frac{\partial \Delta \mathbf{R}_j^i}{\partial \mathbf{b}_{\omega_i}} \delta \mathbf{b}_{\omega_i}\right). \tag{B.4}
$$

Therefore, Jacobian of rotational residual, $\partial \boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}} / \partial \delta \boldsymbol{\phi}_i$, can be achieved as follow:

$$
\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}}{\partial \delta \boldsymbol{\phi}_i} = -J_r^{-1}\left(\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\boldsymbol{R}_{I_i}^{W}\right)\right) \boldsymbol{R}_{I_j}^{W\top} \boldsymbol{R}_{I_i}^{W}. \tag{B.5}
$$

B.1.2 Calculation of $\partial e_{\Delta R_{ij}} / \partial \delta b_{\omega_i}$

$$
\begin{aligned}
&\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\delta \boldsymbol{b}_{\omega_i} + \tilde{\delta} \boldsymbol{b}_{\omega_i}\right) \\
&= \operatorname{Log}\left(\left(\Delta \boldsymbol{R}_j^i\left(\overline{\boldsymbol{b}}_{\omega_i}\right) \operatorname{Exp}\left(\frac{\partial \Delta \boldsymbol{R}_j^i}{\partial \boldsymbol{b}_{\omega_i}}\left(\delta \boldsymbol{b}_{\omega_i} + \tilde{\delta} \boldsymbol{b}_{\omega_i}\right)\right)\right)^{\top} \boldsymbol{R}_{I_i}^{W\top} \boldsymbol{R}_{I_j}^{W}\right) \\
&\simeq \operatorname{Log}\left(\left(\Delta \boldsymbol{R}_j^i\left(\overline{\boldsymbol{b}}_{\omega_i}\right) \boldsymbol{E} \operatorname{Exp}\left(\boldsymbol{J}_r^b \frac{\partial \Delta \boldsymbol{R}_j^i}{\partial \boldsymbol{b}_{\omega_i}} \tilde{\delta} \boldsymbol{b}_{\omega_i}\right)\right)^{\top} \boldsymbol{R}_{I_i}^{W\top} \boldsymbol{R}_{I_j}^{W}\right) \\
&= \operatorname{Log}\left(\operatorname{Exp}\left(-\boldsymbol{J}_r^b \frac{\partial \Delta \boldsymbol{R}_j^i}{\partial \boldsymbol{b}_{\omega_i}} \tilde{\delta} \boldsymbol{b}_{\omega_i}\right)\left(\Delta \boldsymbol{R}_j^i\left(\overline{\boldsymbol{b}}_{\omega_i}\right) \boldsymbol{E}\right)^{\top} \boldsymbol{R}_{I_i}^{W\top} \boldsymbol{R}_{I_j}^{W}\right) \\
&= \operatorname{Log}\left(\operatorname{Exp}\left(-\boldsymbol{J}_r^b \frac{\partial \Delta \boldsymbol{R}_j^i}{\partial \boldsymbol{b}_{\omega_i}} \tilde{\delta} \boldsymbol{b}_{\omega_i}\right) \operatorname{Exp}\left(\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\delta \boldsymbol{b}_{\omega_i}\right)\right)\right) \\
&= \operatorname{Log}\left(\operatorname{Exp}\left(\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\delta \boldsymbol{b}_{\omega_i}\right)\right) \operatorname{Exp}\left(-\operatorname{Exp}\left(\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\delta \boldsymbol{b}_{\omega_i}\right)\right)^{\top} \boldsymbol{J}_r^b \frac{\partial \Delta \boldsymbol{R}_j^i}{\partial \boldsymbol{b}_{\omega_i}} \tilde{\delta} \boldsymbol{b}_{\omega_i}\right)\right) \\
&\simeq \boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\delta \boldsymbol{b}_{\omega_i}\right) - \boldsymbol{J}_r^{-1}\left(\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\delta \boldsymbol{b}_{\omega_i}\right)\right) \operatorname{Exp}\left(\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\delta \boldsymbol{b}_{\omega_i}\right)\right)^{\top} \boldsymbol{J}_r^b \frac{\partial \Delta \boldsymbol{R}_j^i}{\partial \boldsymbol{b}_{\omega_i}} \tilde{\delta} \boldsymbol{b}_{\omega_i},
\end{aligned}
\tag{B.6}
$$

where $E = \operatorname{Exp}\left(\frac{\partial \Delta \boldsymbol{R}_j^i}{\partial \boldsymbol{b}_{\omega_i}} \delta \boldsymbol{b}_{\omega_i}\right)$ and $\boldsymbol{J}_r^b = J_r\left(\frac{\partial \Delta \boldsymbol{R}_{ij}}{\partial \boldsymbol{b}_{\omega_i}} \delta \boldsymbol{b}_{\omega_i}\right)$. Therefore, the Jacobian of rotation residual respect to bias can be written as:

$$
\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}}{\partial \tilde{\delta} \boldsymbol{b}_{\omega_i}} = -J_r^{-1}\left(\mathbf{e}_{\Delta \boldsymbol{R}_{ij}}\left(\delta \boldsymbol{b}_{\omega_i}\right)\right) \operatorname{Exp}\left(\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\delta \boldsymbol{b}_{\omega_i}\right)\right)^{\top} \mathbf{J}_r^b \frac{\partial \Delta \boldsymbol{R}_j^i}{\partial \boldsymbol{b}_{\omega_i}}. \tag{B.7}
$$

B.1.3 Calculation of $\partial e_{\Delta R_{ij}}/\partial \delta \phi_j$

$$\begin{aligned}\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\boldsymbol{R}_{I_j}^W \operatorname{Exp}(\delta\boldsymbol{\phi}_j)\right) &= \log\left(\left(\Delta\boldsymbol{R}_j^i\left(\overline{\boldsymbol{b}}_{\omega_i}\right)\boldsymbol{E}\right)^\top \boldsymbol{R}_{I_i}^{W\top}\left(\boldsymbol{R}_{I_j}^W \operatorname{Exp}(\delta\boldsymbol{\phi}_j)\right)\right) \\ &\simeq \boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\boldsymbol{R}_{I_j}^W\right) + J_r^{-1}\left(\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\boldsymbol{R}_{I_j}^W\right)\right)\delta\boldsymbol{\phi}_j,\end{aligned} \tag{B.8}$$

where Jacobian of rotational residual, $\partial \boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}/\partial \boldsymbol{\phi}_j$, can be achieved as follow:

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}}{\partial \delta\boldsymbol{\phi}_j} = J_r^{-1}\left(\boldsymbol{e}_{\Delta \boldsymbol{R}_{ij}}\left(\boldsymbol{R}_{I_j}^W\right)\right). \tag{B.9}$$

B.2 Jacobian for Transition Residual, $e_{\Delta t_{ij}}$

The residual of translation can be derived as follows:

$$\boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}} \doteq \boldsymbol{R}_{I_i}^{W\top}\left({}^W\boldsymbol{t}_{I_j} - {}^W\boldsymbol{t}_{I_i} - {}^W\boldsymbol{v}_{I_i}\Delta t - \tfrac{1}{2}\boldsymbol{g}\Delta t^2\right) - \left[\Delta\boldsymbol{t}_j^i\left(\overline{\boldsymbol{b}}_{\omega_i}, \overline{\boldsymbol{b}}_{a_i}\right) + \frac{\partial \Delta\boldsymbol{t}_j^i}{\partial \boldsymbol{b}_{\omega_i}}\delta\boldsymbol{b}_{\omega_i} + \frac{\partial \Delta\boldsymbol{t}_j^i}{\partial \boldsymbol{b}_{a_i}}\delta\boldsymbol{b}_{a_i}\right], \tag{B.10}$$

where its Jacobian respect to the lifted state vector is composed as:

$$\boldsymbol{J}_t = \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial \delta\boldsymbol{x}} = \left[\begin{array}{cccccccccccccc} 0, & \cdots, & \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial \delta\boldsymbol{\phi}_i}, & \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial \delta\boldsymbol{t}_i}, & \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial \delta\boldsymbol{v}_i}, & \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial \delta\tilde{\boldsymbol{b}}_{\omega_i}}, & \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial \delta\tilde{\boldsymbol{b}}_{a_i}}, & 0, & \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial \delta\boldsymbol{t}_j}, & 0, & 0, & 0, & \cdots, & 0 \end{array}\right]. \tag{B.11}$$

B.2.1 Calculation of $\partial e_{\Delta t_{ij}}/\partial \delta \phi_i$

$$\begin{aligned}&\boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}\left(\boldsymbol{R}_{I_i}^W \operatorname{Exp}(\delta\boldsymbol{\phi}_i)\right) \\ &\quad= \left(\boldsymbol{R}_{I_i}^W \operatorname{Exp}(\delta\boldsymbol{\phi}_i)\right)^\top\left({}^W\boldsymbol{t}_{I_j} - {}^W\boldsymbol{t}_{I_i} - {}^W\boldsymbol{v}_{I_i}\Delta t - \frac{1}{2}\boldsymbol{g}\Delta t^2\right) - \boldsymbol{C} \\ &\quad\simeq \left(\boldsymbol{I} - \delta\boldsymbol{\phi}_i^\wedge\right)\boldsymbol{R}_{I_i}^{W\top}\left({}^W\boldsymbol{t}_{I_j} - {}^W\boldsymbol{t}_{I_i} - {}^W\boldsymbol{v}_{I_i}\Delta t - \frac{1}{2}\boldsymbol{g}\Delta t^2\right) - \boldsymbol{C} \\ &\quad= \boldsymbol{e}_{\Delta t_{ij}}\left(\boldsymbol{R}_{I_i}^W\right) + \left(\boldsymbol{R}_{I_i}^{W\top}\left({}^W\boldsymbol{t}_{I_j} - {}^W\boldsymbol{t}_{I_i} - {}^W\boldsymbol{v}_{I_i}\Delta t - \frac{1}{2}\boldsymbol{g}\Delta t^2\right)\right)^\wedge \delta\boldsymbol{\phi}_i,\end{aligned} \tag{B.12}$$

where $\boldsymbol{C} = \Delta\boldsymbol{t}_j^i\left(\overline{\boldsymbol{b}}_{\omega_i}, \overline{\boldsymbol{b}}_{a_i}\right) + \frac{\partial \Delta\boldsymbol{t}_j^i}{\partial \boldsymbol{b}_{\omega_i}}\delta\boldsymbol{b}_{\omega_i} + \frac{\partial \Delta\boldsymbol{t}_j^i}{\partial \boldsymbol{b}_{a_i}}\delta\boldsymbol{b}_{a_i}$, then:

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial \delta\phi_i} = \left(\boldsymbol{R}_{I_i}^{W\top}\left({}^W\boldsymbol{t}_{I_j} - {}^W\boldsymbol{t}_{I_i} - {}^W\boldsymbol{v}_{I_i}\Delta t - \frac{1}{2}\boldsymbol{g}\Delta t\right)\right)^\wedge. \tag{B.13}$$

B.2.2 Calculation of $\partial e_{\Delta t_{ij}}/\partial\delta t_i$

$$\begin{aligned}\boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}\left({}^W\boldsymbol{t}_{I_i}+\boldsymbol{R}_{I_i}^W\delta\boldsymbol{t}_i\right) &= \boldsymbol{R}_{I_i}^{W\top}\left({}^W\boldsymbol{t}_{I_j}-{}^W t_{I_i}-{}^W\boldsymbol{v}_{I_i}\Delta t-\frac{1}{2}\boldsymbol{g}\Delta t^2\right)-\boldsymbol{C}\\ &= \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}\left({}^W\boldsymbol{t}_{I_i}\right)-\delta t_i,\end{aligned} \tag{B.14}$$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial\delta\boldsymbol{t}_i}=-I. \tag{B.15}$$

B.2.3 Calculation of $\partial e_{\Delta t_{ij}}/\partial\delta v_i$

$$\begin{aligned}\boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}\left({}^W\boldsymbol{v}_{I_i}+\delta\boldsymbol{v}_i\right) &= \boldsymbol{R}_{I_i}^{W\top}\left({}^W\boldsymbol{t}_{I_j}-{}^W \boldsymbol{t}_{I_i}-{}^W\boldsymbol{t}_{I_i}\Delta t-\delta\boldsymbol{v}_i\Delta t-\frac{1}{2}\boldsymbol{g}\Delta t^2\right)-\boldsymbol{C}\\ &= \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}\left({}^W\boldsymbol{v}_{I_i}\right)+\left(-\boldsymbol{R}_{I_i}^{W\top}\Delta t\right)\delta\boldsymbol{v}_i,\end{aligned} \tag{B.16}$$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial\delta\boldsymbol{v}_i}=-\boldsymbol{R}_{I_i}^{W\top}\Delta t. \tag{B.17}$$

B.2.4 Calculation of $\partial e_{\Delta t_{ij}}/\partial\tilde{\delta} b_{\omega_i}$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial\tilde{\delta}\boldsymbol{b}_{\omega_i}}=-\frac{\partial\Delta\boldsymbol{t}_j^i}{\partial\boldsymbol{b}_{\omega_i}}. \tag{B.18}$$

B.2.5 Calculation of $\partial e_{\Delta t_{ij}}/\partial\tilde{\delta} b_{a_i}$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial\tilde{\delta}\boldsymbol{b}_{a_i}}=-\frac{\partial\Delta\boldsymbol{t}_j^i}{\partial\boldsymbol{b}_{a_i}}. \tag{B.19}$$

B.2.6 Calculation of $\partial e_{\Delta t_{ij}}/\partial\delta t_j$

$$\begin{aligned}\boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}\left({}^W\boldsymbol{t}_{I_j}+\boldsymbol{R}_{I_j}^W\delta\boldsymbol{t}_j\right) &= \boldsymbol{R}_{I_i}^{W\top}\left({}^W\boldsymbol{e}_{I_j}-{}^W \boldsymbol{t}_{I_i}-{}^W\boldsymbol{v}_{I_i}\Delta t-\frac{1}{2}\boldsymbol{g}\Delta t^2\right)-\boldsymbol{C}\\ &= \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}\left({}^W\boldsymbol{t}_{I_j}\right)+\left(\boldsymbol{R}_{I_i}^{W\top}\boldsymbol{R}_{I_j}^W\right)\delta\boldsymbol{t}_j,\end{aligned} \tag{B.20}$$

$$\frac{\partial \boldsymbol{e}_{\Delta t_{ij}}}{\partial\delta\boldsymbol{t}_j}=\boldsymbol{R}_{I_i}^{W\top}\boldsymbol{R}_{I_j}^W. \tag{B.21}$$

B.3 Jacobian for Velocity Residual, $e_{\Delta v_{ij}}$

The residual velocity can be derived as follows:

$$\boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}} \doteq \boldsymbol{R}_{I_i}^{W\top} \left({}^{W}\boldsymbol{v}_{I_j} - {}^{W}v_{I_i} - \boldsymbol{g}\Delta t \right) - \left[\Delta \boldsymbol{v}_j^i \left(\overline{\boldsymbol{b}}_{\omega_i}, \overline{\boldsymbol{b}}_{a_i} \right) + \frac{\partial \Delta \boldsymbol{v}_j^i}{\partial \boldsymbol{b}_{\omega_i}} \delta \boldsymbol{b}_{\omega_i} + \frac{\partial \Delta \boldsymbol{v}_j^i}{\partial \boldsymbol{b}_{a_i}} \delta \boldsymbol{b}_{a_i} \right] \tag{B.22}$$

where its Jacobian respect to the lifted state vector is composed as:

$$\boldsymbol{J}_v = \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{t}_{ij}}}{\partial \delta \boldsymbol{x}} = \left[0, \quad \cdots, \quad \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}}{\partial \delta \boldsymbol{\phi}_i}, \quad 0, \quad \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}}{\partial \delta \boldsymbol{v}_i}, \quad \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}}{\partial \tilde{\delta} \boldsymbol{b}_{\omega_i}}, \quad \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}}{\partial \tilde{\delta} \boldsymbol{b}_{a_i}}, \quad 0, \quad 0, \quad \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}}{\partial \delta \boldsymbol{v}_j}, \quad 0, \quad 0, \quad \cdots, \quad 0 \right] \tag{B.23}$$

B.3.1 Calculation of $\partial e_{\Delta v_{ij}} / \partial \delta \phi_i$

$$\begin{aligned} \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}} \left(\boldsymbol{R}_{I_i}^{W} \operatorname{Exp}\left(\delta \boldsymbol{\phi}_i\right) \right) &= \left(\boldsymbol{R}_{I_i}^{W} \operatorname{Exp}\left(\delta \boldsymbol{\phi}_i\right) \right)^{\top} \left({}^{W}\boldsymbol{v}_{I_j} - {}^{W}\boldsymbol{v}_{I_i} - \boldsymbol{g}\Delta t \right) - \boldsymbol{D} \\ &= \left(\boldsymbol{I} - \delta \boldsymbol{\phi}_i^{\wedge} \right) \boldsymbol{R}_{I_i}^{W\top} \left({}^{W}\boldsymbol{v}_{I_j} - {}^{W}\boldsymbol{v}_{I_i} - \boldsymbol{g}\Delta t \right) - \boldsymbol{D} \\ &= \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}} \left(\boldsymbol{R}_{I_i}^{W} \right) + \left(\boldsymbol{R}_{I_i}^{W\top} \left({}^{W}\boldsymbol{v}_{I_j} - {}^{W}\boldsymbol{v}_{I_i} - \boldsymbol{g}\Delta t \right) \right)^{\wedge} \delta \boldsymbol{\phi}_i, \end{aligned} \tag{B.24}$$

where $D = \Delta \boldsymbol{v}_j^i \left(\overline{\boldsymbol{b}}_{\omega_i}, \overline{\boldsymbol{b}}_{a_i} \right) + \frac{\partial \Delta \boldsymbol{v}_j^i}{\partial \boldsymbol{b}_{\omega_i}} \delta \boldsymbol{b}_{\omega_i} + \frac{\partial \Delta \boldsymbol{v}_j^i}{\partial \boldsymbol{b}_{a_i}} \delta \boldsymbol{b}_{a_i}$, then:

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}}{\partial \delta \boldsymbol{\phi}_i} = \left(\boldsymbol{R}_{I_i}^{W\top} \left({}^{W}\boldsymbol{v}_{I_j} - {}^{W}\boldsymbol{v}_{I_i} - W\Delta t \right) \right)^{\wedge}. \tag{B.25}$$

B.3.2 Calculation of $\partial e_{\Delta v_{ij}} / \partial \delta v_i$

$$\begin{aligned} \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}} \left({}^{W}\boldsymbol{v}_{I_i} + \delta \boldsymbol{v}_i \right) &= \boldsymbol{R}_{I_i}^{W\top} \left({}^{W}\boldsymbol{v}_{I_j} - {}^{W}\boldsymbol{v}_{I_i} - \delta \boldsymbol{v}_i - g\Delta t \right) - \boldsymbol{D} \\ &= \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}} \left({}^{W}\boldsymbol{v}_{I_i} \right) - \boldsymbol{R}_{I_i}^{W\top} \delta \boldsymbol{v}_i, \end{aligned} \tag{B.26}$$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}}{\partial \delta \boldsymbol{v}_i} = -\boldsymbol{R}_{I_i}^{W\top}. \tag{B.27}$$

B.3.3 Calculation of $\partial e_{\Delta v_{ij}} / \partial \tilde{\delta} b_{\omega_i}$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}}{\partial \tilde{\delta} \boldsymbol{b}_{\omega_i}} = -\frac{\partial \Delta \boldsymbol{v}_j^i}{\partial \boldsymbol{b}_{\omega_i}}. \tag{B.28}$$

B.3.4 Calculation of $\partial e_{\Delta v_{ij}}/\partial\tilde{\delta} b_{a_i}$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}}{\partial \tilde{\delta} \boldsymbol{b}_{a_i}} = -\frac{\partial \Delta \boldsymbol{v}_j^i}{\partial \boldsymbol{b}_{a_i}}. \tag{B.29}$$

B.3.5 Calculation of $\partial e_{\Delta v_{ij}}/\partial v_j$

$$\begin{aligned} \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}\left({}^W\boldsymbol{v}_{I_j} + \delta \boldsymbol{v}_j\right) &= \boldsymbol{R}_{I_i}^{W\top}\left({}^W\boldsymbol{v}_{I_j} + \delta \boldsymbol{v}_j - {}^W\boldsymbol{v}_{I_i} - \boldsymbol{g}\Delta t\right) - \boldsymbol{D} \\ &= \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}\left({}^W\boldsymbol{v}_{I_j}\right) + \boldsymbol{R}_{I_i}^{W\top}\delta \boldsymbol{v}_j, \end{aligned} \tag{B.30}$$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{v}_{ij}}}{\partial \delta \boldsymbol{v}_i} = \boldsymbol{R}_{I_i}^{W\top}. \tag{B.31}$$

B.4 Jacobian for Biases Residual, $e_{\Delta b_{\omega_{ij}}}$ and $e_{\Delta b_{a_{ij}}}$

The residual biases can be derived as follows:

$$\boldsymbol{e}_{\Delta \boldsymbol{b}_{\omega_{ij}}} = \boldsymbol{b}_{\omega_j} - \boldsymbol{b}_{\omega_i}, \tag{B.32}$$

$$\boldsymbol{e}_{\Delta \boldsymbol{b}_{a_{ij}}} = \boldsymbol{b}_{a_j} - \boldsymbol{b}_{a_i}. \tag{B.33}$$

where its Jacobian respect to the lifted state vector is composed as:

$$\boldsymbol{J}_{\boldsymbol{b}_\omega} = \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{b}_{\omega_{ij}}}}{\partial \delta \boldsymbol{x}} = \left[\begin{array}{cccccccccccccc} 0, & \cdots, & 0, & 0, & 0, & \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{b}_{\omega_{ij}}}}{\partial \delta \boldsymbol{b}_{\omega_i}}, & 0, & 0, & 0, & 0, & \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{b}_{\omega_{ij}}}}{\partial \delta \boldsymbol{b}_{\omega_j}}, & 0, & \cdots, & 0 \end{array}\right], \tag{B.34}$$

$$\boldsymbol{J}_{b_a} = \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{b}_{a_{ij}}}}{\partial \delta \boldsymbol{x}} = \left[\begin{array}{cccccccccccccc} 0, & \cdots, & 0, & 0, & 0, & 0, & \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{b}_{a_{ij}}}}{\partial \delta \boldsymbol{b}_{a_i}}, & 0, & 0, & 0, & 0, & \frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{b}_{a_{ij}}}}{\partial \delta \boldsymbol{b}_{a_j}}, & \cdots, & 0 \end{array}\right]. \tag{B.35}$$

B.4.1 Calculation of $\partial e_{\Delta b_{\omega_{ij}}}/\partial \delta b_{\omega_i}$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{b}_{\omega_{ij}}}}{\partial \delta \boldsymbol{b}_{\omega_i}} = -\boldsymbol{I}. \tag{B.36}$$

B.4.2 Calculation of $\partial e_{\Delta b_{\omega_{ij}}}/\partial \delta b_{\omega_j}$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{b}_{\omega_{ij}}}}{\partial \delta \boldsymbol{b}_{\omega_j}} = I. \tag{B.37}$$

B.4.3 Calculation of $\partial e_{\Delta b_{a_{ij}}} / \partial \delta b_{a_i}$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{b}_{a_{ij}}}}{\partial \delta \boldsymbol{b}_{a_i}} = -I. \tag{B.38}$$

B.4.4 Calculation of $\partial e_{\Delta b_{a_{ij}}} / \partial \delta b_{a_i}$

$$\frac{\partial \boldsymbol{e}_{\Delta \boldsymbol{b}_{a_{ij}}}}{\partial \delta \boldsymbol{b}_{a_j}} = I. \tag{B.39}$$

Appendix C

Specification of Hardware

The hardware specifications presented in this appendix detail the setup utilized for implementing and testing proposed methods during the collection of the real dataset. This was conducted in support of Australian Research Council Discovery Project DP200101640, as discussed in Section 6.2.2.

FIGURE C.1: Image of Holybro x500

C.1 Holybro X500

- Pixhawk 4 autopilot
- Power Management PM07

- Motors - 2216 KV880(V2 Update)
- Propeller 1045(V2 Update)
- Pixhawk4 GPS
- 433MHz Telemetry Radio / 915MHz Telemetry Radio
- Power and Radio Cables
- Dimensions: 410*410*300mm
- Wheelbase: 500mm
- Weight: 978g

C.2 Pixhawk 4

Figure C.2: Image of Pixhawk4

Main FMU Processor: STM32F765

- 32 Bit Arm® Cortex®-M7, 216MHz, 2MB memory, 512KB RAM

IO Processor: STM32F100

- 32 Bit Arm® Cortex®-M3, 24MHz, 8KB SRAM

On-board sensors:

- Accel/Gyro: ICM-20689
- Accel/Gyro: BMI055
- Magnetometer: IST8310
- Barometer: MS5611

GPS: u-blox Neo-M8N GPS/GLONASS receiver; integrated magnetometer IST8310 Interfaces:

- 8-16 PWM outputs (8 from IO, 8 from FMU)
- 3 dedicated PWM/Capture inputs on FMU
- Dedicated R/C input for CPPM
- Dedicated R/C input for Spektrum / DSM and S.Bus with analog / PWM RSSI input
- Dedicated S.Bus servo output
- 5 general purpose serial ports
- 3 I2C ports
- 4 SPI buses
- Up to 2 CANBuses for dual CAN with serial ESC
- Analog inputs for voltage / current of 2 batteries

Weight and Dimensions:

- Weight: 15.8g
- Dimensions: 44x84x12mm

FIGURE C.3: Image of NVIDIA Jeston NX

C.3 NVIDIA Jetson Xavier NX

- GPU : NVIDIA Volta architecture with 384 NVIDIA CUDA® cores and 48 Tensor cores
- CPU : 6-core NVIDIA Carmel ARM®v8.2 64-bit CPU 6 MB L2 + 4 MB L3
- DL Accelerator : 2x NVDLA Engines
- Vision Accelerator : 7-Way VLIW Vision Processor
- Memory : 8 GB 128-bit LPDDR4x @ 51.2GB/s
- Storage : microSD (not included)
- USB : 4x USB 3.1, USB 2.0 Micro-B
- Others : GPIO, I2C, I2S, SPI, UART
- Mechanical : 103 mm x 90.5 mm x 34.66 mm

C.4 Zed 2

FIGURE C.4: Image of ZED2

Video output:

- 2.2K mode: 15 fps; resolution 4416 x 1242
- 1080p mode: 30/15 fps; resolution 3840 x 1080
- 720p mode: 60/30/15 fps; resolution 2560 x 720 (stereo passthrough mode)
- WVGA mode: 100/60/30/15 fps; resolution 1344 x 376

Depth:

- Resolution: native video (in ultra mode)
- FPS: up to 100 Hz
- Depth range: 20 cm to 20 m
- Field of view: 110° horizontal, 70° vertical, 120° diagonal max.
- Technology: neural stereo depth sensing

Motion:

- motion sensors: accelerometer, gyroscope (data rate: 400 Hz)
- Pose update rate: up to 100 Hz
- Position sensors: barometer, magnetometer (data rate: 25/50 Hz)
- Technology: 6-DoF visual-inertial stereo simultaneous localisation and mapping (SLAM) with advanced sensor fusion and thermal compensation
- Pose drift: 0.35

Image sensors:

- Resolution: dual 4M pixel sensors with 2-micron pixels
- Sensor format: native 16:9 for a larger horizontal field of view
- Sensor size: 1/3" BSI (backside illumination) sensor with high low-light sensitivity
- Shutter with electronically synchronised rolling shutter
- Camera controls: adjust resolution, frame rate, brightness, contrast, saturation, gamma, sharpness, exposure, white balance

Bibliography

[1] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2 2017. ISSN 15523098. doi: 10.1109/TRO.2016.2597321.

[2] Jonghyuk Kim, Hongkyoon Byun, Jose Guivant, and Tor Arne Johansen. Compressed Pseudo-SLAM: Pseudorange Integrated Generalised Compressed SLAM. In *Australasian Conference on Robotics and Automation (ACRA)*. Australian Robotics and Automation Association, 12 2020.

[3] Patrick Geneva. Visual-Inertial Navigation Systems: An Introduction [PowerPoint slides], 2021. URL `https://udel.edu/~ghuang/icra21-vins-workshop/slides/01-vins_tutorial.pdf`.

[4] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3565–3572, 2007. ISBN 1424406021. doi: 10.1109/ROBOT.2007.364024.

[5] Lee E. Clement, Valentin Peretroukhin, Jacob Lambert, and Jonathan Kelly. The Battle for Filter Supremacy: A Comparative Study of the Multi-State Constraint Kalman Filter and the Sliding Window Filter. In *Proceedings -2015 12th Conference on Computer and Robot Vision, CRV 2015*, pages 23–30. Institute of Electrical and Electronics Engineers Inc., 7 2015. ISBN 9781479919864. doi: 10.1109/CRV.2015.11.

[6] Patrick Geneva, Kevin Eckenhoff, and Guoquan Huang. A linear-complexity EKF for visual-inertial navigation with loop closures. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2019-May, pages 3535–3541. Institute of Electrical and Electronics Engineers Inc., 5 2019. ISBN 9781538660263. doi: 10.1109/ICRA.2019.8793836.

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237, 2013. ISSN 0278-3649.

[8] Cedric Le Gentil and Teresa Vidal-Calleja. Continuous latent state preintegration for inertial-aided systems. *International Journal of Robotics Research*, 42(10):874–900, 9 2023. ISSN 17413176. doi: 10.1177/02783649231199537.

[9] Jose Luis Blanco, Francisco Angel Moreno, and Javier Gonzalez. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, 27(4):327–351, 11 2009. ISSN 09295593. doi: 10.1007/s10514-009-9138-7.

[10] Tong Qin, Peiliang Li, and Shaojie Shen. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 34(4): 1004–1020, 8 2018. ISSN 15523098. doi: 10.1109/TRO.2018.2853729.

[11] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. OpenVINS: A Research Platform for Visual-Inertial Estimation. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4666–4672. Institute of Electrical and Electronics Engineers Inc., 5 2020. ISBN 9781728173955. doi: 10.1109/ICRA40945.2020.9196524.

[12] Carlos Campos, Richard Elvira, Juan J.Gomez Rodriguez, Jose M.M. Montiel, and Juan D. Tardos. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 12 2021. ISSN 19410468. doi: 10.1109/TRO.2021.3075644.

[13] Tong Qin, Jie Pan, Shaozu Cao, and Shaojie Shen. A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors. 1 2019.

[14] Jonghyuk Kim, Jiantong Cheng, Jose Guivant, and Juan Nieto. Compressed fusion of gnss and inertial navigation with simultaneous localization and mapping. *IEEE Aerospace and Electronic Systems Magazine*, 32(8):22–36, 2017. doi: 10.1109/MAES.2017.8071552.

[15] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. DTAM: Dense tracking and mapping in real-time. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2320–2327, 2011. doi: 10.1109/ICCV.2011.6126513.

[16] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-Scale Direct monocular SLAM. In *Lecture Notes in Computer Science (including subseries*

Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 8690 LNCS, pages 834–849. Springer Verlag, 2014. ISBN 9783319106045. doi: 10.1007/978-3-319-10605-2_54.

[17] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 7 2016. ISSN 01628828. doi: 10.1109/TPAMI.2017.2658577.

[18] C. Harris and M. Stephens. A Combined Corner and Edge Detector. pages 1–23. British Machine Vision Association and Society for Pattern Recognition, 4 2013. doi: 10.5244/c.2.23.

[19] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3951 LNCS, pages 404–417. Springer, Berlin, Heidelberg, 2006. ISBN 3540338322. doi: 10.1007/11744023_32.

[20] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 11 2004. ISSN 09205691. doi: 10.1023/B:VISI.0000029664.99615.94.

[21] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3951 LNCS, pages 430–443. Springer Verlag, 2006. ISBN 3540338322. doi: 10.1007/11744023_34.

[22] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2564–2571, 2011. ISBN 9781457711015. doi: 10.1109/ICCV.2011.6126544.

[23] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 4 2014. doi: 10.1109/TPAMI.2014.2345390.

[24] Ke Sun, Kartik Mohta, Bernd Pfrommer, Michael Watterson, Sikang Liu, Yash Mulgaonkar, Camillo J. Taylor, and Vijay Kumar. Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight. *IEEE Robotics and Automation Letters*, 3 (2):965–972, 11 2017.

[25] Todd Lupton and Salah Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28(1):61–76, 2 2012. ISSN 15523098. doi: 10.1109/TRO.2011.2170332.

[26] Chang Chen, Hua Zhu, Menggang Li, and Shaoze You. A Review of Visual-Inertial Simultaneous Localization and Mapping from Filtering-Based and Optimization-Based Perspectives. *Robotics*, 7(3):45, 8 2018. ISSN 2218-6581. doi: 10.3390/robotics7030045.

[27] Patrick Beeson, Joseph Modayil, and Benjamin Kuipers. Factoring the mapping problem: Mobile robot map-building in the hybrid spatial semantic hierarchy. *International Journal of Robotics Research*, 29(4):428–459, 4 2010. ISSN 02783649. doi: 10.1177/0278364909100586.

[28] Dorian Gálvez-López and Juan D. Tardós. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. ISSN 15523098. doi: 10.1109/TRO.2012.2197158.

[29] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007. doi: 10.1109/TPAMI.2007.1049.

[30] Georg Klein and David Murray. Parallel tracking and mapping on a camera phone. In *Science and Technology Proceedings - IEEE 2009 International Symposium on Mixed and Augmented Reality, ISMAR 2009*, pages 83–86, 2009. ISBN 9781424453900. doi: 10.1109/ISMAR.2009.5336495.

[31] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. 3-d mapping with an rgb-d camera. *IEEE Transactions on Robotics*, 30(1):177–187, 2014. doi: 10.1109/TRO.2013.2279412.

[32] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H.J. Kelly, and Andrew J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013. doi: 10.1109/CVPR.2013.178.

[33] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, 2014. doi: 10.1109/ICRA.2014.6906584.

[34] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5): 1147–1163, 2 2015. doi: 10.1109/TRO.2015.2463671.

[35] Raul Mur-Artal and Juan D. Tardos. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5): 1255–1262, 10 2016. doi: 10.1109/TRO.2017.2705103.

[36] Ali Tourani, Hriday Bavle, Jose Luis Sanchez-Lopez, and Holger Voos. Visual slam: What are the current trends and what to expect? *Sensors*, 22(23), 2022. ISSN 1424-8220. doi: 10.3390/s22239297.

[37] Weifeng Chen, Guangtao Shang, Aihong Ji, Chengjun Zhou, Xiyang Wang, Chonghui Xu, Zhenxiong Li, and Kai Hu. An overview on visual slam: From tradition to semantic. *Remote Sensing*, 14(13), 2022. ISSN 2072-4292. doi: 10.3390/rs14133010.

[38] Stephan Weiss, Markus W. Achtelik, Simon Lynen, Margarita Chli, and Roland Siegwart. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In *2012 IEEE International Conference on Robotics and Automation*, pages 957–964, 2012. doi: 10.1109/ICRA.2012.6225147.

[39] Shaojie Shen, Yash Mulgaonkar, Nathan Michael, and Vijay Kumar. Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft mav. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4974–4981, 2014. doi: 10.1109/ICRA.2014.6907588.

[40] Igor Cvišić, Josip Ćesić, Ivan Marković, and Ivan Petrovic. Soft-slam: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles. *Journal of Field Robotics*, 35, 11 2017. doi: 10.1002/rob.21762.

[41] Myriam Servières, Valérie Renaudin, Alexis Dupuis, and Nicolas Antigny. Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking. *Journal of Sensors*, 2021, 2021. ISSN 16877268. doi: 10.1155/2021/2054828.

[42] Yanhao Zhang, Teng Zhang, and Shoudong Huang. Comparison of EKF based SLAM and optimization based SLAM algorithms. In *Proceedings of the 13th IEEE Conference on Industrial Electronics and Applications, ICIEA 2018*, pages 1308–1313.

Institute of Electrical and Electronics Engineers Inc., 6 2018. ISBN 9781538637579. doi: 10.1109/ICIEA.2018.8397911.

[43] Shoudong Huang and Gamini Dissanayake. Convergence and consistency analysis for extended Kalman filter based SLAM. *IEEE Transactions on Robotics*, 23(5): 1036–1049, 10 2007. ISSN 15523098. doi: 10.1109/TRO.2007.903811.

[44] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the EKF-SLAM algorithm. *IEEE International Conference on Intelligent Robots and Systems*, pages 3562–3568, 2006. doi: 10.1109/IROS.2006.281644.

[45] Guoquan P. Huang, Anastasios I. Mourikis, and Stergios I. Roumeliotis. Analysis and improvement of the consistency of extended Kalman filter based SLAM. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 473–479, 2008. ISSN 10504729. doi: 10.1109/ROBOT.2008.4543252.

[46] Teng Zhang, Kanzhi Wu, Jingwei Song, Shoudong Huang, and Gamini Dissanayake. Convergence and Consistency Analysis for A 3D Invariant-EKF SLAM. *IEEE Robotics and Automation Letters*, 2(2):733–740, 2 2017.

[47] J. A. Castellanos, R. Martinez-Cantin, J. D. Tardós, and J. Neira. Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems*, 55(1):21–29, 1 2007. ISSN 09218890. doi: 10.1016/j.robot.2006.06.005.

[48] Mingyang Li and Anastasios I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *http://dx.doi.org/10.1177/0278364913481251*, 32(6):690–711, 6 2013. ISSN 0278-3649. doi: 10.1177/0278364913481251.

[49] Joel A. Hesch, Dimitrios G. Kottas, Sean L. Bowman, and Stergios I. Roumeliotis. Camera-IMU-based localization: Observability analysis and consistency improvement. *International Journal of Robotics Research*, 33(1):182–201, 1 2014. ISSN 02783649. doi: 10.1177/0278364913509675.

[50] Guoquan Huang, Michael Kaess, and John J. Leonard. Towards consistent visual-inertial navigation. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4926–4933, 9 2014. ISSN 10504729. doi: 10.1109/ICRA.2014.6907581.

[51] Robert Mahony, Tarek Hamel, and Jean Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, 2008. ISSN 00189286. doi: 10.1109/TAC.2008.923738.

[52] Luca Carlone, Vito Macchia, Federico Tibaldi, and Basilio Bona. Quaternion-based EKF-SLAM from relative pose measurements: Observability analysis and applications. *Robotica*, 33(6):1250–1280, 7 2015. ISSN 14698668. doi: 10.1017/S0263574714000678.

[53] Silvère Bonnabel, Philippe Martin, and Erwan Salaün. Invariant extended Kalman filter: Theory and application to a velocity-aided attitude estimation problem. *Proceedings of the IEEE Conference on Decision and Control*, pages 1298–1304, 2009. ISSN 25762370. doi: 10.1109/CDC.2009.5400372.

[54] Silvere Bonnabel. Left-invariant extended Kalman filter and attitude estimation. *Proceedings of the IEEE Conference on Decision and Control*, pages 1027–1032, 2007. ISSN 25762370. doi: 10.1109/CDC.2007.4434662.

[55] Silvère Bonnabel. Symmetries in observer design: Review of some recent results and applications to EKF-based SLAM. *Lecture Notes in Control and Information Sciences*, 422:3–15, 2012. ISSN 01708643. doi: 10.1007/978-1-4471-2343-9_1/COVER.

[56] Axel Barrau and Silvere Bonnabel. An EKF-SLAM algorithm with consistency properties. 10 2015.

[57] Sebastian Thrun and Yufeng Liu. Multi-robot SLAM with sparse extended information filers. *Springer Tracts in Advanced Robotics*, 15:254–265, 2005. ISSN 1610742X. doi: 10.1007/11008941_27/COVER.

[58] Michael Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: a factored solution to the simultaneous localization and mapping problem. *AAAI/IAAI*, 2002. doi: 10.5555/777092.777184.

[59] Michael Montemerlo and Sebastian Thrun. FastSLAM 2.0. *Springer Tracts in Advanced Robotics*, 27:63–90, 2007. ISSN 1610742X. doi: 10.1007/978-3-540-46402-0_4.

[60] José E. Guivant and Eduardo Mario Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 6 2001. ISSN 1042296X. doi: 10.1109/70.938382.

[61] Jiantong Cheng, Jonghyuk Kim, Zhenyu Jiang, and Xixiang Yang. Compressed Unscented Kalman filter-based SLAM. In *2014 IEEE International Conference on Robotics and Biomimetics, IEEE ROBIO 2014*, pages 1602–1607. Institute of Electrical and Electronics Engineers Inc., 4 2014. ISBN 9781479973965. doi: 10.1109/ROBIO.2014.7090563.

[62] Jose E. Guivant. The Generalized Compressed Kalman Filter. *Robotica*, 35(8): 1639–1669, 8 2017. ISSN 14698668. doi: 10.1017/S0263574716000369.

[63] STANLEY F. SCHMIDT. Application of State-Space Methods to Navigation Problems. volume 3, pages 293–340. Elsevier, 1 1966. doi: 10.1016/b978-1-4831-6716-9.50011-4.

[64] Patrick Geneva, James Maley, and Guoquan Huang. An Efficient Schmidt-EKF for 3D Visual-Inertial SLAM. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:12097–12107, 3 2019. ISSN 10636919. doi: 10.1109/CVPR.2019.01238.

[65] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. Robust visual inertial odometry using a direct EKF-based approach. *IEEE International Conference on Intelligent Robots and Systems*, 2015-December:298–304, 12 2015. ISSN 21530866. doi: 10.1109/IROS.2015.7353389.

[66] Thomas Schneider, Marcin Dymczyk, Marius Fehr, Kevin Egger, Simon Lynen, Igor Gilitschenski, and Roland Siegwart. Maplab: An Open Framework for Research in Visual-Inertial Mapping and Localization. *IEEE Robotics and Automation Letters*, 3(3):1418–1425, 7 2018. ISSN 23773766. doi: 10.1109/LRA.2018.2800113.

[67] Jian Li, Qing Li, and Nong Cheng. A Combined Visual-Inertial Navigation System of MSCKF and EKF-SLAM. *2018 IEEE CSAA Guidance, Navigation and Control Conference, CGNCC 2018*, 8 2018. doi: 10.1109/GNCC42960.2018.9018998.

[68] Sejong Heo, Jaehyuck Cha, and Chan Gook Park. EKF-Based Visual Inertial Navigation Using Sliding Window Nonlinear Optimization. *IEEE Transactions on Intelligent Transportation Systems*, 20(7):2470–2479, 7 2019. ISSN 15249050. doi: 10.1109/TITS.2018.2866637.

[69] Mingyang Li and Anastasios I. Mourikis. Improving the accuracy of EKF-based visual-inertial odometry. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 828–835, 2012. ISSN 10504729. doi: 10.1109/ICRA.2012.6225229.

[70] Benny Dai, Cedric Le Gentil, and Teresa Vidal-Calleja. A Tightly-Coupled Event-Inertial Odometry using Exponential Decay and Linear Preintegrated Measurements. *IEEE International Conference on Intelligent Robots and Systems*, 2022-October: 9475–9482, 2022. ISSN 21530866. doi: 10.1109/IROS47612.2022.9981249.

[71] P. A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 7 2007. ISSN 16153375. doi: 10.1007/s10208-005-0179-9.

[72] Timothy D. Barfoot. *State estimation for robotics*. Cambridge University Press, 1 2017. ISBN 9781316671528. doi: 10.1017/9781316671528.

[73] K. Lynch and F. Park. Modern Robotics: Mechanics, Planning, and Control. 2017.

[74] Ananth Ranganathan, Michael Kaess, and Frank Dellaert. Fast 3D pose estimation with out-of-sequence measurements. *IEEE International Conference on Intelligent Robots and Systems*, pages 2486–2493, 2007. doi: 10.1109/IROS.2007.4399318.

[75] Gabe Sibley, L. Matthies, and G. Sukhatme. Constant Time Sliding Window Filter SLAM as a Basis for Metric Visual Perception. 2007.

[76] Esha D. Nerurkar, Kejian J. Wu, and Stergios I. Roumeliotis. C-KLAM: Constrained keyframe-based localization and mapping. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3638–3643. Institute of Electrical and Electronics Engineers Inc., 9 2014. ISBN 9781479936854. doi: 10.1109/ICRA.2014.6907385.

[77] Tue Cuong Dong-Si and Anastasios I. Mourikis. Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5655–5662, 2011. ISBN 9781612843865. doi: 10.1109/ICRA.2011.5980267.

[78] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008. ISSN 15523098. doi: 10.1109/TRO.2008.2006706.

[79] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John Leonard, and Frank Dellaert. ISAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3281–3288, 2011. ISBN 9781612843865. doi: 10.1109/ICRA.2011.5979641.

[80] Jun Wang, Jingwei Song, Liang Zhao, and Shoudong Huang. A Submap Joining Based RGB-D SLAM Algorithm Using Planes as Features. In *Springer Proceedings in Advanced Robotics*, volume 5, pages 367–382. Springer Science and Business Media B.V., 2018. doi: 10.1007/978-3-319-67361-5_24.

[81] Yongbo Chen, Shoudong Huang, Robert Fitch, and Jianqiao Yu. Efficient Active SLAM Based on Submap Joining, Graph Topology and Convex Optimization. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5159–5166. Institute of Electrical and Electronics Engineers Inc., 9 2018. ISBN 9781538630815. doi: 10.1109/ICRA.2018.8460864.

[82] Liang Zhao, Shoudong Huang, and Gamini Dissanayake. Linear SFM: A hierarchical approach to solving structure-from-motion problems by decoupling the linear and nonlinear components. *ISPRS Journal of Photogrammetry and Remote Sensing*, 141: 275–289, 7 2018. ISSN 09242716. doi: 10.1016/j.isprsjprs.2018.04.007.

[83] Arindam Saha, Bibhas Chandra Dhara, Saiyed Umer, Ahmad Ali AlZubi, Jazem Mutared Alanazi, and Kulakov Yurii. CORB2I-SLAM: An Adaptive Collaborative Visual-Inertial SLAM for Multiple Robots. *Electronics (Switzerland)*, 11(18), 9 2022. ISSN 20799292. doi: 10.3390/electronics11182814.

[84] Shoudong Huang, Zhan Wang, and Gamini Dissanayake. Sparse local submap joining filter for building large-scale maps. *IEEE Transactions on Robotics*, 24(5):1121–1130, 2008. ISSN 15523098. doi: 10.1109/TRO.2008.2003259.

[85] Liang Zhao, Shoudong Huang, and Gamini Dissanayake. Linear SLAM: Linearising the SLAM Problems using Submap Joining. *Automatica*, 100:231–246, 9 2018.

[86] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. *International Journal of Robotics Research*, 34(3):314–334, 3 2015. ISSN 17413176. doi: 10.1177/0278364914554813.

[87] Vladyslav Usenko, Nikolaus Demmel, David Schubert, Jorg Stuckler, and Daniel Cremers. Visual-Inertial Mapping with Non-Linear Factor Recovery. *IEEE Robotics and Automation Letters*, 5(2):422–429, 4 2020. ISSN 23773766. doi: 10.1109/LRA.2019.2961227.

[88] Maxime Ferrera, Alexandre Eudes, Julien Moras, Martial Sanfourche, and Guy Le Besnerais. OV2SLAM: A Fully Online and Versatile Visual SLAM for Real-Time Applications. *IEEE Robotics and Automation Letters*, 6(2):1399–1406, 4 2021. ISSN 23773766. doi: 10.1109/LRA.2021.3058069.

[89] Liang Zhao, Shoudong Huang, Yanbiao Sun, Lei Yan, and Gamini Dissanayake. ParallaxBA: Bundle adjustment using parallax angle feature parametrization. *International Journal of Robotics Research*, 34(4-5):493–516, 4 2015. ISSN 17413176. doi: 10.1177/0278364914551583.

[90] Jonghyuk Kim and Salah Sukkarieh. Real-time implementation of airborne inertial-SLAM. *Robotics and Autonomous Systems*, 55(1):62–71, 1 2007. ISSN 0921-8890. doi: 10.1016/J.ROBOT.2006.06.006.

[91] Hongkyoon Byun, Jonghyuk Kim, Fernando Vanegas, and Felipe Gonzalez. Schmidt or Compressed filtering for Visual-Inertial SLAM? In *Australasian Conference on Robotics and Automation (ACRA)*, 2021.

[92] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3D reconstruction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 363–370, 2006. ISBN 0769525970. doi: 10.1109/CVPR.2006.236.

[93] Javier Civera, Andrew J. Davison, and J. M.Martínez Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, 2008. ISSN 15523098. doi: 10.1109/TRO.2008.2003276.

[94] Manolis I.A. Lourakis and Antonis A. Argyros. SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 36(1), 3 2009. ISSN 00983500. doi: 10.1145/1486525.1486527.

[95] Kurt Konolige. Sparse sparse bundle adjustment. In *British Machine Vision Conference, BMVC 2010 - Proceedings*. British Machine Vision Association, BMVA, 2010. doi: 10.5244/C.24.102.

[96] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011. ISBN 9781612843865. doi: 10.1109/ICRA.2011.5979949.

[97] Li Yang Liu. *Towards Observable Urban Visual SLAM*. PhD thesis, University of Technology Sydney, 2020.

[98] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 6 1981. ISSN 15577317. doi: 10.1145/358669.358692.

[99] Handuo Zhang, Karunasekera Hasith, and Han Wang. A hybrid feature parametrization for improving stereo-SLAM consistency. *IEEE International Conference on Control and Automation, ICCA*, pages 1021–1026, 8 2017. ISSN 19483457. doi: 10.1109/ICCA.2017.8003201.

[100] Chang-Ryeol Lee and Kuk-Jin Yoon. Exploiting Feature Confidence for Forward Motion Estimation. 4 2017.

[101] Hongkyoon Byun, Liang Zhao, Jonghyuk Kim, and Shoudong Huang. Comparison Between MATLAB Bundle Adjustment Function and Parallax Bundle Adjustment. In *2022 17th International Conference on Control, Automation, Robotics and Vision, ICARCV 2022*, pages 60–65. Institute of Electrical and Electronics Engineers Inc., 2022. ISBN 9781665476874. doi: 10.1109/ICARCV57592.2022.10004279.

[102] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision (Cited by: 11343)*, volume 2. Cambridge University Press, 2004. ISBN 0521540518. URL `http://www.robots.ox.ac.uk/~vgg/hzbook/`.

[103] Liang Zhao, Shoudong Huang, Lei Yan, Jack Jianguo Wang, Gibson Hu, and Gamini Dissanayake. Large-scale monocular SLAM by local bundle adjustment and map joining. *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*, pages 431–436, 2010. doi: 10.1109/ICARCV.2010.5707820.

[104] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W. Achtelik, and Roland Siegwart. The EuRoC micro aerial vehicle datasets. *International Journal of Robotics Research*, 35(10):1157–1163, 9 2016. ISSN 17413176. doi: 10.1177/0278364915620033.

[105] Shoudong Huang, Zhan Wang, Gamini Dissanayake, and Udo Frese. Iterated SLSJF: A sparse local submap joining algorithm with improved consistency. 2008.

[106] Ellon Paiva Mendes, Simon Lacroix, and Joan Solà. Parallax angle parametrization in incremental SLAM. In *2016 14th International Conference on Control, Automation, Robotics and Vision, ICARCV 2016*. Institute of Electrical and Electronics Engineers Inc., 2016. ISBN 9781509035496. doi: 10.1109/ICARCV.2016.7838805.