A New Tightly-Coupled Dual-VIO for a Mobile Manipulator with Dynamic Locomotion

by

Jianxiang Xu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2024

© Jianxiang Xu 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis presents a novel approach to address the challenges encountered by a mobile manipulator engaged in dynamic locomotion within cluttered environments. The proposed technique involves the use of a dual monocular visual-inertial odometry (dual-VIO) strategy, which integrates two independent monocular VIO modules, one at the mobile base and the other at the end effector (EE). These modules are intricately coupled at the low level of the factor graph to provide a robust solution. The approach leverages arm kinematics to treat each monocular VIO as a positional anchor in relation to the other, thereby introducing a soft geometric constraint during VIO pose optimization. This mechanism effectively stabilizes both estimators, mitigating potential instability during highly dynamic locomotions. The performance of the proposed approach has been rigorously evaluated through extensive experimental testing, directly comparing it to the concurrent operation of independent dual Monocular VINS (VINS-Mono). The envisaged impact extends beyond the specific application, as the approach may lay the groundwork for multi-VIO fusion and enhanced system redundancy within the realm of Active-SLAM (A-SLAM).

Acknowledgements

I am deeply grateful for the opportunity to work under the exceptional guidance of my supervisor, Professor Soo Jeon. His invaluable mentorship, patience, and unwavering support have profoundly shaped my academic and professional development. Professor Jeon's consistent encouragement and passion have played a pivotal role in fostering my knowledge and expertise. Additionally, his exceptional editorial and academic experience have significantly enhanced my works and publications. Overall, it has been my privilege to have Professor Soo Jeon as my supervisor.

I extend my sincere thanks to all my colleagues at the Waterloo Mechanical Systems & Control Laboratory, including Jeong-woo Han, Shiyi Yang, Arnab Joardar, Sengiyumva Kisole, Tim Van Meijel, and Simon Zou. Their support and camaraderie have made my time here enriching and unforgettable. In particular, I would like to express my gratitude to Dr. Jeong-woo and Dr. Shiyi for their early mentorship with hardware and research. I also acknowledge Sengiyumva for contributing ideas for the theory of geometric impedance control in Chapter 3 of this thesis, which was a joint collaboration for the Model Predictive Control (MPC) research course project. Lastly, I want to express my appreciation to Arnab for consistently providing valuable ideas and feedback throughout my master's program.

I am grateful to Brandon DeHart and Robert Wagner at RobotHub for generously providing technical support and experiences at the facility, especially in the Vicon Motion Capture System (Vicon MoCap). I also express my gratitude to the reading group hosted by Dr. Yuhao Chen at the VIP (Vision and Image Processing) Lab. Their insights have greatly enriched my understanding of NeRF and SLAM.

I want to thank the Ontario Graduate Scholarship (OGS), the University of Waterloo, and Professor Soo Jeon for funding my program. I also extend my thanks to my mother for her financial support throughout my degree. I am extremely grateful for the academic resources obtained through the University of Waterloo for both my Undergrad and Master's degrees.

Last but not least, I want to express my profound appreciation to my beloved partner, Dinara, my sister, Annie, and above all, my mother, for their unwavering support and presence throughout my life.

Dedication

For my beloved partner Dinara,
my little sister Annie,
and mom Huiling.

To the past and future me

Table of Contents

Author's Declaration **ii**

Abstract **iii**

Acknowledgements **iv**

Dedication **v**

List of Figures **x**

List of Tables **xiv**

List of Abbreviations **xv**

List of Symbols **xx**

1 Introduction **1**

1.1 Motivation 1

1.2 Contributions 4

1.3 Thesis Outline 4

2 Preliminary **5**

2.1 Taxonomy of Visual Inertial Odometry (VIO) 5

2.1.1 Camera Modules Commonly Used in VIO 7
2.1.2 Front-End: Indirect vs. Direct 8
2.1.3 Back-End: Filter vs. Optimization 10
2.1.4 Mapping: Implicit vs. Explicit 10
2.1.5 Loop-closure 11
2.1.6 Sensor Fusion: Tightly-Coupled vs. Loosely-Coupled 12
2.1.7 Benchmark on State-of-The-Art (SOTA) VIO 12
2.1.8 Towards A-SLAM 14
2.1.9 Factor Graph 14
2.2 Anatomy of VINS-Mono 19
2.2.1 Front-End 20
2.2.2 Initialization 22
2.2.3 Back-End 25
2.3 Camera and IMU Calibration 27
2.3.1 Characterize IMU Noises with Allan Variance [1] 27
2.3.2 Camera and IMU Calibration with Kalibr 29
2.4 Preliminary Concepts for Manipulator Kinematics 31
2.4.1 Manifold 31
2.4.2 Lie Groups and Lie Algebra 31
2.4.3 Lie groups and Lie algebra in $\mathbf{SO}(3)$ 32
2.4.4 Lie groups and Lie algebra in $\mathbf{SE}(3)$ 33
2.4.5 3-Sphere Group $\mathbf{S}^3$ and Unit Quaternion $\mathbb{Q}$ 34
2.5 Mathematical Foundation for Manipulator Kinematics 35
2.5.1 Translational Motion in Euclidean space 35
2.5.2 Rotational Motion in Different Representation 35
2.5.3 Rigid Body Motion in $\mathbf{SE}(3)$ 38
2.5.4 Error Function(s) on $\mathbf{SE}(3)$ 39

2.5.5 Twists Representation for a Revolute Joint . . . 39
2.5.6 WAM Forward Kinematics Models . . . 40
2.6 Early Feasibility Study of Dual-Visual Inertial Odometry (VIO) . . . 41
2.6.1 Different Camera Setup for Dual-VIO . . . 41
2.6.2 Stability of VINS-Mono on a Mobile Manipulator . . . 43

3 A New Tightly-Coupled Dual-VIO 44
3.1 Introduction . . . 44
3.2 Related Works . . . 47
3.2.1 VIO Fusion on Different Platforms . . . 47
3.2.2 Multi-VIO Fusion . . . 48
3.3 Formulation of Proposed Dual VIOs . . . 49
3.3.1 Formulation of Factor Graph . . . 50
3.3.2 Incorporation of Arm Odometry Factor . . . 52
3.4 Implementation . . . 54
3.4.1 System Architecture . . . 54
3.4.2 Computational Advantage . . . 55

4 Experiment 58
4.1 Experimental Setup . . . 58
4.1.1 Hardware . . . 59
4.1.2 Environment . . . 60
4.1.3 Choreography . . . 61
4.1.4 Dataset . . . 63
4.1.5 Alignment . . . 63
4.1.6 Calibration Configs . . . 64
4.1.7 Performance Metric . . . 65
4.2 Results for Dynamic Arm Motions (LR and UD) . . . 67

4.2.1 Demonstrations and Trajectories (Left-And-Right (LR)) 67
4.2.2 Demonstrations and Trajectories (Up-And-Down (UD)) 71
4.2.3 Performance Results and Discussion (Left-And-Right (LR) and Up-And-Down (UD)) 75
4.3 Additional Results for Static Arm Motions 77
4.3.1 Demonstrations and Trajectories (Home (H)) 77
4.3.2 Demonstrations and Trajectories (Down (D)) 80
4.3.3 Demonstrations and Trajectories (Extended (E)) 83
4.3.4 Demonstrations and Trajectories (Up (U)) 86
4.3.5 Performance Discussion on Static Arm Motions 89

5 Conclusion and Future Work **90**

References **91**

List of Figures

1.1 A time-lapse demonstration of a Mobile Manipulator with Multi-floor Operation by taking the elevator with the assistance of pre-fixed Augmented Reality University of Cordoba Markers (ArUco Markers) 2

1.2 Eye-in-Hand Field of View (FOV) + base ZED stereo camera FOV for Waterloo Steel Robot (in an office room [2]) 3

2.1 Extended framework of Visual-SLAM (V-SLAM) fusion (with Inertial Measurement Unit (IMU), Global Positioning System (GPS), Ultra Wide-Band (UWB), and more) 5

2.2 Onion diagram of visual-based Simultaneous Localization And Mapping (SLAM) systems 6

2.3 Three different camera configurations with D455 stereo camera module with built-in IMU: monocular setup can be achieved by utilizing RGB camera only; stereo setup can be achieved by disabling the active infrared projector; depth setup can be achieved by utilizing the IR stereo and the active IR projector. 7

2.4 Illustrations of front-end difference between indirect and direct methods . . 9

2.5 Trajectory comparison of VIO with/without loop-closure module 11

2.6 Examples of factor graph formulation [3]. 15

2.7 A toy SLAM example with three robot poses and two landmarks. 16

2.8 Simplified VINS-Mono block diagram illustration 19

2.9 VINS-Mono Front-End: Monocular Visual Processing Pipeline 20

2.10 VINS-Mono Initialization: Structure From Motion (SfM) and Visual-Inertial Alignment 23

2.11 D455 Intel Realsense Camera (D455) End-Effector (EE) IMU Gyroscope Characteristics via Allan Variance [1] (AVar) 28
2.12 D455 EE IMU Accelerometer Characteristics via AVar 28
2.13 D455 Calibration with Kalibr Toolbox [4] (Kalibr) 29
2.14 D455 EE Camera Model (MEI) Intrinsic Calibration with Kalibr 30
2.15 D455 EE Camera Visual-IMU Extrinsic Calibration with Kalibr 30
2.16 WAM Arm From Barrett Inc. (WAM) Kinematics Model 40
2.17 Intel RealSense Cameras [5] 41
2.18 Intel RealSense Multi-Camera IR Camera Interference 42

3.1 Our mobile manipulator comprises a 3-Degree-of-Freedom (DOF) SUMMIT XL From Robotnik Inc. (SUMMIT) mobile base ① and a 7-DOF WAM Arm From Barrett Inc. (WAM) robotic arm ②. It has two D455 cameras configured with only monocular-RGB (RGB) and IMU; one located at the EE ③ and the other at the base ④. A Vicon MoCap system ⑤ is used for Ground Truth (GT). [6] 46
3.2 Frames and their relations: Solid arrows show constrained transformation; dashed arrows show unconstrained transformation; dotted lines show time-varying trajectories with a starting knot [6]. To simplify matters, we assume that the state estimation is performed at the camera's IMU frame for both the base and the EE (i.e. $\chi^w_{b_B}, \chi^w_{b_E} \in \mathbf{SE}(3)$). The kinematic parameters for the arm are calibrated offline based on the Vicon MoCap and 3D model at its zero-configuration, where the arm is straightened upwards ($g_{st}(\mathbf{0}) \in \mathbf{SE}(3)$). 49
3.3 Illustration of the sliding window dual factor graph structure with the arm odometry factor coupling two states, with $\mathbf{x}_{b_k}$ as the primary state and $\mathbf{x}'_{b_k}$ as the corresponding secondary state at frame k [6]. The bottom timeline shows an approximated representation of measurement timings and sampling strategy; IMU data is sampled and pre-integrated between two frames, images are software synchronized, and arm joint states are sampled at each frame. 50
3.4 The block diagram illustrates the modified pipeline for the proposed dual-VIO with arm odometry. The blue blocks are the innovation parameters, differentiating the implementation difference of baseline and proposed tightly-coupled VIO. 54

4.1 Hardware configuration diagram 59

4.2 The environment of a cluttered facility for experiments; ② uneven terrain , ③ reflective glass window, ④ low-feature illuminated ceiling, and ⑤ repetitive robots. The green masking-tape (①) on the floor helps operators to locate the robot starting position without collision during the experiment [6]. 60

4.3 A collection of six different arm motions with base fixed. Figure 4.3(a) showcases the home position of the arm without motion; Figure 4.3(b), 4.3(c), and 4.3(d) demonstrate static arm motion by moving the arm to the desired configuration [i.e. Down (D), Extended (E), Up (U)] initially and retracting back to the Home (H) [4.3(a)] at the end of the base movements. Figure 4.3(e) and 4.3(f) describe dynamic oscillating motions by sweeping the arm Left-And-Right (LR) and Up-And-Down (UD), respectively [6]. . 61

4.4 A collection of seven base motions with arm fixed at the Home (H) position. In particular 4.4(a) showcases both reverse and forward motions, while 4.4(b) and 4.4(c) showcases Triangle (TRI) and Square (SQR) motions. Figure 4.4(d), 4.4(e), and 4.4(f) illustrates Spin (SPI), Circle (CIR), Waggle-Bee-Dance (BEE) motions, respectively. 62

4.5 Vicon MoCap initial alignment 63

4.4 A collection of 7 time-lapsed images in the first columns (a,d,g,j,m,p,s) and 14 trajectory plots in the second and third columns for dual-VIO (without loop-closure) in seven different base motions (i.e., Forward (FWD), Reverse (RVR), Spin (SPI), Triangle (TRI), Square (SQR), Circle (CIR), and Waggle-Bee-Dance (BEE)) with arm sweeping Left-And-Right (LR) [6] . . 69

4.5 Boxplots for ATE errors in terms of translation and orientation components for both EE and Base VIOs (without loop closure) with seven different base maneuvers while arm waving Left-And-Right (LR) [6] 70

4.4 A time-lapsed image (a) and trajectory plots (b,c) for dual-VIO (without loop-closure) in Waggle-Bee-Dance (BEE) base motion with arm sweeping Up-And-Down (UD) [6] 73

4.5 Boxplots for ATE errors in terms of translation and orientation components for both EE and Base VIOs (without loop closure) with seven different base maneuvers while arm waving Up-And-Down (UD) [6] 74

4.5 A time-lapsed image (a) and trajectory plots (b,c) for dual-VIO (without loop-closure) in Waggle-Bee-Dance (BEE) base motion with arm at Home (H) position 78

4.6 Boxplots for ATE errors for both base and EE VIOs (without loop closure) in terms of translation and orientation components at the Home (H) configuration. 79
4.6 A time-lapsed image (a) and trajectory plots (b,c) for dual-VIO (without loop-closure) in Waggle-Bee-Dance (BEE) base motion with arm at Down (D) position 81
4.7 Boxplots for ATE errors for both base and EE VIOs (without loop closure) in terms of translation and orientation components at the Down (D) configuration. 82
4.7 A time-lapsed image (a) and trajectory plots (b,c) for dual-VIO (without loop-closure) in Waggle-Bee-Dance (BEE) base motion with arm at Extended (E) position 84
4.8 Boxplots for ATE errors for both base and EE VIOs (without loop closure) in terms of translation and orientation components at the Extended (E) configuration. 85
4.8 A time-lapsed image (a) and trajectory plots (b,c) for dual-VIO (without loop-closure) in Waggle-Bee-Dance (BEE) base motion with the arm at Up (U) position 87
4.9 Boxplots for ATE errors for both base and EE VIOs (without loop closure) in terms of translation and orientation components at the Up (U) configuration. 88

List of Tables

2.1 Optimization Algorithms for Nonlinear Least Squares Minimization 18

2.2 Calibrated AVar Results for base (B) and End-Effector (E) D455 cameras with each measured at two different places: one on the table and the other at the base or EE. 27

4.1 Choreography Configuration for Seven Base Movements [6] 62

4.2 Calibration configs used for the mobile base and EE D455 64

4.3 Mean ATE in translation $\mathbf{e_p}$ at VIOs with dynamic locomotion 75

4.4 Mean ATE in orientation $\mathbf{e_q}$ at VIOs with dynamic locomotion 75

List of Abbreviations

3DGS 3D Gaussian Splatting. 6

A-SLAM Active-SLAM. iii, vii, 4, 14, 44, 45

AC-SLAM Active-Collaborative-SLAM. 14

ALVIO Adaptive Line And Point Feature-Based Visual Inertial Odometry [7]. 13

AMCL Adaptive Monte Carlo Localization. 1

ARM Advanced RISC Machine. 59

ArUco Markers Augmented Reality University of Cordoba Markers. x, 1, 2, 45

ATE Absolute Trajectory Error. xii–xiv, 65, 66, 70, 74, 75, 79, 82, 85, 88, 89

AVar Allan Variance [1]. vii, xi, xiv, 27, 28, 43, 58, 64

BA Bundle Adjustment. 12, 23–25, 51, 55

BEE Waggle-Bee-Dance. xii, xiii, 62, 69, 73, 75, 76, 78, 81, 84, 87

BoW Bag-of-Words. 10, 12

C-SLAM Collaborative-SLAM. 14

CAN Controller Area Network. 59

Ceres Ceres Solver. 23, 25

CIR Circle. xii, 62, 69, 72, 73, 75, 76, 78, 79, 81, 84, 87

CNN Convolutional Neural Network. 9

CV Computer Vision. 6

D Down. ix, xii, xiii, 61, 77, 80–82

D435i D435i Intel Realsense Camera. 41

D455 D455 Intel Realsense Camera. xi, xiv, 27–30, 41, 43, 46, 58, 64

DOF Degree-of-Freedom. xi, 45, 46, 52

DTAM Dense Tracking And Mapping. 8, 9, 11

E Extended. ix, xii, xiii, 61, 77, 83–85

EE End-Effector. xi–xiv, 1, 3, 27–30, 40, 45–47, 49, 50, 52, 53, 58, 62, 64, 67, 69–71, 73–77, 79, 80, 82, 83, 85, 86, 88–90

EKF Extended Kalman Filter. 10

FK Forward Kinematics. 40, 52, 53

FOV Field of View. x, 3, 45, 48, 90

FWD Forward. xii, 61, 62, 67, 69, 71, 73, 75, 77, 79, 80, 83, 86

GN Gauss-Newton. 18

GNSS Global Navigation Satellite System. 47

GPS Global Positioning System. x, 5, 47

GT Ground Truth. xi, 13, 46, 59, 63, 65, 67

H Home. ix, xii, xiii, 27, 28, 61, 62, 77–79

IMU Inertial Measurement Unit. x, xi, xix, 3, 5, 7, 8, 10, 13, 20–30, 41, 43–48, 50–52, 58, 63, 64

Kalibr Kalibr Toolbox [4]. xi, 29, 30, 58, 64

KF Kalman Filter. 10, 48

Kimera an Open-Source Library For Real-Time Metric-Semantic SLAM [8]. 6, 11, 13

KLT Kanade-Lucas-Tomasi. 20

L515 L515 Intel Realsense Camera. 41

LiDAR Light Detection And Ranging. 1, 41, 44, 45

LM Levenberg-Margquardt. 18

LR Left-And-Right. viii, ix, xii, 61, 67–71, 75, 76

MAP Maximum-a-Posteriori. 10, 15, 17

MHE Moving Horizon Estimator. 47

MLE Maximum Likelihood Estimation. 16

MoE Mixture of Experts. 48

MPC Model Predictive Control. 39

MSCKF Multi-State Constraint Kalman Filter [9]. 10, 13, 44, 47

NeRF Neural Radiance Field. 6

NICE-SLAM Neural Implicit Scalable Encoding For SLAM. 11, 12

OKVIS Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimization. 11

OpenCV Open Computer Vision [10]. 20, 24

ORB-SLAM Oriented FAST And Rotated BRIEF SLAM [11]. 13

PDF Probability Density Funcstion. 15

PDL Powell's Dogleg. 18

PGO Pose-Graph Optimization. 15

PnP Perspective-n-Point. 23, 24

PREEMPT-RT Pre-Emptive Real-Time. 59

RGB RGB. xi, 43, 46, 58

ROS Robot Operating System. 27, 58, 59

ROVIO Robust Visual Inertial Odometry Using a Direct EKF-Based [12]. 13

RTE Relative Trajectory Error. 66

RVR Reverse. xii, 61, 62, 68, 69, 71, 75, 77, 80, 83, 86

SD Steepest Decent. 18

SfM Structure From Motion. x, 15, 22, 23, 45

SLAM Simultaneous Localization And Mapping. iii, x, xv, xvii, xix, 3, 5–7, 11–16, 44, 45, 66, 90

SOTA State-of-The-Art. vii, 12

SPI Spin. xii, 61, 62, 68, 69, 71–73, 75–77, 80, 83, 86

SQR Square. xii, 62, 68, 69, 72, 73, 75, 76, 78, 79, 81, 84, 87

SUMMIT SUMMIT XL From Robotnik Inc.. xi, 45, 46

Summit-PC Adlink MXE-211. 59

ToF Time-of-Flight. 8, 44

TRI Triangle. xii, 62, 68, 69, 72, 73, 75, 76, 78, 81, 84, 87

U Up. ix, xii, xiii, 61, 77, 86–88

UAV Unmanned Aerial Vehicle. 1, 47

UD Up-And-Down. viii, ix, xii, 61, 67, 71–76

UDP User Datagram Protocol. 59

UGV Unmanned Ground Vehicle. 1

UWB Ultra Wide-Band. x, 5, 11, 12, 47

V-SLAM Visual-SLAM. x, 5–7, 10, 11, 15, 29, 44

VI-SLAM Visual-Inertial-SLAM. 44

Vicon MoCap Vicon Motion Capture System. iv, xi, xii, 46, 49, 58, 59, 63, 67, 79

VINS Visual Inertial Navigation System. iii, xix, 10, 13, 42

VINS-Fusion Stereo Visual Inertial Navigation System (VINS) Fusion. 13, 14, 58, 67

VINS-Fusion-GPU GPU-Accelerated VINS Fusion. 13, 14

VINS-Fusion-IMU Stereo VINS With IMU Fusion. 13, 14

VINS-Mono Monocular VINS. iii, vii, viii, x, 9–14, 19, 20, 23, 25, 27, 29, 30, 43, 45, 54, 64

VIO Visual Inertial Odometry. iii, vi, viii, x–xiv, 4–14, 19, 21, 22, 27, 41, 44–55, 58, 63, 66, 67, 69–71, 73–90

VO Visual Odometry. 5–8, 12–14, 44, 48

WAM WAM Arm From Barrett Inc.. viii, xi, 27, 40, 45, 46, 59

WAM-PC Jetson Orin. 59

ZOH Zero-Order-Hold. 21

List of Symbols

$\mathcal{N}(\mu, \sigma^2)$ A Gaussian distribution with the mean μ and the covariance $\Sigma \equiv \sigma^2$.

$\mathbf{SE}(\star)$ Special Euclidean Space.

$\mathbf{SO}(\star)$ Special Orthogonal Space.

$\| \bullet \|_1$ The l-1 norm operator.

$\| \bullet \|_2$ The l-2 norm operator.

$\| \bullet \|_F$ The Frobenius norm operator, l-2 norm of a matrix.

$\mathcal{G}$ Lie Group.

$\mathfrak{g}$ Lie Algebra.

$\mathbf{SE}(3)$ Special Euclidean Group in 3 Dimensions.

$\mathbf{SO}(3)$ Special Othorgonal Group in 3 Dimensions.

χ State in $\mathbf{SE}(3)$.

$\mathbf{x}$ optimization state vector.

$\mathbf{X}$ optimization horizon in sliding window.

$(\bullet)^\wedge$ Hat operator.

$(\bullet)^\vee$ Vee operator.

$\hat{\star}$ A unit vector $\{\hat{\star} \,|\, \| \star \| = 1\}$.

$\widehat{\star}$ An estimator.

$\bar{\star}$ A true value.

$T^{\bullet}_{\star}$ The fixed coordinate transform between $\bullet$ and $\star$.

$\chi^{\bullet}_{\star}$ Pose state representation in frame $\bullet$ from body $\star$.

$\mathbf{x}'$ Complementary optimization state vector.

Chapter 1

Introduction

1.1 Motivation

Consider a demonstration of a mobile manipulator operating in a multi-floor indoor environment as shown in the time-lapse illustrations in Figure 1.1[1]. Within this context, the mobile platform effectively integrates Light Detection And Ranging (LiDAR) technology to execute precise localization via Adaptive Monte Carlo Localization (AMCL) with a pre-established map. Meanwhile, the manipulator equipped with the End-Effector (EE) camera locates the ArUco Markers and estimates the relative spatial positions of buttons to interact with. Throughout the experiments behind the scenes, the demonstration highlights the inherent challenges associated with accurate LiDAR-based localization in confined passageways and corridors, particularly when encountering obstructions while operating in close proximity to walls. Notably, the efficacy of the localization process profoundly impacts the success rate of smooth execution of the multi-floor task (i.e., moving from the third-floor laboratory to the first floor using the elevator).

Visual-based localization, on the other hand, offers a detailed and comprehensive understanding of the environment by leveraging visual cues and spatial information. This approach has been suitable for many indoor and outdoor Unmanned Aerial Vehicle (UAV) and Unmanned Ground Vehicle (UGV). In particular, visual perception presents an attractive option for indoor applications, as it is less susceptible to challenges common in outdoor environments such as weather conditions, complex lighting, changing terrains, textureless surfaces, and unbounded scenes.

[1]See the full multi-floor demo video on YouTube: `https://www.youtube.com/watch?v=rkKu2eE83Ss`



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 1.1: A time-lapse demonstration of a Mobile Manipulator with Multi-floor Operation by taking the elevator with the assistance of pre-fixed ArUco Markers

However, visual localization (or more generally visual-based Simultaneous Localization And Mapping (SLAM)) faces other challenges related to sensor occlusions and interference, visual aliasing, reflections, dynamic objects, and scale ambiguity. In the above context, as illustrated in Figure 1.1(e), the base camera poses notable challenges in visual perception due to its limited field of view and persistent occlusions. While augmenting the field of view through additional cameras is a viable approach, data bandwidth management and fusion become prohibitively resource-intensive with constrained onboard computing resources. In the indoor environment, it is inevitable to experience partially occluded views and dynamic scenes with visual ambiguity. Consequently, achieving a comprehensive visual mobile platform remains a significant challenge.

Let us now consider an articulated mobile manipulator with two pairs of a monocular camera and a low-cost IMU: one pair at the EE and the other at the mobile base, as depicted in Figure 1.2. The camera at the EE provides flexible and extended perception to observe the environment, aiding in navigation and interaction. In addition, the actuated manipulator provides proactive resolution of diverse perceptual challenges, e.g., actively searches for regions to relocalize when lost.

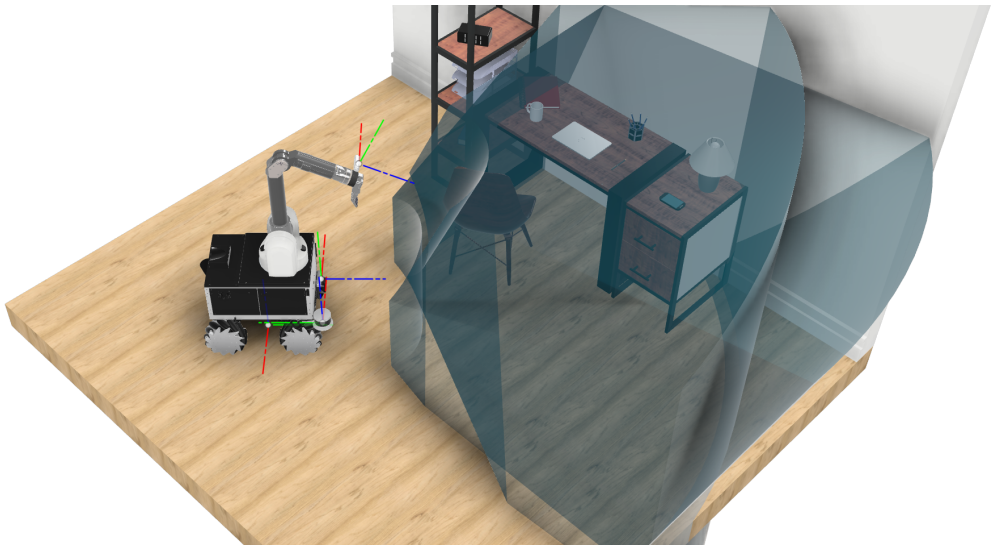


Figure 1.2: Eye-in-Hand FOV + base ZED stereo camera FOV for Waterloo Steel Robot (in an office room [2])

This objective underpins the primary motivation of this endeavour, namely, the development of a dual-visual-based mobile manipulator. This envisioned system utilizes dynamic views with enhanced spatial visibility from the end-effector camera through an actuated

manipulator and the static views from the base camera to achieve this goal, thereby enabling autonomous navigation in unfamiliar and complex indoor environments.

1.2 Contributions

We have conceived a novel tightly-coupled[2] sensor fusion technique by jointly optimizing dual monocular Visual Inertial Odometry (VIO) for an articulated mobile manipulator. This method effectively addresses the trade-off between robustness and computational complexity, while preserving the independence of each VIO for system redundancy and failure recovery. Furthermore, we conducted comprehensive experimental testing to validate the proposed approach, revealing its potential to facilitate multi-VIO fusion and system redundancy within the domain of A-SLAM.

The main achievement of this thesis has been published under the title, "A New Tightly-Coupled Dual-VIO for a Mobile Manipulator With Dynamic Locomotion", in the IEEE/ASME Transactions on Mechatronics (TMECH) and was presented at the Advanced Intelligent Mechatronics (AIM 2024) Conference in Boston [6].

1.3 Thesis Outline

This thesis is organized in a constructive manner to pave the way for a robust dual Visual Inertial Odometry (VIO) fusion to serve as the foundation of the visual state estimator tailored for a mobile manipulator. It commences with Chapter 2, providing the essential backgrounds and comprehensive literature review about related works. Chapter 3 is the primary technical contribution of this thesis to tackle the dual-VIO for robust localization throughout highly dynamic locomotions. Chapter 4 discusses the experimental setup and results for the proposed method. Lastly, Chapter 5 concludes the thesis.

[2]In some literature (e.g. [13]), the term "semi-tight-coupling" has been used, but we adopt "tight-coupling" as a more generally accepted term.

Chapter 2

Preliminary

2.1 Taxonomy of Visual Inertial Odometry (VIO)

In recent developments, visual-based Simultaneous Localization And Mapping (SLAM) has enabled robots to map and navigate unknown environments using visual data. V-SLAM and Visual Odometry (VO) are often interchangeable, but with different objectives. The VO emphasizes more on the real-time state estimation, whereas the V-SLAM focuses on global and consistent estimation [14]. In brief, one may consider VO to be a full SLAM after closing the loop (a.k.a., loop-closure) [15]. Figure 2.1 illustrates three major components

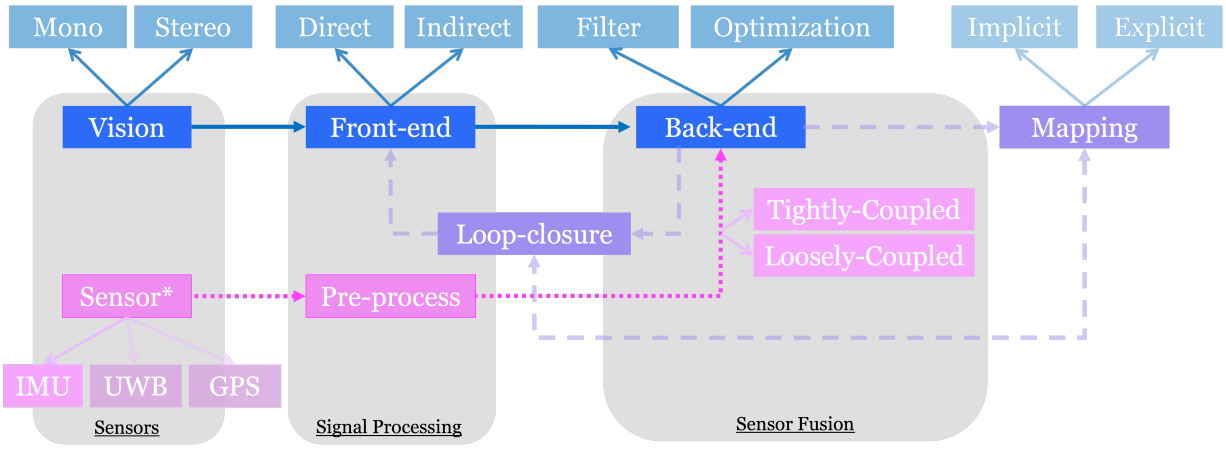


Figure 2.1: Extended framework of V-SLAM fusion (with IMU, GPS, UWB, and more)

(in blue blocks) and two optional components (in purple blocks) for classical V-SLAM/VO algorithms: vision sensors, front-end, back-end, loop-closure, and mapping.

In broader terms, the front-end module corresponds to sensor signal processing, while the back-end is analogous to sensor fusion (as depicted in gray blocks in Figure 2.1). The mapping module is less focused in many literature surveys for VO/VIO and is often considered an integral part of the loop closure that enables V-SLAM and VO to function as a complete SLAM system. However, as there is growing popularity on 3D reconstruction in Computer Vision (CV) community (e.g., Neural Radiance Field (NeRF) and 3D Gaussian Splatting (3DGS)), we denote mapping as a standalone module (similar to an Open-Source Library For Real-Time Metric-Semantic SLAM [8] (Kimera)).

However, the reliance on visual data alone (VO) can be limiting, particularly in environments where visual features are sparse or conditions are not favourable for reliable visual perception (e.g., occlusion, poor lighting, etc.). One effective approach involves integrating additional sensor data into the existing framework, as depicted by the pink blocks in Figure 2.1. In particular, researchers have increasingly turned to Visual Inertial Odometry (VIO), which integrates visual data with inertial measurements. This fusion not only enhances the robustness and accuracy of localization but also allows for more reliable performance in diverse and challenging environments. By leveraging the strengths of both visual and inertial sensors, VIO represents a significant advancement from traditional V-SLAM, offering a more comprehensive solution for real-time autonomous navigation [13].

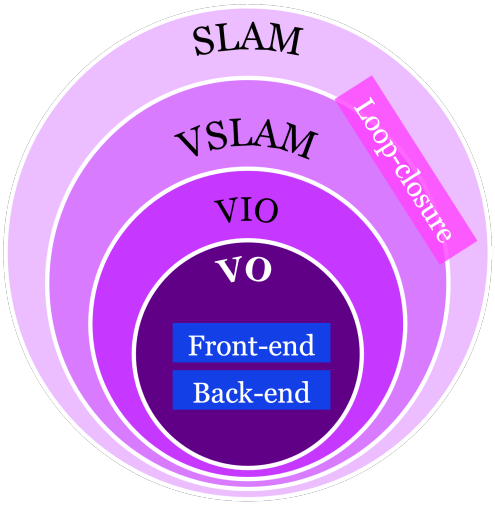


Figure 2.2: Onion diagram of visual-based SLAM systems

As a result, VIO has become a popular and critical foundation for a V-SLAM. As

illustrated in Figure 2.2, the front-end and back-end are the common submodules for both VO and VIO systems. The loop-closure module can be integrated into VO/VIO, serving as the key ingredient towards a full-SLAM system.

In this section, a detailed categorization of each module shown in Figure 2.1 will be discussed with a focus on VIO, providing the backgrounds needed for the problem and formulation.

2.1.1 Camera Modules Commonly Used in VIO

VO and VIO systems are often categorized based on the vision sensor type, as different sensors affect the design and performance of the front end and the back end. Typically, there are three major camera types commonly used in V-SLAM: monocular camera, stereo camera, and depth camera. Figure 2.3 shows an example of such cameras.

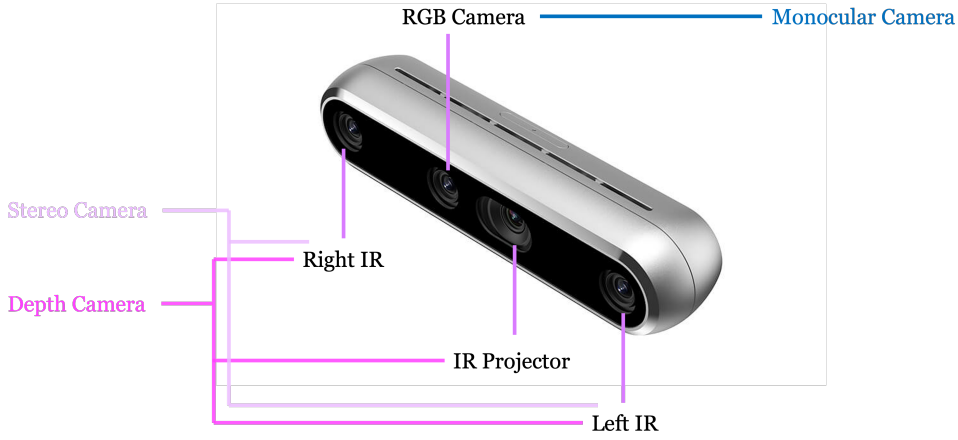


Figure 2.3: Three different camera configurations with D455 stereo camera module with built-in IMU: monocular setup can be achieved by utilizing RGB camera only; stereo setup can be achieved by disabling the active infrared projector; depth setup can be achieved by utilizing the IR stereo and the active IR projector.

The merits and demerits of these three different types of cameras are summarized below;

- Monocular Camera: Low-cost and simple layout but suffers from scale ambiguity and cannot obtain depth when stationary or during pure rotation.

- Stereo Camera: Computes depth through visual measurements directly from pixels but requires more computational resources.
- Depth Camera: Directly obtains depth through infrared structured light or Time-of-Flight (ToF) but requiring external emitting sources of patterns may cause sensor interference when multiple cameras are deployed.

In this thesis, we focus on monocular VIO due to its inherent simplicity and cost efficiency while preserving wealth of visual information [16]. Although stereo and RGB-D cameras offer superior 3D reconstruction capabilities, they necessitate higher data bandwidth. Additionally, RGB-D cameras, acting as active sensors, may induce interference in scenarios involving multiple modules within the field.

Conversely, employing a monocular camera for pose estimation presents formidable challenges, as it relies exclusively on the triangulation between successive-frame features, as opposed to calibrated stereo triangulation [17]. Consequently, the pairing of a monocular camera with an IMU for VIO is much more common as opposed to VO. Whereas a stereo camera can operate standalone as VO or in conjunction with IMU as VIO.

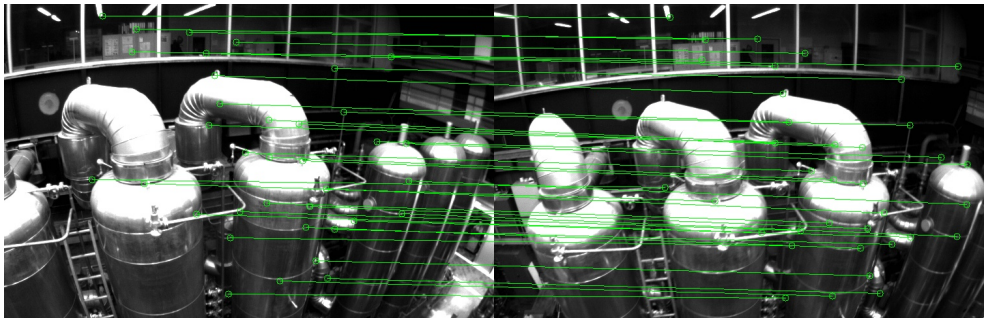
2.1.2 Front-End: Indirect vs. Direct

The front-end module is responsible for feature extraction and tracking, which can be categorized into two mainstream methods (as shown in Figure 2.4): Indirect and Direct.

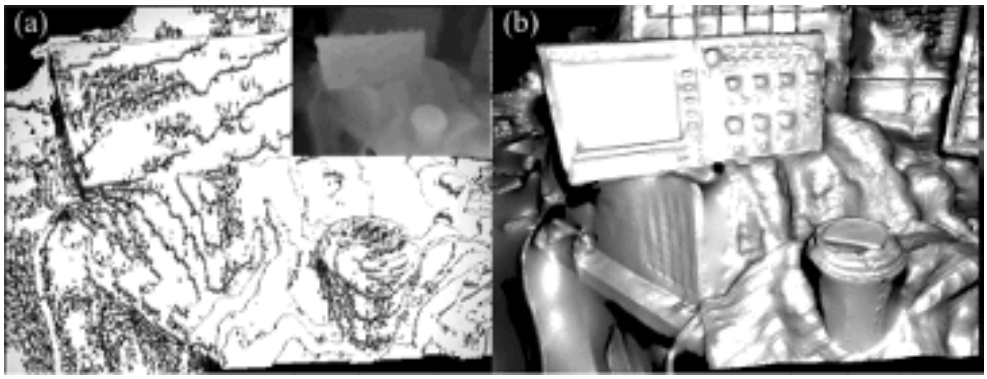
The indirect method (as shown in Figure 2.4(a)), also known as the sparse method (or feature-based method), extracts distinctive geometric features (such as points, lines, or planes) from images, tracks them over time, and builds an implicit map while retaining important and unique information. This method has been widely used [16]. These features are often extracted with well-established descriptors (such as SIFT [20], SURF [21], ORB [22]), forming a compressed representation of the scene.

The direct method (as illustrated in Figure 2.4(b)), also known as the dense method (e.g., Dense Tracking And Mapping (DTAM) algorithm in 2011 [19]), directly computes the photometric error (the variance in pixel intensities) on all pixels between consecutive frames or stereo frames [16]. The direct method is beneficial in featureless regions (such as the blue sky) and can construct dense maps directly.

However, the direct method is often susceptible to noise and illumination, while the feature-based method provides robustness against environmental changes. As a result, the indirect methods have been pre-dominant in VO and VIO with less focus on map



(a) Indirect (Sparse) Method (VINS-Mono [18])



(b) Direct (Dense) Method (DTAM [19])

Figure 2.4: Illustrations of front-end difference between indirect and direct methods

reconstruction. Meanwhile, as computer vision advances with Convolutional Neural Network (CNN) [23], more researchers are adopting different learning-based techniques to partially or fully integrate into the front-end (e.g., a semantic feature-based localization method [24].)

Overall, the performance of the front end has a direct impact on the accuracy of the back end [25]. Hence, many works focused on improving the front-end with additional features (e.g. point and line features [26], shape features [27]), or image enhancements [28]. However, some researchers realized that the additional features show slight improvement only in feature-rich scenarios with a large sacrifice on the additional computation resources. In a changing environment, VIO becomes more susceptible to the dynamic elements. As a result, some works proposed to handle motion discrepancy through outlier detections and rejections [29] [30].

2.1.3 Back-End: Filter vs. Optimization

The back-end module is often associated with sensor fusion, which can be categorized into filter-based and graph-based optimization; the former relies on the Bayes filter while the latter relies on Maximum-a-Posteriori (MAP) estimation, respectively. When we assume simple linear Gaussian additive noise, MAP inference becomes equivalent to Kalman smoothing, and the tracking problem leads to the Kalman Filter (KF) [3]. In practice, the filter-based method performs better with high-rate, locally consistent data (IMU), while graph-based techniques are more efficient with low-rate, globally consistent, sparse features (images) [31, 32].

Filter-based methods are the most common approach in sensor fusion. It estimates the current position gradually through nonlinear probabilistic filters like Extended Kalman Filter (EKF). Fundamentally, the filter-based methods rely on the principle of Bayes filters as their foundational form with certain relaxation [33], such as Gaussian approximation with EKF (e.g. Multi-State Constraint Kalman Filter [9] (MSCKF) [9]). This implies that the computational complexity and the dimensionality do not vary at each step due to its continuous marginalization of past states. But its performance can be sub-optimal due to the early fix of linearization points [34].

The graph-based approaches build a graph where nodes represent the camera poses, and landmarks and edges represent constraints between them based on motion models and observed feature correspondences. The graph is then optimized to find the most likely configuration of camera poses and landmark locations that satisfy all the constraints. Hence, it is also known as optimization-based VIO, and the graph provides an intuitive representation of the Markovian chain. As a result, most graph-based methods are derived from Bayesian inference and draw on recent advances in nonlinear optimization solvers (e.g. Ceres [35] used in Monocular VINS (VINS-Mono) [18]).

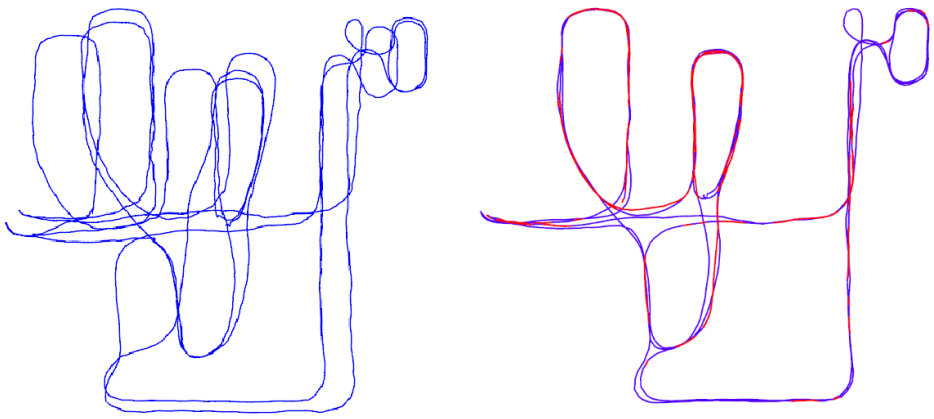
2.1.4 Mapping: Implicit vs. Explicit

In some literature, V-SLAM is distinguished by its mapping strategy, which can be sparse or dense map reconstruction [16]. In most feature-based (indirect) front-end approaches, the mapping strategy is always sparse while dense for some of the direct front-end. To better differentiate the mapping strategy from the front-end classifications used for VIO, we would like to categorize the mapping according to their representation types: implicit or explicit. Implicit mapping aligns with sparse mapping, including feature-based Bag-of-Words (BoW) (e.g., VINS-Mono [18]) and recent neural-network approaches (e.g., deep

neural networks [36], Neural Implicit Scalable Encoding For SLAM (NICE-SLAM) [37]). Conversely, explicit mapping pertains to dense mapping and encompasses occupancy grid or mesh-based 3D reconstruction (e.g., Kimera and DTAM). Implicit and explicit mapping serve different purposes: the former offers a concise representation facilitating loop closure to correct large drifting errors and fast relocalization, and the latter provides detailed structural information for planning and surveying purposes. Hence, more comprehensive V-SLAM (e.g., Kimera) incorporates both to enhance the overall reconstruction while improving the localization.

2.1.5 Loop-closure

In many VIO works (such as VINS-Mono [18] and Kimera), the loop-closure is implemented as a concurrent module that runs parallel with the base VIO module (front-end and back-end) at a much lower rate. Loop-closure detection helps to eliminate cumulative trajectory errors and map reconstruction errors upon revisiting the same landmarks [16] (as shown in Figure 2.5). Hence, it is a critical module to enable global positioning for relocalization and map merging. However, the loop-closure can easily be afflicted by the visual-aliasing problem, where two locations may have similar features, resulting in a corrupted closure of the global trajectory [38]. This module is often unused if global positioning sensors are available to correct the drift (e.g., UWB-aided multi-robot state estimation [39]).



(a) OKVIS [40] [18] (b) VINS-Mono with loop closure [18]

Figure 2.5: Trajectory comparison of VIO with/without loop-closure module

The loop-closure step computes the global pose graph optimization in a form of full Bundle Adjustment (BA), and is often implemented as an additional module running concurrently to generate a separate global trajectory. All the keyframes and optimized VIO poses are monitored in the loop-closure module, and all poses are optimized globally when aliased features are detected between two keyframes over the time. This module provides a globally consistent trajectory, minimizing the drift errors from the real-time VIO odometries. As a result, a VIO with a loop-closure module is often considered as a full SLAM [8].

The loop closure pipeline often depends on the mapping representation and is mostly utilized in implicit mapping. In the BoW model (implicit mapping, VINS-Mono [18]), the loop closure can be performed through vocabulary matching, created by clustering the descriptors from a large set of images (quantization). In the neural network approach (e.g., deep neural networks [36], NICE-SLAM [37]), the network model is often overfitted by a training set, and the matching happens within the neural implicit representation of the surroundings.

2.1.6 Sensor Fusion: Tightly-Coupled vs. Loosely-Coupled

As illustrated in Figure 2.1, VIO is a form of sensor fusion with VO. In addition, many recent works fuse additional sensors (e.g., UWB) into VIO to correct large drifts. We can categorize VIO based on sensor fusion: tightly-coupled and loosely-coupled.

A tightly coupled approach fuses the states of each sensor after independent signal pre-processing into one joint optimization to obtain the pose information. This approach utilized visual and other sensor measurements jointly to obtain higher pose estimation accuracy and has been the mainstream fusion method [16].

The loosely coupled approach blends poses from different estimators through a weighted combination. It is a simple and efficient way to unite various sensors and odometry algorithms through a filter-based method, and thus widely used in classical multi-modal fusion (e.g. [41]).

2.1.7 Benchmark on State-of-The-Art (SOTA) VIO

In 2021, a group of researchers conducted a study of a variety of State-of-The-Art (SOTA) VIO and VO on Jetson platforms [17]. The sensor suite contains D435i camera with 640×480 resolution at 30 [Hz], Pixhawk 4 mini IMU with 3-axes accelerometer and 3-axes

gyro at 100 [Hz], and Ground Truth (GT) OptiTrack Mocap at 50 [Hz]. The compared algorithms contain mono and stereo, covering from VIO and VO: VINS-Mono, Stereo VINS Fusion (VINS-Fusion), Adaptive Line And Point Feature-Based Visual Inertial Odometry [7] (ALVIO), Oriented FAST And Rotated BRIEF SLAM [11] (ORB-SLAM), and Robust Visual Inertial Odometry Using a Direct EKF-Based [12] (ROVIO). All of these algorithms fall into the category of sparse VIO, in which feature key points are selected rather than the direct photometric loss. The results can be summarized as follows:

Real-time Computing Performance

1. In terms of CPU usage, Kimera and ORB-SLAM stereo algorithms exhibited relatively higher intensity and required a larger number of features per frame compared to other algorithms.

2. VINS-Mono, ALVIO, and ROVIO demonstrated no significant differences across all Jetson boards, indicating adequate resource allocation. Additionally, GPU-Accelerated VINS Fusion (VINS-Fusion-GPU) exhibited stability, relying primarily on GPU resources, suggesting sufficient resources to support uninterrupted operation.

3. Oriented FAST And Rotated BRIEF SLAM [11] (ORB-SLAM), Multi-State Constraint Kalman Filter [9] (MSCKF), and an Open-Source Library For Real-Time Metric-Semantic SLAM [8] (Kimera) displayed larger errors contingent on the onboard resources, with the Xavier board yielding relatively superior accuracy. Correspondingly, VINS-Fusion and Stereo VINS With IMU Fusion (VINS-Fusion-IMU) revealed similar performance degradation on less robust Jetson boards (NX versus Xavier).

4. Notably, on the TX2 platform, VINS-Fusion-GPU delivered the best trajectory for rotational motion, leveraging GPU resources to address CPU limitations.

5. Stereo methods reliant solely on CPU processing manifested larger errors compared to monocular-based methods due to constraints in per-frame processing time.

6. Within the Xavier platform, which features enhanced CPU and memory performance relative to the TX2, stereo methods surpassed monocular-based methods.

7. Overall, error rates were lower on the Xavier board than on the NX board, attributing improved hardware tZ reduced per-frame processing time.

Overall Estimation Performance

Among VINS-Fusion (stereo VO), VINS-Fusion-GPU (GPU accelerated), VINS-Mono (monocular VIO) and VINS-Fusion-IMU (stereo VIO), the robustness against rotations in the order of stereo VIO, stereo VO, and mono VIO. All three methods demonstrated excellent performance for yaw errors. Specifically, the VINS-Fusion shows a smaller error for normal rotation, while VINS-Fusion-IMU and VINS-Fusion-GPU exhibit a smaller error for fast rotation, IMU is specialized in rapid motion, whereas the camera is specialized in relatively slow motion [17].

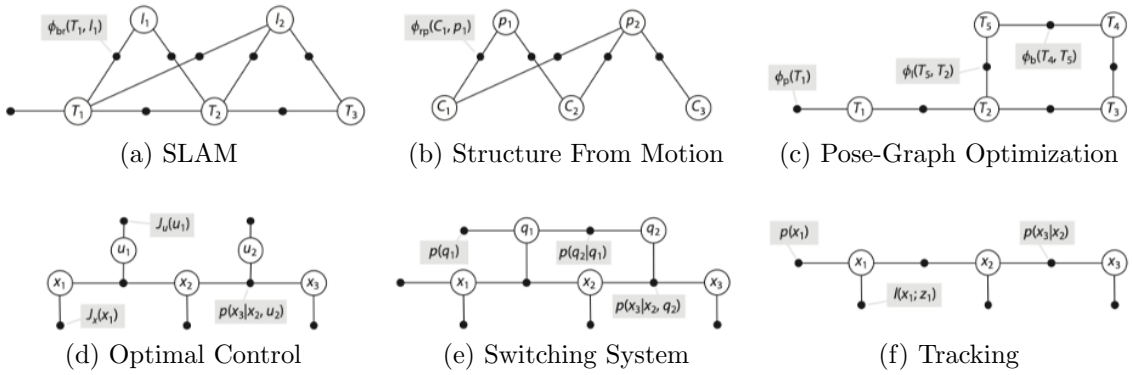
2.1.8 Towards A-SLAM

As the field of SLAM evolves, an increasing number of researchers are delving into the nuanced pursuit of finding an optimal equilibrium between the exploitation and exploration of unknown environments. This involves thoughtful integration of information theory, control theory, and reinforcement learning with SLAM [42]. For instance, the concept of Active-SLAM (A-SLAM) aims to perform autonomous navigation and exploration of such environments by the seamless closure of the planning and controller loop with SLAM [43] [44]. In parallel, Collaborative-SLAM (C-SLAM) amalgamates SLAM across multiple robots to expedite the process of exploration and exploitation through a methodology of division and conquering. Lastly, Active-Collaborative-SLAM (AC-SLAM) presents a fusion of these two approaches [42].

2.1.9 Factor Graph

Throughout the years, many topics in autonomous robotics, including estimation, mapping, planning, and control, have an optimization problem at their core. Typically, the objective in these problem domains comprises numerous localized factors or terms. Factor graphs present an elegant and profound topological structure for guiding the formulation of these problems into a graphical optimization [3].

Figure 2.6 illustrates different implementations of the factor graph for different problems. In brief, the factor graph is a graphical representation governing multiple domains with similar concepts for optimization-based problems. In particular, we are mostly interested in Simultaneous Localization And Mapping (SLAM) formulation (Figure 2.6(a)) of the factor graph for a mobile manipulator.



(a) SLAM (b) Structure From Motion (c) Pose-Graph Optimization

(d) Optimal Control (e) Switching System (f) Tracking

Figure 2.6: Examples of factor graph formulation [3].

Maximum-a-Posteriori (MAP) formulation with Bayes

In a SLAM problem, the goal for the robot is to infer the unknown environment while localizing itself simultaneously in the evolving environment from its sensors [45]. However, due to sensor noises and measurement uncertainty, the inferred state derived from sensors can hardly be the true state. At least, we can use the probability theory to assign a subjective degree of belief to uncertain events. This approach is analogous to our evaluation of knowledge pertaining to global politics through various media, influenced by subjective degrees of personal preference.

To model a believe over continuous and multivariate random variables $\mathbf{x} \in \mathbb{R}^n$, we can construct a Probability Density Funcstion (PDF):

$$p(\mathbf{x}) \qquad s.t. \int p(\mathbf{x}) d\mathbf{x} = 1 \tag{2.1}$$

Equivalently, to recover our unknown states $\boldsymbol{X}$ from observed measurements $\boldsymbol{Z}$, it is simply the conditional density $p(\boldsymbol{X}|\boldsymbol{Z})$. In V-SLAM, this probability density can become complex and highly dimensional due to the number of states to be estimated over time and the huge number of measurements. However, visual landmarks are often highly sparse, resulting in a sparse probability. Hence, one can factorize the density over a much smaller domain. A probabilistic graphical model, like a factor graph, provides an intuitive mechanism to describe and exploit these factors in the density.

The SLAM problem is equivalent to a MAP estimator. Additionally, by incorporating Bayes' law, we can express the posterior density $p(\boldsymbol{X}|\boldsymbol{Z})$ as the product of the measurement

density $p(\boldsymbol{Z}|\boldsymbol{X})$ and the prior $p(\boldsymbol{X})$ over the normalization factor $p(\boldsymbol{Z})$:

$$\boldsymbol{X}^{\mathrm{MAP}} = \arg\max_{\boldsymbol{X}} p(\boldsymbol{X}|\boldsymbol{Z}) = \arg\max_{\boldsymbol{X}} \frac{p(\boldsymbol{Z}|\boldsymbol{X})p(\boldsymbol{X})}{p(\boldsymbol{Z})} \tag{2.2}$$

To note, the denominator of the Bayes' rule does not depend on $\boldsymbol{X}$ [33]:

$$\boldsymbol{X}^{\mathrm{MAP}} = \arg\max_{\boldsymbol{X}} p(\boldsymbol{X}|\boldsymbol{Z}) \equiv \arg\max_{\boldsymbol{X}} p(\boldsymbol{Z}|\boldsymbol{X})p(\boldsymbol{X}) \tag{2.3}$$

Consider Maximum Likelihood Estimation (MLE) formulation to maximize the likelihood of states over the joint probability of measurements:

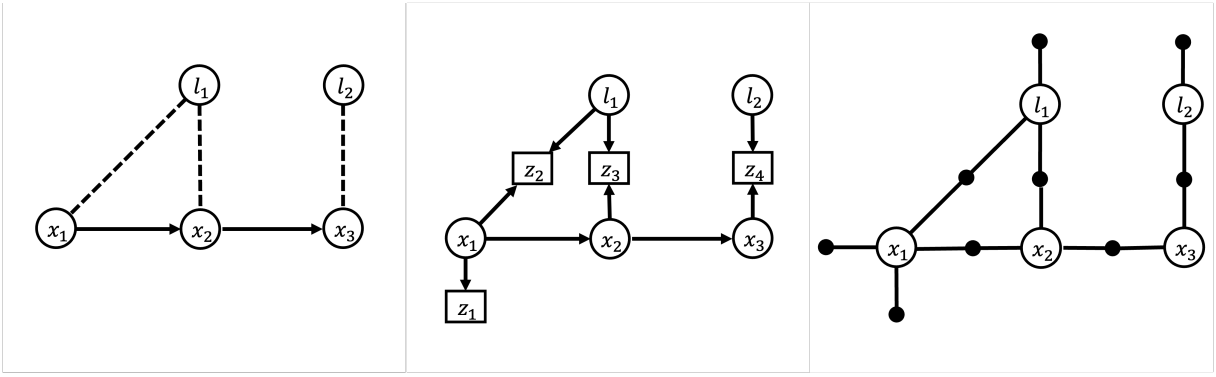
$$\boldsymbol{X}^{\mathrm{MLE}} = \arg\max_{\boldsymbol{X}} L(\boldsymbol{X}) \quad L(\boldsymbol{X}) = L(\boldsymbol{X};\boldsymbol{Z}) \equiv p(\boldsymbol{Z}|\boldsymbol{X}) = \prod_{i=1}^{n} p(\mathbf{z}_i|\boldsymbol{X}) \tag{2.4}$$

Combining the two (eq. (2.5) and eq. (2.4)), we can obtain:

$$\boldsymbol{X}^{\mathrm{MAP}} = \arg\max_{\boldsymbol{X}} p(\boldsymbol{X}|\boldsymbol{Z}) \equiv \arg\max_{\boldsymbol{X}} L(\boldsymbol{X};\boldsymbol{Z})\, p(\boldsymbol{X}) \tag{2.5}$$

An example of SLAM formulation

Let us now consider a toy problem [45] as illustrated in Figure 2.7. In Bayes' law, we can



(a) Illustration of SLAM Problem (b) Bayes Network (c) Factor Graph

Figure 2.7: A toy SLAM example with three robot poses and two landmarks.

equivalent the posterior term $p(\boldsymbol{X}|\boldsymbol{Z})$ through the MAP inference:

$$\begin{aligned} p(\boldsymbol{X}|\boldsymbol{Z}) \propto \quad & p(x_1)\, p(x_2|x_1)\, p(x_3|x_2) && (2.6)\\ & \times p(l_1)\, p(l_2) && (2.7)\\ & \times l(x_1; z_1) && (2.8)\\ & \times l(x_1, l_1; z_2)\, l(x_2, l_1; z_3)\, l(x_3, l_2; z_4) && (2.9) \end{aligned}$$

where, eq. (2.6) is the Markov chain on the poses x_1, x_2, x_3, presenting prior knowledge or derived from known control inputs; eq. (2.7) is the prior densities based on the landmarks l_1 and l_2 (only if there exists a prior map); eq. (2.8) is the likelihood corresponding to the absolute pose measurements on the first pose x_1 (initialization); eq. (2.9) is s a product of three likelihood of measurements at three poses.

To formulate an equivalent factor graph (a bipartite graph $F = (\mathcal{U}, \mathcal{V}, \mathcal{E})$), Figure 2.7(c) depicts the explicit representation: each small black node represents a factor $\phi_i \in \mathcal{U}$ in the posterior $p(\boldsymbol{X}|\boldsymbol{Z})$; a circle denotes a variable $x_j \in \mathcal{V}$; edges $e_{ij} \in \mathcal{E}$ forms an unique connection between two variables over a factor. Let $\boldsymbol{X}_i$ be the set of variable nodes adjacent to a factor node ϕ_i. As a result, the independence relationships are encoded by the edges e_{ij} of the factor graph, with each factor ϕ_i a function of adjacent variables $x_j \in \boldsymbol{X}_i$. Hence, a factor graph F defines the factorization of a global function $\phi(\boldsymbol{X})$ as

$$\phi(\boldsymbol{X}) = \prod_i \phi_i(\boldsymbol{X}_i) \tag{2.10}$$

For instance, the resultant factor graph for the toy problem becomes:

$$\begin{aligned} \phi(l_1, l_2, x_1, x_2, x_3) \propto \quad & \phi_1(x_1)\, \phi_2(x_2, x_1)\, \phi_3(x_3, x_2) && (2.11)\\ & \times \phi_4(l_1)\, \phi_5(l_2) && (2.12)\\ & \times \phi_6(x_1) && (2.13)\\ & \times \phi_7(x_1, l_1)\, \phi_8(x_2, l_1)\, \phi_9(x_3, l_2) && (2.14) \end{aligned}$$

Consequently, the MAP inference becomes the maximization of the product of all factor graph potentials:

$$\boldsymbol{X}^{\mathrm{MAP}} = \arg\max_{\boldsymbol{X}} \phi(\boldsymbol{X}) = \arg\max_{\boldsymbol{X}} \prod_i \phi_i(\boldsymbol{X_i}) \tag{2.15}$$

Equivalently, by taking the log and negative, Equation (2.15) is equivalent to minimizing the negative log.

$$\boldsymbol{X}^{\mathrm{MAP}} = \arg\min_{\boldsymbol{X}} \sum_i -\log \phi_i(\boldsymbol{X_i}) \tag{2.16}$$

Assuming a simple Gaussian noise model, now assume that all factors are of the form:

$$\phi_i(\boldsymbol{X}_i) \propto \exp\left\{-\frac{1}{2}\|h_i(\boldsymbol{X}_i) - z_i\|^2_{\Sigma_i}\right\} \tag{2.17}$$

where $h_i(\cdot)$ is the nonlinear measurement function. The resultant minimization problem (eq. (2.16)) becomes a non-linear least-squares:

$$\boldsymbol{X}^{\text{MAP}} = \arg\min_{\boldsymbol{X}} \sum_i \|h_i(\boldsymbol{X}_i) - z_i\|^2_{\Sigma_i} \tag{2.18}$$

Solving nonlinear least-squares problems requires an iterative solution starting from a decent initial guess. Table 2.1 listed four common algorithms. Steepest Decent (SD) is a simple method but suffers slow convergence at the minimum, while Gauss-Newton (GN) works best for near-quadratic problems but suffers divergence for general nonlinear problems. Levenberg-Margquardt (LM) combines the advantage of both GN and SD, resulting a general purpose solver. Powell's Dogleg (PDL) is a more efficient algorithm leveraging the trust region method to adaptively combine the result of GN and SD [45, 46].

Table 2.1: Optimization Algorithms for Nonlinear Least Squares Minimization

Algorithm	**Convergence**	**Quality**	**Robustness**	**Complexity**
Steepest Decent (SD)	Slow	Poor	Low	Simple
Gauss-Newton (GN)	Moderate	Poor	Moderate	Moderate
Levenberg-Margquardt (LM)	Fast	Great	High	Complex
Powell's Dogleg (PDL)	Fastest	Great	High	Complex

2.2 Anatomy of VINS-Mono

We are particularly interested in monocular-VIO due to its simplicity, as discussed in Section 2.1.2. Among all existing VIOs, VINS-Mono has already demonstrated its robust performance with fewer resources [47], [17], [48], making it the baseline of many relevant works, including ours. Readers interested in details about VINS-Mono may refer to the original works [18] for in-depth fundamentals and architecture; we will also present the remarkable work here based on the direct interpretation of the codebase[1].

In VINS-Mono, the system starts with measurement processing to extract and track features and pre-integrate IMU measurements as illustrated in Figure 2.8. The initialization step will bootstrap parameters necessary for subsequent nonlinear optimization based VIO. Lastly, the pose graph optimization will perform sliding window optimization to refine poses within the sliding window. Meanwhile, a loop-closure runs parallel to perform global trajectory optimization upon revisiting the same spot. VINS-Mono will produce two trajectories: the VIO real-time trajectory and an infrequent global trajectory upon each loop closure.

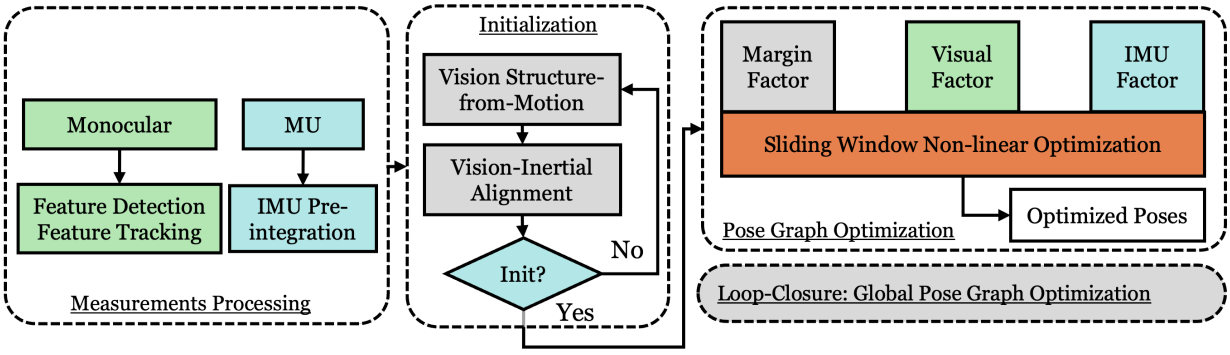


Figure 2.8: Simplified VINS-Mono block diagram illustration

In our works, we are interested in improving the run-time odometry (i.e., VIO) rather than global closure consistency. A more robust odometry will ensure a stable trajectory, resulting in a better global trajectory and a globally consistent map. Hence, in this section, we will review the literature on the building stones necessary for this thesis. Interested readers are encouraged to refer to the original paper in VINS-Mono [18]. As illustrated in

[1]To note, the original paper [18] may have discrepancies with the code, this section will present VINS-Mono in details on the restructured codebase to provide sufficient background to our work.

Figure 2.8, this section will cover the measurements processing stage (front-end), initialization, and pose graph optimization (back-end).

2.2.1 Front-End

VINS-Mono intakes two measurements at different rates: a monocular image feed at 30 [Hz] and IMU measurements at 100 [Hz]. The IMU measurements are continuously acquired until the subsequent image measurement is processed. The measurements processing pipeline is composed of two processes: the sparse visual process and the IMU pre-integration.

Sparse Visual processing

If the monocular image consists of multiple colour channels (e.g., RGB), it is often transformed into a grayscale image, as structural features are derived from pixel intensity rather than chromaticity. Subsequently, the image undergoes processing via a sparse (indirect) approach, involving feature extraction from the most recent frame and tracking between successive frames. As depicted in Figure 2.9, this procedure encompasses four distinct steps for each monocular frame acquisition:

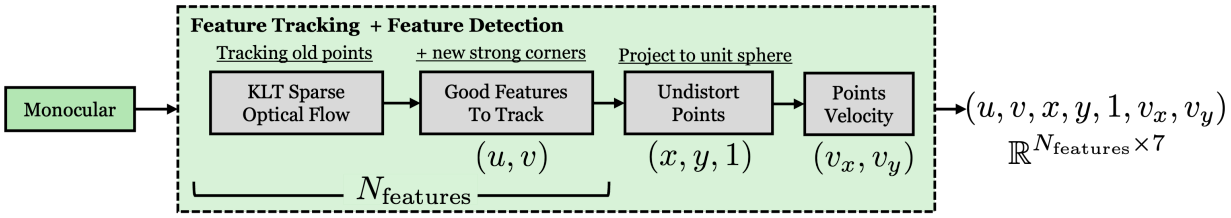


Figure 2.9: VINS-Mono Front-End: Monocular Visual Processing Pipeline

1. Existing features are tracked with Kanade-Lucas-Tomasi (KLT) sparse optical flow algorithm using '`calcOpticalFlowPyrLK`' from Open Computer Vision [10] (OpenCV)
2. Additional new corner features are identified to uphold total $N_{features} = 150$ features with '`goodFeaturesToTrack`' from OpenCV. This detector enforces a minimum separation of pixels between neighbouring features, maintaining a uniform distribution of features.
3. 2D features are undistorted by lifting from the image plane to its projective ray and then normalized onto a unit sphere.

4. Lastly, the velocity of the tracked normalized points is computed via first-order time derivative

In the process, keyframes are selected based on extracted features using specific criteria. These criteria entail identifying instances where the average parallax from the previous keyframe exceeds a predetermined threshold or detecting a decline in tracking quality with the tracked features falling below a specified threshold. Moreover, parallax is remedied through the incorporation of a short-term integration of gyroscope measurements to address parallax caused by pure rotation.

IMU Pre-integration

In VIO systems, IMU measurements, encompassing accelerations and angular velocities, are obtained at high frequencies, whereas visual measurements (images) are captured at lower frequencies. Pre-integrating a batch of IMU measurements between consecutive frames can condense the high-frequency data into a concise representation synchronized with the consecutive frames. Consequently, it mitigates the later optimization state complexity.

Let us consider an IMU measurements model, composed with a three-axis accelerometer $(.)_{\mathbf{a}}$ and a three-axis gyroscope $(.)_{\boldsymbol{\omega}}$ for measuring linear acceleration $\mathbf{a}_t \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$ respectively [49]. The measurements $\tilde{(.)}_t$ can be modelled in terms of the true measurement $(.)_t$, with an additional slowly varying sensor bias $\mathbf{b}$ and an additive white noise $\boldsymbol{\eta}$:

$$\begin{aligned} \tilde{\mathbf{a}}_t &= \mathbf{a}_t + \mathbf{b}_{\mathbf{a}_t} + \mathbf{R}^t_{\boldsymbol{\omega}}\, \mathbf{g}^{\boldsymbol{\omega}} + \mathbf{n}_{\mathbf{a}} \\ \tilde{\boldsymbol{\omega}}_t &= \boldsymbol{\omega}_t + \mathbf{b}_{\boldsymbol{\omega}_t} + \mathbf{n}_{\boldsymbol{\omega}} \end{aligned} \tag{2.19}$$

Specifically, acceleration and gyroscope biases $(\mathbf{b}_{\mathbf{a}}, \mathbf{b}_{\boldsymbol{\omega}})$ have their derivatives as white noises $(\mathbf{n}_{\dot{\mathbf{b}}_{\mathbf{a}}} \sim \mathcal{N}\left(0, \sigma^2_{\dot{\mathbf{b}}_{\mathbf{a}}}\right), \mathbf{n}_{\dot{\mathbf{b}}_{\boldsymbol{\omega}}} \sim \mathcal{N}\left(0, \sigma^2_{\dot{\mathbf{b}}_{\boldsymbol{\omega}}}\right))$ with additive noises $(\mathbf{n}_{\mathbf{a}} \sim \mathcal{N}\left(0, \sigma^2_{\mathbf{a}}\right), \mathbf{n}_{\boldsymbol{\omega}} \sim \mathcal{N}\left(0, \sigma^2_{\boldsymbol{\omega}}\right))$.

For two consecutive frames k and $k+1$, there exists several inertial measurements in time interval $[t_k, t_{k+1}]$. To note, the pre-integration derivation presented here is a quaternion-based IMU preintegration, which provides the same numerical results as the on-manifold method during Zero-Order-Hold (ZOH) discretization in [49].

$$
\begin{aligned}
\boldsymbol{\alpha}_{k+1}^{k} &= \int\int_{t\in[t_k,t_{k+1}]} \mathbf{R}_t^k(\tilde{\mathbf{a}}_t - \mathbf{b}_{\mathbf{a}_t})dt^2 \\
\boldsymbol{\beta}_{k+1}^{k} &= \int_{t\in[t_k,t_{k+1}]} \mathbf{R}_t^k(\tilde{\mathbf{a}}_t - \mathbf{b}_{\mathbf{a}_t})dt \\
\boldsymbol{\gamma}_{k+1}^{k} &= \int_{t\in[t_k,t_{k+1}]} \boldsymbol{\Omega}_t^k(\tilde{\boldsymbol{\omega}}_t - \mathbf{b}_{\boldsymbol{\omega}_t})\boldsymbol{\gamma}_t^{b_k}dt
\end{aligned}
\tag{2.20}
$$

where,

$$
\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -(\boldsymbol{\omega})^{\wedge} & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, \quad (\boldsymbol{\omega})^{\wedge} = \begin{bmatrix} 0 & -\boldsymbol{\omega}_z & \boldsymbol{\omega}_y \\ \boldsymbol{\omega}_z & 0 & -\boldsymbol{\omega}_x \\ -\boldsymbol{\omega}_y & \boldsymbol{\omega}_x & 0 \end{bmatrix} \tag{2.21}
$$

To note, $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ represents cumulative changes for pre-integration, regardless of the initial state. The product of rotation R_t^k still lies in the group manifold $\mathbf{SO}(3)$, representing a single group operation. However, all cumulate changes are dependent on bias estimates and noise estimates. Usually, the general assumption is that the noises are white noises, yielding zero in the nominal case. Bias is often assumed constant between two keyframes to avoid repetitive integration over the changing biases. Biases are compensated later.

By utilizing the first-order approximations with respect to the bias, adjustments to $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ can be made when there is a minor change in the bias estimation. Conversely, in the case of substantial changes, it is necessary to repropagate with the new bias estimation. This method allows for dynamic conservation of computational resources.

As a result, IMU measurements are pre-integrated into a single pose estimation that will be used in the next initialization and optimization step, along with the features captured at the moment.

2.2.2 Initialization

Estimator initialization is a key pre-requisite for an accurate and converging VIO, as monocular tightly coupled VIO is a highly nonlinear system. Consequently, the author proposed initialization through the loose coupling of the two sensors through two-step initialization: vision-only Structure From Motion (SfM) and visual-inertial alignment. Figure 2.10 illustrates a detailed flow chart about the process.

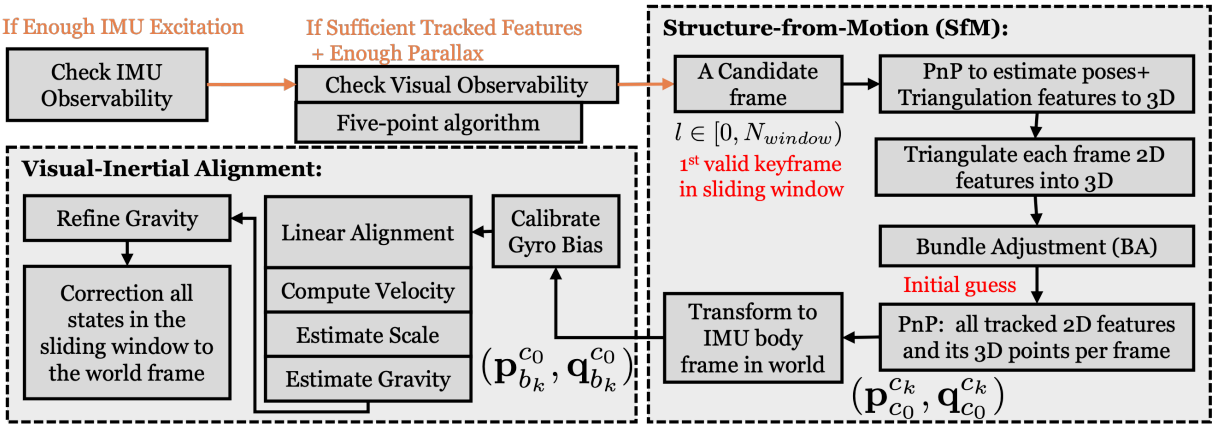


Figure 2.10: VINS-Mono Initialization: SfM and Visual-Inertial Alignment

Vision-Only SfM in Sliding Window

The initialization process commences with a vision-only SfM technique to realize a graph of up-to-scale camera poses and feature positions. The objective is to reconstruct the 3D environment from a series of 2D images. This is usually solved by formulating a BA problem, a nonlinear least squares problem composed typically of points and camera parameters (position, intrinsics), and solved by a nonlinear optimization solver like Ceres Solver (Ceres) via dense Cholesky factorization.

Initially, the world frame is unknown. Hence, the first camera frame $(\cdot)^{c_0}$ is assumed to be the reference frame for SfM. The initialization process starts with checking the observability of the IMU (whether the IMU is excited enough) and checking the correspondences between two consecutive frames $(k, k+1)$ (whether the stable feature tracking and parallax $(n_{feat}^{tracked} > 30, n_{pixels}^{parallax} > 20)$). The first keyframe set will be considered the candidate frame, and its relative rotation and up-to-scale translation will be recovered with the five-point algorithm [50].

For every two-frame combination, matched points will be used to compute the camera pose relative to the world frame via Perspective-n-Point (PnP) method (Given a set of n 3D points and their corresponding 2D projections in the image, one has to estimate the pose of a calibrated camera [50]). The estimated camera poses will then compute the 3D coordinates of additional points observed in the images (a.k.a., triangulation). In VINS-Mono, this process is performed in two passes: the last frame with all frames from the candidate frame and the candidate frame with all other frames. Lastly, all the estimated points are triangulated into 3D points.

Now, a global Bundle Adjustment (BA) can be formulated to optimize the camera's poses and 3D points simultaneously. This process minimizes the reprojection error (the difference between the observed and projected points) across all images and points.

Finally, the optimized poses and 3D points will be passed into PnP to conceive the final rotation and translation vectors that transform world points into the camera frame via OpenCV. After a simple inverse, we can obtain the camera poses w.r.t.the initial frame $(\cdot)^{c_0}$. As extrinsic parameters $(\mathbf{p}_c^b, \mathbf{q}_c^b)$ between the camera and IMU are known, all frame poses $(\mathbf{p}_{c_k}^{c_0}, \mathbf{q}_{c_k}^{c_0})$ can be transformed from the camera frame to the body (IMU) frame for the later alignment.

However, all the optimized translations from visual features are relative, and the true scale is to be determined. Let us assume there exists a scale factor (s) to correlate the translation deduced from the visual feature to map to the true scale in the real world:

$$\mathbf{p}_{c_0}^{b_k} = s\bar{\mathbf{p}}_{c_0}^{b_k} \quad \mathbf{p}_{c_0}^{c_k} = s\bar{\mathbf{p}}_{c_0}^{c_k} \tag{2.22}$$

Hence, the corrected transformation from the visual pose to the body pose is now:

$$\begin{aligned} \mathbf{q}_{b_k}^{c_0} &= \mathbf{q}_{c_k}^{c_0} \otimes (\mathbf{q}_c^b)^{-1} \\ s\,\bar{\mathbf{p}}_{c_0}^{b_k} &= s\,\bar{\mathbf{p}}_{c_0}^{c_k} - \boldsymbol{R}_{c_0}^{b_k}\mathbf{p}_c^b \end{aligned} \tag{2.23}$$

where, the scale factor (s) will be determined when aligning with the IMU measurements.

Visual-Inertial Alignment

Before performing graph optimization, the visual estimation would have to be aligned with the inertia measurements. First, the gyro bias will be corrected by optimizing the error in the gyro measurements from the visual features with respect to the IMU pre-integration.

As the IMU preintegration term is linearized with its first-order approximations. the gyro term can be approximated with a bias term:

$$\boldsymbol{\gamma}_{k+1}^k \approx \hat{\boldsymbol{\gamma}}_{k+1}^k \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{J}_{\mathbf{b}_\omega}^{\gamma}\delta\mathbf{b}_{\omega_k} \end{bmatrix} \tag{2.24}$$

We obtain the corrected bias $\mathbf{b}_\omega$ by minimizing the difference of the consecutive image measurements with the IMU pre-integrated gyro measurement $\hat{\boldsymbol{\gamma}}_{k+1}^k$:

$$\min_{\delta\mathbf{b}_\omega} \sum_{k\in\mathcal{Z}} \left\| (\mathbf{q}_{b_{k+1}}^{c_0})^{-1} \otimes \mathbf{q}_{b_k}^{c_0} \otimes \hat{\boldsymbol{\gamma}}_{k+1}^k \right\|^2 \tag{2.25}$$

Then, the initial states of velocity, gravity, and metric scale are packed into one optimization state vector and optimized by solving a linear least-square problem to minimize the alignment error between the visual measurements and IMU pre-integrations:

$$\begin{aligned} \min_{\mathbf{X}_I} \sum_{k \in \mathcal{Z}} \left\| \hat{\mathbf{z}}_{b_k+1}^{b_k} - \mathbf{H}_{b_k+1}^{b_k} \mathbf{X}_I \right\|^2 \\ \mathbf{X}_I = \left[\mathbf{v}_{b_k}^{b_k},\ \mathbf{v}_{b_{k+1}}^{b_{k+1}},\ \mathbf{g}^{c_0}, \mathbf{s} \right]^T \end{aligned} \tag{2.26}$$

Lastly, the gravity direction vector is further refined by fixing its magnitude with a known value and reoptimizing the eq. (2.26) with an adjusted formulation of $\hat{\mathbf{z}}_{b_k+1}^{b_k}$.

2.2.3 Back-End

The back end of VINS-Mono is also based on the formulation of visual-inertial BACeres:

$$\min_{\mathbf{X}} \left\{ \|\mathbf{r}_p - H_p \mathbf{X}\|^2 + \sum_{k \in \mathcal{K}} \sum_{\zeta_{l,j} \in \mathcal{C}_k} \rho \left(\|\mathbf{r}_{\mathcal{C}}(\zeta_{l,j}, \mathbf{X})\|^2_{P_{\zeta_{l,j}}} \right) + \sum_{k \in \mathcal{K}} \|\mathbf{r}_{\mathcal{Z}}(z_k, \mathbf{X})\|^2_{P_{z_k}} \right\} \tag{2.27}$$

where the full state vector in the sliding window is defined as:

$$\mathbf{X} = \left[\mathbf{x}_{b_0}, \ldots, \mathbf{x}_{b_n}, \mathbf{p}_c^b, \mathbf{q}_c^b, \lambda_0, \ldots, \lambda_m \right] \tag{2.28}$$

with the state at each frame is a vector

$$\mathbf{x}_{b_k} = \text{col} \left(\mathbf{p}_{b_k}^w, \mathbf{v}_{b_k}^w, \mathbf{q}_{b_k}^w, \mathbf{b}_a, \mathbf{b}_g \right), \quad k \in \mathcal{K} \tag{2.29}$$

where, $\mathbf{p}_{b_k}^w \in \mathbb{R}^3$ denotes position, $\mathbf{v}_{b_k}^w \in \mathbb{R}^3$ is velocity, $\mathbf{q}_{b_k}^w \in \mathbb{Q} \subset \mathbb{R}^4$ is quaternion for orientation, and $\mathbf{b}_a, \mathbf{b}_g \in \mathbb{R}^3$ denote biases of accelerometer and gyroscope, respectively.

The IMU residual ($\mathbf{r}_{\mathcal{Z}}$) is computed through the error between the relative state transformation ($\mathbf{x}_{b_k} \rightarrow \mathbf{x}_{b_{k+1}}$) over two consecutive frames and the predicted relative transformation ($z_k = \hat{z}_{b_{k+1}}^{b_k}$) through the pre-integration of IMU measurements. The visual residual ($\mathbf{r}_{\mathcal{C}}$) is related to the pixel error of the reprojected pixel $\zeta_{l,j} \in \mathcal{C}_k$ from the current frame w.r.t. the same feature in the next frame $\zeta_{l,j} \in \mathcal{C}_{k+1}$.

Residual of IMU

The IMU residual is computed through the error between the relative transformation $(\chi_k \rightarrow \chi_{k+1})$ between the consecutive frames and the predicted relative transformation $(\hat{\alpha}^{b_k}_{b_{k+1}}, \hat{\beta}^{b_k}_{b_{k+1}}, \hat{\gamma}^{b_k}_{b_{k+1}})$ from the pre-integration of IMU measurements.

$$\mathbf{r}_{\mathcal{Z}}(z_k, \mathbf{X}) = \begin{bmatrix} \delta\boldsymbol{\alpha}^{b_k}_{b_{k+1}} \\ \delta\boldsymbol{\beta}^{b_k}_{b_{k+1}} \\ \delta\boldsymbol{\theta}^{b_k}_{b_{k+1}} \\ \delta\mathbf{b}^a \\ \delta\mathbf{b}^g \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{b_k}_w \left(\mathbf{p}^w_{b_{k+1}} - \mathbf{p}^w_{b_k} + \frac{1}{2}\mathbf{g}^w \Delta t_k^2 - \mathbf{v}^w_{b_k} \right) - \hat{\boldsymbol{\alpha}}^{b_k}_{b_{k+1}} \\ \mathbf{R}^{b_k}_w \left(\mathbf{v}^w_{b_{k+1}} + \mathbf{g}^w \Delta t_k^2 - \mathbf{v}^w_{b_k} \right) - \hat{\boldsymbol{\beta}}^{b_k}_{b_{k+1}} \\ 2 \left[\mathbf{q}^{w\,-1}_{b_k} \otimes \mathbf{q}^w_{b_{k+1}} \otimes \hat{\boldsymbol{\gamma}}^{b_k}_{b_{k+1}} \right]_{xyz} \\ \mathbf{b}^{b_{k+1}}_a - \mathbf{b}^{b_k}_a \\ \mathbf{b}^{b_{k+1}}_g - \mathbf{b}^{b_k}_g \end{bmatrix} \tag{2.30}$$

To note, $\left(\hat{\boldsymbol{\alpha}}^{b_k}_{b_{k+1}}, \hat{\boldsymbol{\beta}}^{b_k}_{b_{k+1}}, \hat{\boldsymbol{\gamma}}^{b_k}_{b_{k+1}} \right)$ are IMU pre-integration, and equivalent to $(\Delta\mathbf{R}_{ij}, \Delta\mathbf{v}_{ij}, \Delta\mathbf{p}_{ij})$.

Residual of Camera Vision

The residual for a feature observation in one image compared to another is defined as the pixel re-projection error on a unit sphere. This approach accommodates the optical characteristics of various camera types [18].

$$\mathbf{r}_{\mathcal{C}}(\zeta_{l,j}, \mathbf{X}) = \begin{bmatrix} \hat{u}^{c_j}_l \\ \hat{v}^{c_j}_l \end{bmatrix} - \pi_c \left(\mathbf{R}^c_b \left(\mathbf{R}^{b_j}_w \left(\mathbf{R}^{w}_{b_i} \left(\mathbf{R}^b_c \frac{1}{\lambda_l} \pi_c^{-1} \left(\begin{bmatrix} u^{c_i}_l \\ v^{c_i}_l \end{bmatrix} \right) + \mathbf{p}^b_c \right) + \mathbf{p}^w_{b_i} - \mathbf{p}^w_{b_j} \right) - \mathbf{p}^b_c \right) \right) \tag{2.31}$$

2.3 Camera and IMU Calibration

2.3.1 Characterize IMU Noises with Allan Variance [1]

For an accurate VIO, we need to run a routine to know how "noisy" the IMU is. To obtain the IMU parameters for VINS-Mono, Allan Variance [1] (AVar) is a great toolbox to compute the measurement noise and bias random walk noise for both gyroscope and accelerometer in a standard 6-axis IMU. This tool is a full Robot Operating System (ROS) compatible tool used by many researchers in VIO community.

For our study, at least 3 hours of data are collected for every D455 camera. Both IMUs are configured at 200 [Hz] rate. In addition, both cameras are collected on a stationary table, and the robot is powered on at the nominal configuration. Specifically, the base camera is mounted in the front of the mobile base, while the EE camera is mounted at the EE of the WAM at the Home (H) position. The resultant standard deviations for sensor noise and random walk bias of accelerometer and gyroscope are tabulated in Table 2.2 [2]. From the table, we can observe significant noise is experienced at the EE. Similarly, the

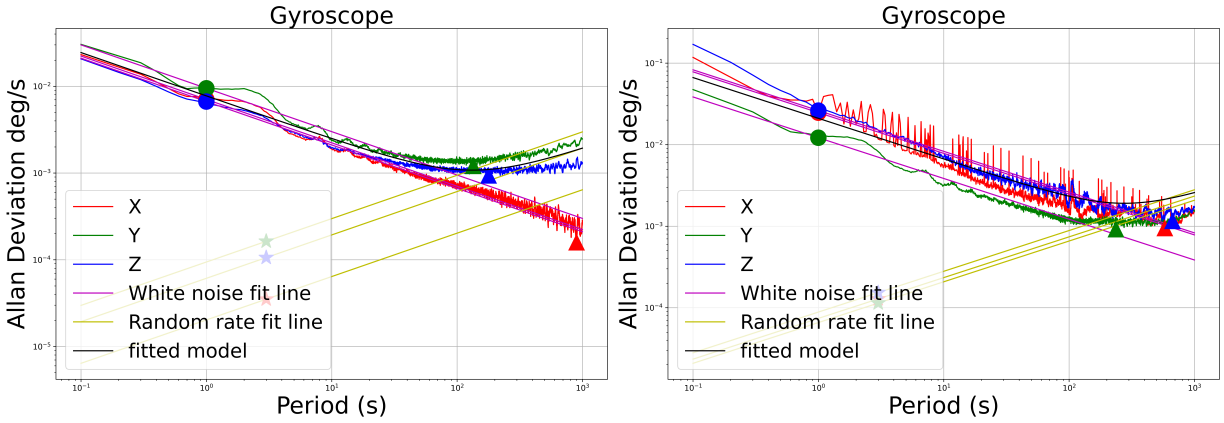
Table 2.2: Calibrated AVar Results for base (B) and End-Effector (E) D455 cameras with each measured at two different places: one on the table and the other at the base or EE.

	Accelerometer Noise $\sigma_{\mathbf{n_a}} \left[\frac{\text{m}}{\text{s}^2}\frac{1}{\sqrt{\text{Hz}}}\right]$	Accelerometer Random Walk $\sigma_{\dot{\mathbf{b}}_\mathbf{a}} \left[\frac{\text{m}}{\text{s}^2}\frac{1}{\sqrt{\text{Hz}}}\right]$	Gyroscope Noise $\sigma_{\mathbf{n}_{\boldsymbol{\omega}}} \left[\frac{\text{rad}}{\text{s}}\frac{1}{\sqrt{\text{Hz}}}\right]$	Gyroscope Random Walk $\sigma_{\dot{\mathbf{b}}_{\boldsymbol{\omega}}} \left[\frac{\text{rad}}{\text{s}^2}\frac{1}{\sqrt{\text{Hz}}}\right]$
D455(B) On Table	0.001244	8.041855e-05	0.000207	1.654800e-06
D455(B) At Base	**0.003946**	5.247260e-05	0.000162	1.954810e-06
D455(E) On Table	0.0010379	3.265699e-05	0.000166	2.848286e-06
D455(E) At EE	**0.0099982**	**0.001057**	**0.000457**	2.672308e-06

resultant figure for EE can be seen in Figure 2.11 and Figure 2.12 for the gyroscope and accelerometer, respectively.

This is further validated when we performed the VINS-Mono at the EE. The nominal noise parameters that are characterized on the stationary table quickly diverge, while the parameters characterized on a powered EE converge but have varying performance depending on the configuration of the WAM.

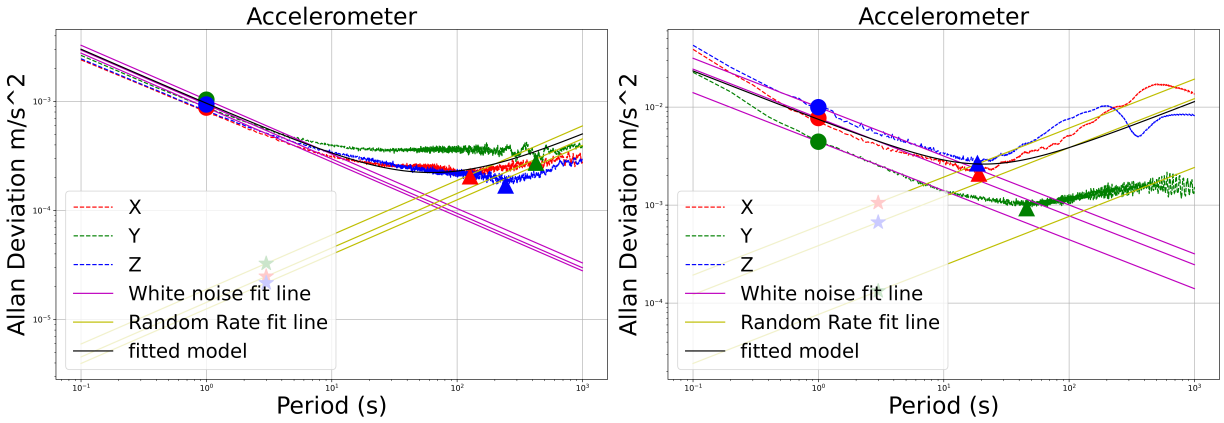
[2] To note, the camera is mapped with serial numbers to ensure consistency throughout the study



(a) D455 Gyro on a stationary table

(b) D455 Gyro on the EE at Home (H)

Figure 2.11: D455 EE IMU Gyroscope Characteristics via AVar



(a) D455 Acc on a stationary table

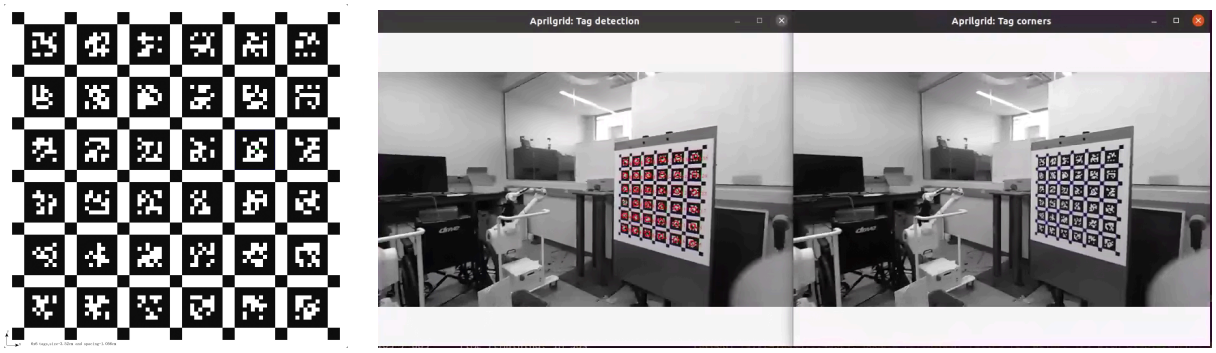
(b) D455 Acc on the EE at Home (H)

Figure 2.12: D455 EE IMU Accelerometer Characteristics via AVar

2.3.2 Camera and IMU Calibration with Kalibr

The Kalibr Toolbox [4] (Kalibr), developed by the Autonomous Systems Lab at ETH Zurich, offers solutions for rolling shutter camera calibration, multi-camera calibration, visual-inertial calibration (CAM-IMU), and multi-inertial calibration (IMU-IMU). The toolbox is widely utilized within the V-SLAM research community. In particular, we utilize the monocular camera calibration and visual-inertial calibration in the context of our experiment.

Our calibration process entails the collection of 30 to 60 seconds of data while physically manipulating the camera in hand, while ensuring that the calibration board remains within the camera's field of view throughout the procedure. The calibration board is magnetically attached to the steel sheet to ensure a consistent planar surface. The calibration board contains a 6×6 april tags forming an April grid for both tag and corner detection as illustrated in Figure 2.13.

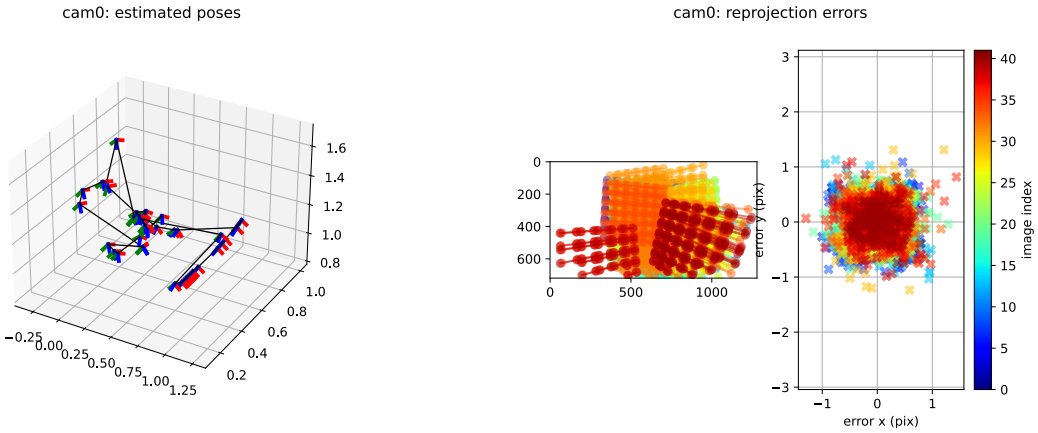


(a) April Grid (6x6) (b) Kalibr Calibration with Tag and Corner Detection

Figure 2.13: D455 Calibration with Kalibr

For monocular camera intrinsic calibration, we use an omnidirectional camera model ('`omni`') for the projection model and radial-tangential ('`radtan`') for the distortion models. The radial-tangential projection contains four parameters (k_1, k_2, p_1, p_2). The omnidirectional model consists of five parameters: a mirror parameter ξ, two for focal length (f_u, f_v), and two for principal points (p_u, p_v). This is the same model used in VINS-Mono for the '`MEI`' model, which is labelled using the last name of the first author who proposed the omnidirectional camera model [51]. Figure 2.14 illustrates the resultant intrinsic calibration result for the EE camera.

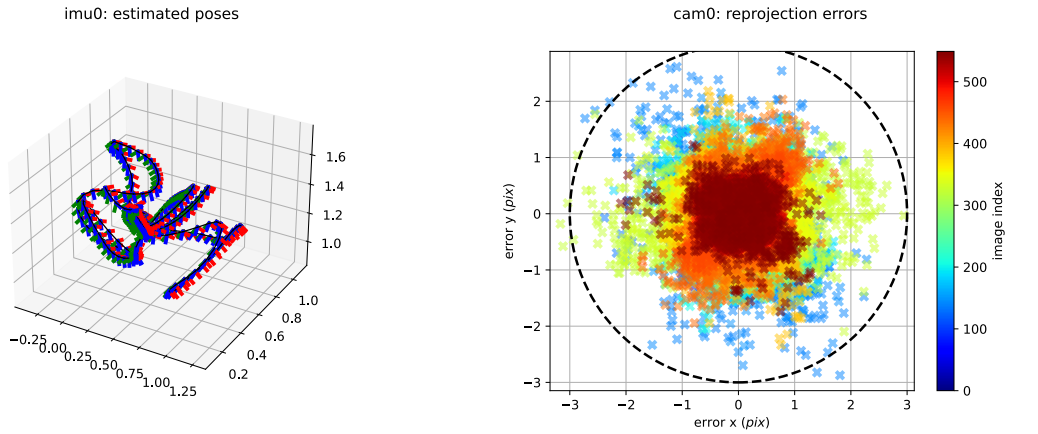
For visual-inertial extrinsic calibration, we use the IMU sensor characteristics captured



(a) Estimated Poses (b) Reprojection Errors

Figure 2.14: D455 EE Camera Model (MEI) Intrinsic Calibration with Kalibr

in the stationary scenario for both base and EE cameras. The timeshift and extrinsic for both cameras are being used as the extrinsic parameters for each of the cameras separately. To note, the default extrinsic parameters in Kalibr is T_{imu}^{cam}, whereas the VINS-Mono intakes the inverted transformation T_{cam}^{imu}. Similarly, Figure 2.15 illustrates the resultant extrinsic calibration result for the EE camera.



(a) Estimated Poses (b) Reprojection Error

Figure 2.15: D455 EE Camera Visual-IMU Extrinsic Calibration with Kalibr

2.4 Preliminary Concepts for Manipulator Kinematics

2.4.1 Manifold

A manifold is simply a generalization of curves and surfaces from higher dimensions that locally resembles Euclidean space. The collection (atlas) of such local Euclidean coordinates (charts) that cover the manifold defines the structure of the manifold. These charts allow one to carry out computations to the Manifold in high dimensional space as though in a Euclidean space $\mathbb{R}^n$ such as differentiability and tangent spaces [52].

2.4.2 Lie Groups and Lie Algebra

A Lie group ($\mathcal{G}$) is a particular smooth group, where its group operations $(g, h) \mapsto gh$ and $g \mapsto g^{-1}$ are smooth. For every lie group, there exists a respective Lie algebra $\mathfrak{g}$, which is a real vector space with an alternating bilinear map called the Lie bracket $\mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$, that satisfies the Jacobi identity. The Lie algebra for a Lie group provides a "linear model" of the group [53], providing a backbone for the computation.

Let's define some special groups:

- The General Linear Group ($\mathbf{GL}(n, \mathbb{R}) = \mathbb{R}^{n\times n}$) is the group of all $n \times n$ nonsingular real matrices.
- The Special Linear Group ($\mathbf{SL}(n, \mathbb{R})$) is the subgroup of $\mathbf{GL}(n, \mathbb{R})$:

$$\mathbf{SL}(n) = \{\mathbf{R} \in \mathbf{GL}(n, \mathbb{R}) \,|\, \det\{\mathbf{R}\} = 1\} \tag{2.32}$$

- The Special Orthogonal Group ($\mathbf{SO}(n)$) is the subgroup of $\mathbf{SL}(n, \mathbb{R})$:

$$\begin{aligned} \mathbf{SO}(n) &= \left\{\mathbf{R} \in \mathbf{SL}(n, \mathbb{R}) \,|\, \mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}\right\} \\ &= \left\{\mathbf{R} \in \mathbf{GL}(n, \mathbb{R}) \,|\, \mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det\{\mathbf{R}\} = 1\right\} \end{aligned} \tag{2.33}$$

- The Special Euclidean Group ($\mathbf{SE}(3)$) is a subgroup $\mathbf{SE}(3) \subseteq \mathbf{GL}(4, \mathbb{R})$:

$$\mathbf{SE}(3) = \left\{ \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbf{GL}(4, \mathbb{R}) \,|\, \mathbf{R} \in \mathbf{SO}(3), \mathbf{p} \in \mathbb{R}^3 \right\} = \mathbb{R}^3 \times \mathbf{SO}(3) \tag{2.34}$$

Intuitively, one may consider a Lie group (e.g.$\mathbf{SE}(3)$) is a particular smooth structure living in the high dimensional space (e.g.$\mathbf{SE}(3) \in \mathbb{R}^{4\times4}$), and its corresponding Lie algebra encodes such high dimensional structure with the minimal representation (e.g.$\mathfrak{se}(3) \in \mathbb{R}^6$).

In general, Lie groups play a crucial role in the fields of vision and robotics, representing everything from camera poses to the configuration of a robotic manipulator's end-effector. A Lie group is a manifold with a smooth group operation defined on it. Using Lie group theory and the corresponding Lie algebra, we can develop problem formulations and models that utilize smooth geometry, which is differentiable everywhere, while avoiding over-parameterization of the representation.

2.4.3 Lie groups and Lie algebra in $\mathbf{SO}(3)$

The particular orthogonal group $\mathbf{SO}(3)$ describes the group of 3-D rotation matrices, and it forms a smooth manifold, $\mathbf{SO}(3) = \left\{\mathbf{R} \in \mathbb{R}^{3\times3} \,|\, \mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det\{\mathbf{R}\} = 1\right\}$. Uniquely, its inverse is the matrix transpose, $\mathbf{R}^{-1} = \mathbf{R}^T$.

Its tangent space is the lie algebra denoted as $\mathfrak{so}(3)$ and coincides with space of 3×3 skew-symmetric matrices. The lie algebra is often described in a concise $\mathbb{R}^3$ vector, often known as its chart coordinate.

Vee and Hat operators

The vee-operator $(.)^\vee : \mathfrak{so}(3) \to \mathbb{R}^3$ and the hat-operator $(.)^\wedge : \mathbb{R}^3 \to \mathfrak{so}(3)$ [3] conveniently maps between the lie algebra and its coordinates:

$$\begin{aligned}(\boldsymbol{\phi})^\wedge &= \left(\begin{bmatrix}\boldsymbol{\phi}_1\\ \boldsymbol{\phi}_2\\ \boldsymbol{\phi}_3\end{bmatrix}\right)^\wedge = \begin{bmatrix}0 & -\boldsymbol{\phi}_3 & \boldsymbol{\phi}_2\\ \boldsymbol{\phi}_3 & 0 & -\boldsymbol{\phi}_1\\ -\boldsymbol{\phi}_2 & \boldsymbol{\phi}_1 & 0\end{bmatrix}\\ \left((\boldsymbol{\phi})^\wedge\right)^\vee &= \left(\begin{bmatrix}0 & -\boldsymbol{\phi}_3 & \boldsymbol{\phi}_2\\ \boldsymbol{\phi}_3 & 0 & -\boldsymbol{\phi}_1\\ -\boldsymbol{\phi}_2 & \boldsymbol{\phi}_1 & 0\end{bmatrix}\right)^\vee = \begin{bmatrix}\boldsymbol{\phi}_1\\ \boldsymbol{\phi}_2\\ \boldsymbol{\phi}_3\end{bmatrix} = \boldsymbol{\phi}\end{aligned} \tag{2.35}$$

Geometrically, the $\boldsymbol{\phi} \in \mathbb{R}^3$ vector represents the direction of the rotation axis, and its magnitude $\|\boldsymbol{\phi}\| \in \mathbb{R}$ represents the rotational magnitude about that axis $\frac{\boldsymbol{\phi}}{\|\boldsymbol{\phi}\|}$. Naturally,

[3] To note, in some literatures, $\widehat{\star}$ is oftenly used as a compact form, but may be confused with $\hat{\star}$ for unit vectors and $\widehat{\star}$ for estimators; instead, we will be explicitly representing the hat notation as $(\xi)^\wedge$.

an axis-angle representation $(\boldsymbol{\omega}, \theta)$ arises with a unit axis of rotation $\boldsymbol{\omega} \in \{\mathbb{R}^3 \,|\, \|\boldsymbol{\omega}\| = 1\}$ and a rotational magnitude $\theta \in \mathbb{R}$.

$$(\boldsymbol{\phi})^\wedge = \theta\ (\boldsymbol{\omega})^\wedge \iff \theta = \|\boldsymbol{\phi}\|,\ \boldsymbol{\omega} = \frac{\boldsymbol{\phi}}{\|\boldsymbol{\phi}\|} \tag{2.36}$$

Hence, $\boldsymbol{\phi} = \theta\boldsymbol{\omega}$. Like some literatures as [54], we often see representations in alternative axis-angle form $(\boldsymbol{\omega}, \theta)|_{\|\boldsymbol{\omega}\|=1}$ instead of a compact form $\boldsymbol{\phi} \in \mathbb{R}^3$.

In practice, $(\boldsymbol{\omega}, \theta)|_{\|\boldsymbol{\omega}\|=1}$ not only provides a more intuitive representation but also provides computation advantages in terms of speed and accuracy by eliminating the repetitive normalization division and floating quantization errors for small angles. As stated in 2.39, an axis-angle representation preserves small angle properties much more effectively while preserving the full resolution of the axis.

Property of skew-symmetric matrices

$$(\mathbf{a})^\wedge \mathbf{b} = -\,(\mathbf{b})^\wedge \mathbf{a} \qquad \forall \mathbf{a}, \mathbf{b} \in \mathbb{R}^3 \tag{2.37}$$

Exponential Mapping in $\mathbf{SO}(3)$

To map from the lie algebra $(\boldsymbol{\phi})^\wedge$ to its manifold $\boldsymbol{R} \in \mathbf{SO}(3)$, at the identity. This exponential mapping $\exp_{\mathbf{SO}(3)}(.) : \mathfrak{so}(3) \to \mathbf{SO}(3)$ is obtained through the simplified Rodrigues' rotational formula with an axis-angle representation $(\boldsymbol{\omega}, \theta)|_{\|\boldsymbol{\omega}\|=1}$ [54]:

$$\begin{aligned} R =&\ \exp_{\mathbf{SO}(3)}\left(\theta\,(\boldsymbol{\omega})^\wedge\right) \\ &= \mathbf{I}_3 + (\boldsymbol{\omega})^\wedge \sin(\theta) + ((\boldsymbol{\omega})^\wedge)^2\,(1 - \cos(\theta)) \\ &\left(\left\|(\boldsymbol{\omega})^\wedge\right\| = 1\right) \end{aligned} \tag{2.38}$$

For a small angle ($\theta \approx 0$) assumption, a first-order approximation at its identity [54] can be obtained:

$$\exp_{\mathbf{SO}(3)}\left(\theta\,(\boldsymbol{\omega})^\wedge\right) \approx \mathbf{I}_3 + (\boldsymbol{\omega})^\wedge \tag{2.39}$$

2.4.4 Lie groups and Lie algebra in $\mathbf{SE}(3)$

Let $\mathcal{G} \subset \mathbf{SE}(3)$ be a special Euclidean subgroup of a generic matrix Lie group and $\mathfrak{g} \subset \mathfrak{se}(3)$ its corresponding Lie algebra. Consider a dynamical system evolving on $\mathcal{G}$ with a state

realization χ such that

$$\dot{\chi} = \chi\xi^b = \chi\xi^s \quad \xi^b, \xi^s \in \mathfrak{g} \tag{2.40}$$

where ξ^b is the twist in the body frame and ξ^s in the spatial frame. The system $\chi\xi^b$ is left invariant and $\chi\xi^s$ is right invariant. The hat-map is an isomorphism that maps an element in the vector space $\mathbb{R}^n$ to the tangent space of the matrix Lie group at the identity [55]:

$$(.)^\wedge : \mathbb{R}^n \to \mathfrak{g} \mid \xi \mapsto (\boldsymbol{\xi})^\wedge \implies \mathbb{R}^6 \to \mathfrak{se}(3),\ \mathbb{R}^3 \to \mathfrak{so}(3) \tag{2.41}$$

and its inverse is the vee-map

$$(.)^\vee : \mathfrak{g} \to \mathbb{R}^n \mid (\boldsymbol{\xi})^\wedge \mapsto \xi \implies \mathfrak{se}(3) \to \mathbb{R}^6,\ \mathfrak{so}(3) \to \mathbb{R}^3 \tag{2.42}$$

For any $\phi \in \mathbb{R}^n$, the Lie exponential map is defined as

$$\exp(.) : \mathbb{R}^n \to \mathcal{G}, \quad \exp(\phi) = \exp_m((\phi)^\wedge) \tag{2.43}$$

where $\exp_m(.)$ is the exponential of square matrices.

2.4.5 3-Sphere Group $\mathbf{S}^3$ and Unit Quaternion $\mathbb{Q}$

The group $\mathbf{S}^3$ is the Lie group of unit quaternions $\mathbb{Q}$ under quaternion multiplication, with its topology of a quaternion as the unit 3-sphere in $\mathbb{R}^4$ [56].

$$\begin{aligned} \mathbf{q} &= w + \mathbf{i}x + \mathbf{j}y + \mathbf{k}z \quad \in \mathbb{Q} \\ &= \big(w, x, y, z\big) \end{aligned} \tag{2.44}$$

Quaternion is double covered with the rotation group $\mathbf{SO}(3)$, meaning that there are two corresponding quaternions q and $-q$ representing the same rotation for each rotation in $\mathbf{SO}(3)$. This redundancy can lead to problems in numerical algorithms involving optimization or iterative convergence. This implies the convergence of the pose optimization in quaternion may result in either of two opposing discontinuous points on $\mathbf{S}^3$, resulting in oscillatory numerical convergence and numerical instability [57].

2.5 Mathematical Foundation for Manipulator Kinematics

2.5.1 Translational Motion in Euclidean space

The formal definition for a rigid body ($\mathcal{O} \subset \mathbb{R}^3$) is a collection of particles ($p_i \in \mathcal{O}$) such that the distance between any two particles remains fixed ($\|p_i(t) - p_j(t)\| = \|p_i(0) - p_j(0)\| \, \forall t \in \mathbb{Z}^+$) regardless of any motions of the body or forces exerted on the body.

A rigid motion is defined as a continuous movement of the particles in the object such that the distance between any two particles remains fixed at all times. The motion can often described a parameterized curve:

$$p(t) = (x(t), y(t), z(t)) \in \mathbb{R}^3 \tag{2.45}$$

A rigid displacement is the net movement of a rigid body from one point to another:

$$d = \|p(t) - p(0)\|_2 \tag{2.46}$$

A rigid body transformation $g : \mathbb{R}^3 \to \mathbb{R}^3$ is a mapping that preserves length and coordinates space (cross-product):

$$\begin{aligned} \text{Length:} \quad & \|g(p) - g(q)\| = \|p - q\|, \, \forall p, q \in \mathbb{R}^3 \\ \text{Cross-product:} \quad & g_\star(\mathbf{v} \times \mathbf{w}) = g_\star(\mathbf{v}) \times g_\star(\mathbf{w}), \, \forall \mathbf{v}, \mathbf{w} \in \mathbb{R}^3 \end{aligned} \tag{2.47}$$

2.5.2 Rotational Motion in Different Representation

The body's orientation can be identified by the relative orientation difference between a coordinate frame attached to the body (the body frame B) and a fixed or inertial coordinate frame (A). Consider right-handed conventions for all coordinate frames, let the base frame A be represented with its principal axes $R_A = \begin{bmatrix} x & y & z \end{bmatrix} = \mathbf{I}$ at the identity. The corresponding axes of the principal axes of B w.r.t., the base frame A, can be defined as:

$$R_{AB} = \begin{bmatrix} \mathbf{x}_{AB} & \mathbf{y}_{AB} & \mathbf{z}_{AB} \end{bmatrix} \in \mathbb{R}^{3\times3} \tag{2.48}$$

The resultant (2.48) is the basis of the rotation matrix [54], with two key properties: the columns are mutually orthonormal, and the determinant is $+1$ for the right-handed frame.

Rotation Matrix and $\mathbf{SO}(3)$

Through definition by construction in (2.48), we may define the space of the rotation matrices with $\mathbb{R}^{n\times n}$ in any $\mathbb{R}^n$ as the special orthogonal group as defined by (2.33). In $\mathbb{R}^3$, the corresponding rotation matrix is in the $R \in \mathbf{SO}(3)$ group.

The rotation matrix can be considered as the orientation component of the pose, but also a rigid body transformation. Considering the point $\mathbf{q}_A \in R_A$ in the frame A, and its corresponding projections in the frame B can be described as:

$$\mathbf{q}_B = \begin{bmatrix}\mathbf{x}_{AB} & \mathbf{y}_{AB} & \mathbf{z}_{AB}\end{bmatrix} \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = R_{AB}\,\mathbf{q}_A \tag{2.49}$$

, with properties that preserve distance and orientation as (2.47).

Conveniently, we can utilize any group theory in $\mathbf{SO}(3)$ to the general rotation matrix, for instance, its axioms [54]:

$$\begin{aligned} \text{Closure:} \quad & R_{AC} = R_{AB}\,R_{BC} \in \mathbf{SO}(3) \\ \text{Identity:} \quad & R_{AB}\,\mathbf{I} = R_{AB},\ \forall R_{AB} \in \mathbf{SO}(3) \\ \text{Inverse:} \quad & R_{AB}^{-1} = R_{AB}^{T} \quad , R_{AB}\,R_{AB}^{-1} = R_{AB}\,R_{AB}^{T} = \mathbf{I} \\ \text{Associativity:} \quad & R_{AB}\,(R_{BC}\,R_{CD}) = (R_{AB}\,R_{BC})\,R_{CD} \end{aligned} \tag{2.50}$$

Exponential Coordinates and $\mathfrak{so}(3)$

One can also consider a rotation to be represented by a point $\mathbf{q} \in \mathbb{R}^3$ rotating about its principal axis (as a unit vector $\hat{\boldsymbol{\omega}} \in \mathbb{R}^3$) with a specific angle ($t = \theta \in \mathbb{R}$ in radians). Its tip velocity at the point q, can be written as [54]:

$$\dot{\mathbf{q}}(t) = (\hat{\boldsymbol{\omega}})^{\wedge}\,\mathbf{q}(t) = \hat{\boldsymbol{\omega}} \times \mathbf{q}(t) \tag{2.51}$$

By integrating the time-invariant linear differential equation, we may obtain:

$$\mathbf{q}(t) = e^{(\hat{\boldsymbol{\omega}})^{\wedge} t}\mathbf{q}(0) \tag{2.52}$$

Considering $t = 0$ to be the initial configuration of the point and $t = \theta$ to be the terminal configuration of the point, the resultant net rotation can be described as the exponential coordinates:

$$\mathbf{R}(\hat{\boldsymbol{\omega}}, \theta) = e^{(\hat{\boldsymbol{\omega}})^{\wedge}\theta} \tag{2.53}$$

This is considered the canonical coordinates of the rotation group, and the equivalent linearized approximation can be expressed with the Rodrigues' formula (2.38). In some literature and libraries, a general vector representation is used for compactness: $\boldsymbol{\omega} = \hat{\boldsymbol{\omega}}\theta$. The resultant skew-symmetric matrix can be defined as $(\boldsymbol{\omega})^\wedge = (\hat{\boldsymbol{\omega}}\theta)^\wedge \in \mathfrak{so}(3)$.

An explicit axis-angle representation $(\boldsymbol{\omega}, \theta)$ is more intuitive (with physical meanings) to represent revolute joints and mathematical advantages during the computation (especially in small angle representation). The corresponding lie algebra provides mathematical leverage by directly incorporating the Lie group structure.

Euler angles

Another common method of describing the orientation is through quantifying roll, pitch, and yaw, which are rotation angles about the x-axis, y-axis, and z-axis correspondingly. This is often called XYZ Euler angles, and there are many other conventions of the Euler angles. Despite its intuitive representation as used in aviation, it is a fundamentally flawed representation as it has singularities.

$$R_{AB} = R_{\mathbf{x}}(\phi) R_{\mathbf{y}}(\theta) R_{\mathbf{z}}(\psi) = e^{\hat{\mathbf{x}}\phi} e^{\hat{\mathbf{y}}\theta} e^{\hat{\mathbf{z}}\psi} \tag{2.54}$$

Quaternion

Similar to the $\mathbf{SO}(3)$ group, quaternions (as defined in (2.44)) form a non-commutative group under multiplication, which satisfies the group axioms:

$$\begin{aligned}
\text{Closure:} \quad & \mathbf{q}_{AC} = \mathbf{q}_{AB} \otimes \mathbf{q}_{BC} \in \mathbb{Q} \\
\text{Identity:} \quad & \mathbf{q}_{AB} \otimes \mathbf{q}_{\mathbf{I}} = \mathbf{q}_{AB},\ \forall \mathbf{q}_{AB} \in \mathbb{Q},\ \mathbf{q}_{\mathbf{I}} = \big(1, 0, 0, 0\big) \\
\text{Inverse:} \quad & \mathbf{q}_{AB}^{-1} = \frac{\bar{\mathbf{q}}_{AB}}{\|\mathbf{q}_{AB}\|_2} \quad , \mathbf{q}_{AB} \otimes \mathbf{q}_{AB}^{-1} = \big(1, 0, 0, 0\big) \\
\text{Associativity:} \quad & \mathbf{q}_{AB} \otimes (\mathbf{q}_{BC} \otimes \mathbf{q}_{CD}) = (\mathbf{q}_{AB} \otimes \mathbf{q}_{BC}) \otimes \mathbf{q}_{CD}
\end{aligned} \tag{2.55}$$

Quaternions generalize complex numbers to provide a global parameterization of $\mathbf{SO}(3)$, which does not suffer from singularities, with a compact and efficient representation for rotations. Given (2.53), the associated unit quaternion is:

$$\mathbf{q}_{AB} = (\cos(\theta/2), \boldsymbol{\omega} \sin(\theta/2)) \tag{2.56}$$

2.5.3 Rigid Body Motion in SE(3)

A rigid motion preserves the distance between points and the angle between vectors. We can model the rigid motion with a rigid body transformation for an instantaneous frame w.r.t.an inertial frame. Hence, it consists of a rotation $R_{AB}(t) \in \mathbf{SO}(3)$ and translation $\mathbf{p}_B^A(t) \in \mathbb{R}^3$ at time t with respect to inertial frame A. This transformation $(R_{AB}, \mathbf{p}_B^A) : \mathbb{R}^3 \mapsto \mathbb{R}^3$ can be represented in terms of the configuration of a rigid body as an element $\chi_B^A \in \mathbf{SE}(3)$ or as the exponential of twists:

$$\bar{\mathbf{q}}^A = \begin{bmatrix} \mathbf{q}^A \\ 1 \end{bmatrix} = \chi_B^A(\theta) \begin{bmatrix} \mathbf{q}^B \\ 1 \end{bmatrix} = \exp\big((\boldsymbol{\xi})^\wedge \theta\big) \begin{bmatrix} \mathbf{q}^B \\ 1 \end{bmatrix} = \begin{bmatrix} R_{AB}(\theta) & \mathbf{p}_B^A(\theta) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{q}^B \\ 1 \end{bmatrix} \tag{2.57}$$

where, $t \equiv \theta \in \mathbb{R}$ is the angle of rotation, $\mathbf{q} \in \mathbb{R}^3$ is the rigid point in the Euclidean space, and $\bar{\mathbf{q}} \in \mathbb{R}^4$ is the homogeneous representation of $\mathbf{q} \in \mathbb{R}^3$.

Exponential Coordinates in SE(3)

Definition 1 *The exponential map* $\{\exp(.) : \mathfrak{g} \to \mathcal{G}\}$ *wraps the tangent element around the shorts path of the Lie group converting elements of its Lie algebra into those the Lie group. Within the realm of* $\mathfrak{se}(3)$ *and its proper subgroups, it is worth noting that the exponential map exhibits both surjective mapping and local diffeomorphism properties [58].*

Property 1 *Let* $\|.\|$ *be an* ***l2-norm*** *operator and given that* $(\boldsymbol{\omega})^\wedge \in \mathfrak{so}(3)$, $\omega, v \in \mathbb{R}^3$ *and* $(\boldsymbol{\xi})^\wedge = ((\boldsymbol{\omega})^\wedge, v) \in \mathfrak{se}(3)$. *Then, the corresponding* $\chi \in \boldsymbol{SE}(3)$ *can be expressed in terms of an exponential map (*$\exp_{\boldsymbol{SE}(3)}(.) : \mathfrak{se}(3) \to \boldsymbol{SE}(3)$*) [54]:*

$$\chi((\boldsymbol{\xi})^\wedge) = \exp_{\boldsymbol{SE}(3)}((\boldsymbol{\xi})^\wedge) = \begin{bmatrix} \exp_{\boldsymbol{SO}(3)}((\boldsymbol{\omega})^\wedge) & H(\omega)v \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \tag{2.58}$$

where $R \in \boldsymbol{SO}(3) := \exp_{\boldsymbol{SO}(3)}((\boldsymbol{\omega})^\wedge)$ *is the rotation* R *component, with its exponential mapping* $\exp_{\boldsymbol{SO}(3)}(.) : \mathfrak{so}(3) \to \boldsymbol{SO}(3)$ *obtained through the Rodrigues' rotational formula:*

$$\begin{aligned} R &= \exp_{\boldsymbol{SO}(3)}((\boldsymbol{\omega})^\wedge) \\ &= \mathbf{I}_3 + \sin(\|\omega\|)\frac{(\boldsymbol{\omega})^\wedge}{\|\omega\|} + (1 - \cos(\|\omega\|))\left(\frac{(\boldsymbol{\omega})^\wedge}{\|\omega\|}\right)^2, \omega \neq 0 \end{aligned} \tag{2.59}$$

and $H(\omega) \in \{\mathbb{R}^{3\times3} \,|\, \det\{H(\omega)\} = 1\} : v \in \mathbb{R}^3 \mapsto p \in \mathbb{R}^3$ *is the transnational mapping expressed as:*

$$H(\omega) = \frac{1}{\|\omega\|^2}\left((\mathbf{I}_3 - R)(\boldsymbol{\omega})^\wedge + \omega\omega^T\right), \quad \omega \neq 0 \tag{2.60}$$

To note, $\|\omega\| = \sqrt{\omega_1 + \omega_2 + \omega_3}$.

The exponential coordinate in an open set of the Lie algebra structure of the origin of $\mathcal{G}$ is denoted by $\xi = \log(\chi) \in \mathfrak{g}$. The unwrapping operation is the logarithmic map $\log(.) : \mathcal{G} \to \mathfrak{g}$.

2.5.4 Error Function(s) on SE(3)

The configuration error can alternatively be defined as a scalar-valued function of the form [59]

$$\begin{aligned} \Psi_t &= \frac{1}{2}\left\|I - \chi_{r,t}^{-1}(q)\chi_t(q)\right\|_F^2 = \frac{1}{2}\|I - \Phi_t\|_F^2 \\ \Phi_t &= \chi_{r,t}^{-1}\chi_t \in \mathcal{G} \end{aligned} \tag{2.61}$$

whose reference frame is the state of the system. Equation (2.61) can be decomposed into its rotational and translation components to obtain

$$\Psi_t = \frac{1}{2}\,\mathrm{Tr}\begin{bmatrix} 2I - \mathbf{R}^T\mathbf{R}_r - \mathbf{R}_r^T\mathbf{R} & * \\ * & I - \mathbf{R}_r^T(p - p_r) \end{bmatrix} \tag{2.62}$$

To note, eq. (2.61) is a scalar error metric in $\mathbf{SE}(3)$ that is differentiable and invertible. This metric is commonly used in Model Predictive Control (MPC) formulation and evaluation metric in performance evaluation. And its inverse, $\Psi_t^{-1} := \frac{1}{2}\left\|I - \chi_t^{-1}\chi_{r,t}\right\|_F^2$ is equivalent to that of Ψ_t [60].

2.5.5 Twists Representation for a Revolute Joint

Consider a revolute joint with an axis of rotation $\omega \in \mathbb{R}^3, \|\omega\| = 1$, and a point on the axis $q \in \mathbb{R}^3$, the tip velocity of a point $p(t)$ rotating about this axis is $\dot{p}(t) = \omega \times (p(t) - q)$.

As the tangent velocity is $v = -\omega \times q$, the homogeneous form for the tip velocity can be expressed in terms of a twist:

$$\dot{\bar{p}} = (\boldsymbol{\xi})^\wedge \bar{p}, \quad \dot{\bar{p}} = \begin{bmatrix} \dot{p} \\ 0 \end{bmatrix}, \; \bar{p} = \begin{bmatrix} p \\ 1 \end{bmatrix}, \quad (\boldsymbol{\xi})^\wedge = \begin{bmatrix} (\boldsymbol{\omega})^\wedge & v \\ 0 & 0 \end{bmatrix} \tag{2.63}$$

where, a twist $(\boldsymbol{\xi})^\wedge \in \mathfrak{se}(3) \subset \mathbb{R}^{4\times4}$ is defined as an element of $\mathfrak{se}(3)$, or an infinitesimal generator of the Euclidean group.

Thus, $\xi := (v, \omega) \in \mathbb{R}^6$ represents the twist coordinates for the twist $(\boldsymbol{\xi})^\wedge \in \mathfrak{se}(3) \subset \mathbb{R}^{4\times4}$, and is mapped with:

$$\xi = \left(\begin{bmatrix}(\boldsymbol{\omega})^\wedge & v\\ 0 & 0\end{bmatrix}\right)^\vee = \begin{bmatrix}v\\ \omega\end{bmatrix}, \quad (\boldsymbol{\xi})^\wedge = \left(\begin{bmatrix}v\\ \omega\end{bmatrix}\right)^\wedge = \begin{bmatrix}(\boldsymbol{\omega})^\wedge & v\\ 0 & 0\end{bmatrix} \tag{2.64}$$

2.5.6 WAM Forward Kinematics Models

To note, the notation $(.)^s$ and $(.)^b$ represent spatial and body frame, while $(.)_{st}$ represents from start to tool frame in a manipulator. Based on WAM frame, we can simplify the kinematics w.r.t.zero calibration configuration $g_{st}(0)$ (where the arm is straightened upwards). Every revolute joint can be defined as $\theta_i \in [0, 2\pi) \equiv \mathbb{S}^1$.

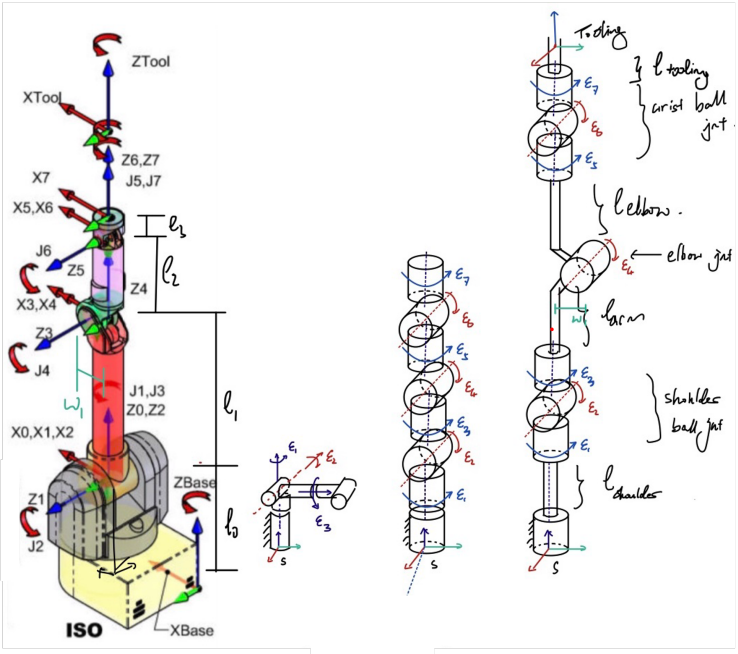


Figure 2.16: WAM Arm From Barrett Inc. (WAM) Kinematics Model

We can model the WAM ($\overbrace{\mathbb{S}^1 \times \cdots \times \mathbb{S}^1}^{7} \equiv \mathbb{T}^7$, a.k.a.7-Torus) in twist coordinates $\xi_i \in \mathbb{R}^6$ for each revolute joint. The initial transformation ($g_{st}(\theta = 0)$) is the zero-calibration configuration between the EE and the base frame. The Forward Kinematics (FK) **s.t.**g_{st} :

$\mathbb{T}^7 \to \mathbf{SE}(3)$ can then be defined with the product of the exponentials formula:

$$g_{st}(\theta) = e^{(\boldsymbol{\xi}_1)^\wedge \theta_1} e^{(\boldsymbol{\xi}_2)^\wedge \theta_2} e^{(\boldsymbol{\xi}_3)^\wedge \theta_3} \dots e^{(\boldsymbol{\xi}_7)^\wedge \theta_7} g_{st}(0) = \begin{bmatrix} \mathbf{R}(\theta) & p(\theta) \\ 0 & 1 \end{bmatrix} g_{st}(0) \tag{2.65}$$

2.6 Early Feasibility Study of Dual-VIO

2.6.1 Different Camera Setup for Dual-VIO

To understand the effects of different camera hardware impacts multi-VIO fusion, a series of preliminary experiments were conducted with different perception modules in the lab. We have conducted the preliminary study with three different hardware:

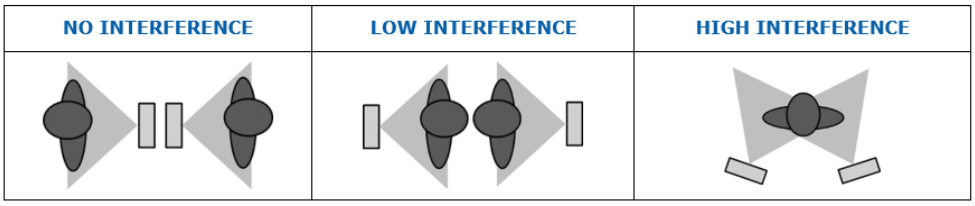


(a) L515 (b) D435i (c) D455

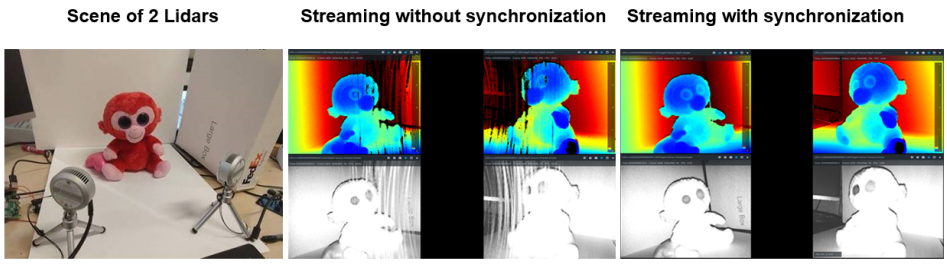
Figure 2.17: Intel RealSense Cameras [5]

1. L515 Intel Realsense Camera (L515) (Figure 2.17(a)) has a rolling-shutter RGB monocular camera with a solid-state LiDAR for depth sensing
2. D435i Intel Realsense Camera (D435i) (Figure 2.17(b)) has a global-shutter IR stereo camera with an IR dot projector but a rolling-shutter RGB monocular camera and an onboard IMU
3. D455 (Figure 2.17(c)) has a global-shutter IR stereo camera, a global shutter RGB monocular camera and an onboard IMU

Based on empirical experiments, we have summarized the following factors that will affect the VINS result:



(a) IR Interference with multi-camera setup [61]



(b) Illustration of the interference [61]

Figure 2.18: Intel RealSense Multi-Camera IR Camera Interference

1. Camera Shutter Type: Rolling shutters can hugely impact monocular and stereo performance. Global shutter cameras are deemed superior, as the modelling assumes all pixels are captured simultaneously. Some papers (e.g., [62]) and works focus on the formulation of the rolling shutter cameras to enable better performance than the baseline that does not take it into account.

2. The active IR cameras interfere with each other, creating interference patterns as illustrated in Figure 2.18 [61].

3. Indoor Lighting: In the context of real-world applications, it is feasible to deactivate the projector and rely on ambient light for imaging purposes. Our experimental framework has demonstrated the continued efficacy of the RGB camera in D455, even under conditions of reduced illumination. The passive IR camera, lacking a dot projector, exhibits notable difficulty in faithfully capturing scenes under similar lighting conditions.

4. Accurate Extrinsic versus Initial Estimation: A good initial estimation can improve the chance of convergence, but an accurate extrinsic from the calibration can ensure the accuracy and convergence of the algorithms.

5. The VINS algorithm assumes the extrinsic based on the transformation from cam-

eras to IMU, T_{ic}: extrinsic from "Color" To "Gyro". Kalibr utility generates the configuration with respect to IMU instead.

2.6.2 Stability of VINS-Mono on a Mobile Manipulator

The stability of VINS-Mono significantly depends on the calibration of the system:

1. IMU parameters with AVar [1] to estimate IMU noise and random walk can cause the system to go unstable since the IMU calibration is done in static without vibration. The run-time IMU parameterization depends on the system vibrations. Both cameras are subject to various IMU vibration noises in our system depending on the robot's pose and motion. Hence, one set of fine-tuning parameters may not guarantee convergence in other scenarios.

2. Online extrinsic parameterization often performs poorly and is only stable in simple motions.

3. Camera soft synchronization can also be a complex issue, as it depends on the version of the firmware, library, and platform used.

In conclusion, the best configuration would be utilizing two D455 cameras with only the global shutter RGB sensor enabled for the best result, with minimal bandwidth and zero interference. Cameras should be calibrated on the powered-on robot for both cameras to determine the parameters for the VINS-Mono to include the nominal vibrations from the system.

Chapter 3

A New Tightly-Coupled Dual-VIO

3.1 Introduction

Mobile robots have been increasingly adopted in warehouses, while the manipulator arm has long operated in many manufacturing plants. A mobile manipulator, a combination of both, has become an attractive candidate for various tasks requiring dynamic movement and dexterous manipulation [63].

Historically, the mobile robot navigation problem comprises three sub-problems; mapping, localization, and motion planning [43]. The first two problems are combined in the context of SLAM, proposed by Smith et al. in 1986 [64]. As the interest in this field grows, SLAM has progressed in various forms with different sensors (IMU, LiDAR, ToF, encoders and vision) with various fusion approaches (filtering or graph optimization) [14]. In particular, V-SLAM has been popular in recent years due to its low cost and rich perception [16]. Often, V-SLAM and VO are used interchangeably in the literature with slightly different objectives; the former focuses on the global and consistent estimate of a device's path, while the latter emphasizes the real-time pose estimations [14]. One may regard the VO with loop-closure as a complete SLAM. To enhance the robustness and performance of SLAM, high rate IMU data have been incorporated to realize Visual-Inertial-SLAM (VI-SLAM) and VIO, which was used for a tightly-coupled fusion with vision in MSCKF [9].

As the SLAM becomes more mature, a new frontier work in Active-SLAM (A-SLAM) has gained attention, aiming to unify all three sub-problems of navigation by bringing the task of actively planning robot locomotion to balance between exploitation (to decrease pose uncertainty) and exploration (to increase the map coverage) [42, 43].

A mobile manipulator can be a great platform to adopt A-SLAM with additional sensors at the End-Effector (EE) of the manipulator. However, many SLAM modules of mobile manipulators rely solely on the base platform aided by other costly sensors such as LiDAR [65]. Some use EE localization with markers (e.g. ArUco Markers) [66, 67] which are often unusable in a rapidly changing unknown environment with dynamic locomotion [63, 66]. Manipulator arms provide greater spatial flexibility and precise kinematic information through high-resolution encoders. However, due to transmission gaps and mechanical vibration, the EE may run into uncontrollable instability in global poses [68]. As a result, the manipulator can be in an uncertain state of kinematics and dynamics [67, 69], especially in the vicinity of singularity. Consequently, the nominal calibration of IMU noise is practically unsuitable for VIO systems when the EE is in dynamic motion.

Many mobile platforms use omnidirectional wheels to enable agile and holonomic movements. However, the performance of dead-reckoning estimates can be poor due to wheel slippage [63, 70]. These wheels are not ideal for VIO, because excitations from uneven terrains can make the odometry diverge, especially when the FOV is limited and features are poorly extracted with frequent occlusions [71]. Moreover, the initialization of VIO may become degenerative for some planar motions, resulting in inaccurate or unobservable spatiotemporal calibration [72, 73]. In fact, for VINS-Mono [18], these motions can be problematic for its initial SfM, causing a delayed initialization.

Mobile manipulators are frequently equipped with an additional camera with IMU near the wrist or the EE (a.k.a. the eye-in-hand vision) for perception and manipulation. This makes two VIOs readily available from the system, one at the mobile base and the other near the EE. Combining these two sensory modules offers great flexibility when one sensor is unreliable or unavailable. However, the increased DOF comes at a price: increased system uncertainties due to coupled excitation when engaged in dynamic movement on different unknown terrains [63, 74]. This results in highly non-uniform pose uncertainties for both VIO modules. Current SLAM systems often fail when the robot's motion is too complex or the environment is too challenging, e.g. highly dynamic locomotion in cluttered environments [44].

This motivates us to develop a more robust low-cost real-time dual-VIO system for a mobile manipulator with dynamic locomotion. In this paper, dynamic locomotion refers to coordinated movement involving both the base platform and the manipulator arm. Figure 3.1 depicts a basic setup for our study, which employs one set of a monocular camera with an IMU installed on the holonomic mobile base (3-DOF SUMMIT XL From Robotnik Inc.) and another set at the EE of the manipulator (7-DOF WAM Arm From Barrett Inc.).

To the authors' best knowledge, this is the first paper introducing a tightly-coupled dy-

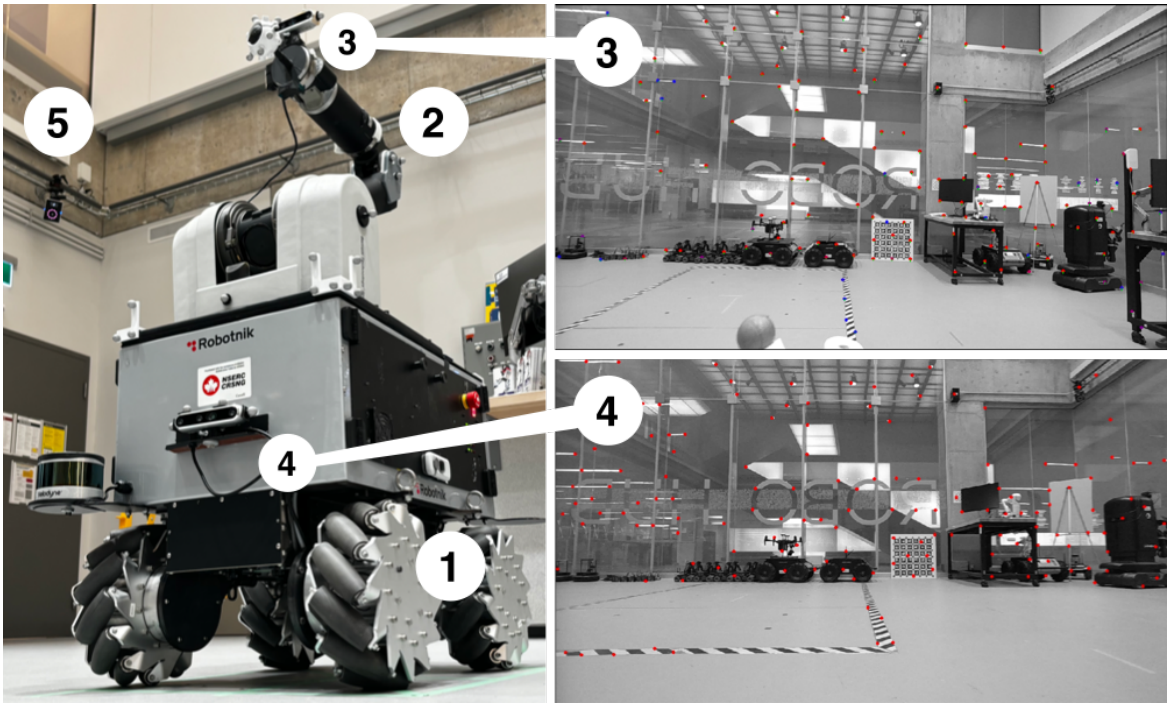


Figure 3.1: Our mobile manipulator comprises a 3-DOF SUMMIT mobile base ① and a 7-DOF WAM robotic arm ②. It has two D455 cameras configured with only monocular-RGB and IMU; one located at the EE ③ and the other at the base ④. A Vicon MoCap system ⑤ is used for Ground Truth (GT). [6]

namic dual monocular VIOs on a mobile manipulator by combining proprioception (VIO) and kinesthesia (arm odometry) at the low level of the factor graph. Our key contributions are two-fold;

- Propose a new tight-coupled dual-VIO approach that balances robustness and computational complexity by incorporating the arm odometry factor.
- Implementation of a modular distributed dual-VIO architecture that can be applied to general configurations of mobile manipulator platforms such as the one with multiple manipulators or a legged robot with a manipulator.

3.2 Related Works

3.2.1 VIO Fusion on Different Platforms

Modelling errors, sensor noise, feature misidentification, and external disturbances are inevitable, hindering accurate estimation and introducing drifts [75]. To address these issues, VIO is frequently integrated with other sensors (e.g. GPS or encoders) tailored towards a specific platform (e.g. aerial, legged, or mobile manipulator) for different environments (e.g. indoor or outdoor). For instance, UAVs often require additional fixed positioning systems to provide low-rate correction. In outdoors, Global Navigation Satellite System (GNSS) is commonly combined with VIO through optimization (e.g. [72, 76]) or filtering (e.g. [77] based on MSCKF). Similarly, UWB data can be fused with VIO (e.g. [78]) in indoors. However, fixed positioning systems are often unavailable in cluttered and unknown environments where mobile and legged robots operate. Instead, dead reckoning sensors are used with VIO. Vehicle encoders are commonly fused tightly into the VIO through the vehicle dynamics model (e.g. [79]) or the kinematics model (e.g. [80]) to improve its robustness.

For legged robots, dynamic locomotion involves more sudden, jerky movements and impacts from ground touchdowns. These movements can significantly disrupt vision sensors, making it challenging for general VIO to estimate the robot's state accurately [81]. Hence, the additional joint encoders can be tightly coupled with VIO in the form of legged odometry factor in the factor graph to minimize the drift [47, 82, 83].

In mobile manipulators, similar challenges arise not only from the ground impacts but also from the dynamic coupling between the base and the arm during the dynamic locomotion. As a result, the base disturbances (from wheel-ground interactions) propagate to the arm vibration, while the arm vibrations propagate back to the base body [70]. This oscillatory coupling amplifies the disturbance experienced at both the base and the EE VIO modules at certain modes, depending on the arm configuration. To address this challenge, [66] has tried to combine the VIO at the mobile base and that at the EE through Moving Horizon Estimator (MHE) with the help of fiducial markers. The study suggests that the IMU is more effective in measuring the EE when compared to the base. The results also indicate that combining the base camera with the arm odometry helps to calibrate the cameras' position with respect to the manipulator and the joint angle biases.

These findings motivated us to develop a dual-VIO system capable of localizing during dynamic locomotion in an unstructured, marker-free environment. This is achieved by simultaneously maintaining two factor-graph-based VIOs closely interacting with each other through arm kinematics.

3.2.2 Multi-VIO Fusion

Depending on the degree or level with which multiple sensors are fused, one may classify a general multi-sensor fusion into two categories;

Loosely-Coupled

The core idea of loosely-coupled fusion is to blend poses from different estimators through a weighted combination. It is a simple and efficient way to unite various sensors and odometry algorithms through a filter-based method, and thus widely used in classical multi-modal fusion (e.g. [41]). [84] showed improved robustness in estimation and faster reconstruction of dense point cloud through a delayed fusion of multiple-VO integrating IMU into an error-state space KF. In fact, the multi-VIO can be considered as an example of Mixture of Experts (MoE) problem [85]. However, failure of one sensor may cause deterioration or divergence of the whole estimation [86]. In practice, it is not easy to characterize different uncertainties and errors [85].

Tightly-Coupled

Tightly-coupled methods prefer to fuse sensor data locally through a single optimization (e.g. non-overlapping multi-cameras [87, 88, 89]) and solve the pose and position jointly. Often, this approach relies on the flexibility and modularity of graph optimization. It may suffer from computational complexity due to scaling issues; the network keeps growing as more optimization states are introduced [85].

In Multi-VIO Fusion, all methods above improved robustness with faster reconstruction, benefiting from the increased FOV and sensory redundancy. We propose a new technique involving a moderate tight-coupling[1] to balance robustness and computational complexity while maintaining the independence of each VIO for system redundancy and failure recovery.

[1]In some literature (e.g. [13]), the term "semi-tight-coupling" has been used, but we adopt "tight-coupling" as a more generally accepted term.

3.3 Formulation of Proposed Dual VIOs

The objective of the dual-VIO is to estimate the poses ($\chi \in \mathbf{SE}(3)$) of the camera bodies with respect to their initial poses in the world frame (w) both at the mobile base (b_B) and the EE (b_E), concurrently on the fly throughout locomotion. Figure 3.2 depicts the configuration of coordinate frames used in this paper. For each motion variable (e.g. the $\chi^w_{b_E}$), the superscript denotes its frame of reference while the subscript denotes the frame whose motion is described by the variable. The pose estimate is directly obtained from the state at frame k (denoted by $\mathbf{x}_{b_k}$) which is to be estimated by the factor graph (See Eq. (3.2)).

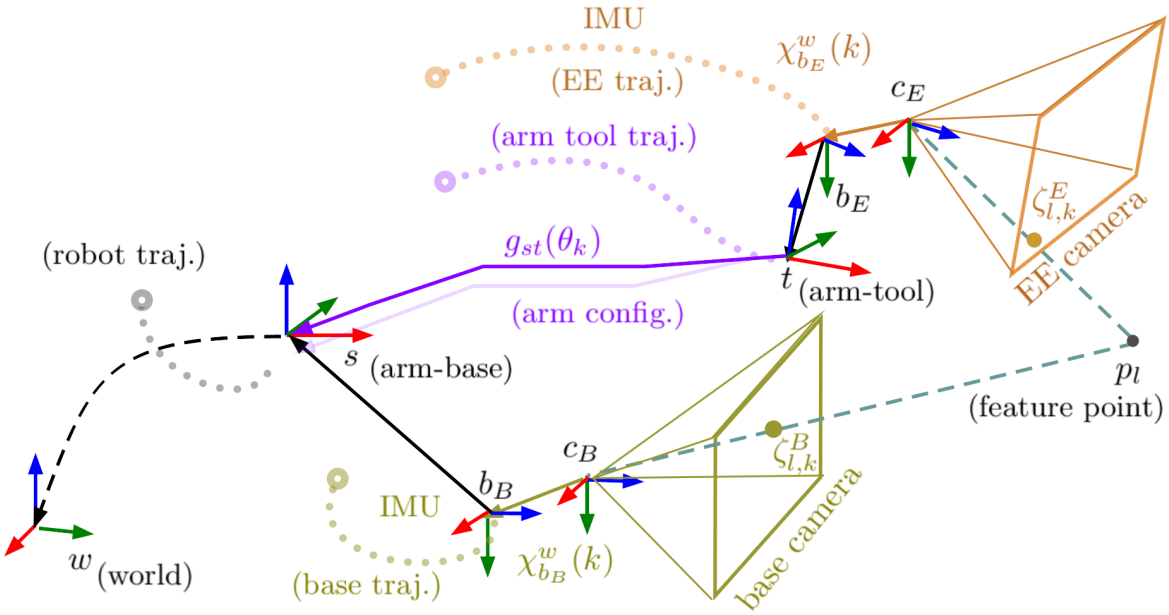


Figure 3.2: Frames and their relations: Solid arrows show constrained transformation; dashed arrows show unconstrained transformation; dotted lines show time-varying trajectories with a starting knot [6]. To simplify matters, we assume that the state estimation is performed at the camera's IMU frame for both the base and the EE (i.e. $\chi^w_{b_B}, \chi^w_{b_E} \in \mathbf{SE}(3)$). The kinematic parameters for the arm are calibrated offline based on the Vicon MoCap and 3D model at its zero-configuration, where the arm is straightened upwards ($g_{st}(\mathbf{0}) \in \mathbf{SE}(3)$).

3.3.1 Formulation of Factor Graph

The proposed dual-VIO maintains two sets of factor graphs running concurrently in parallel to optimize two independent sets of states, one for the EE and the other for the base (i.e. b_E and b_B, respectively, in Fig. 3.2). This is distinct from fusing the optimized odometry in a later stage globally (i.e. loosely-coupled fusion) or jointly optimizing both states (i.e. tightly-coupled fusion). Instead, a moderate coupling is introduced through the arm odometry factor from the previously optimized states at the opposing VIO, propagated through the arm kinematics.

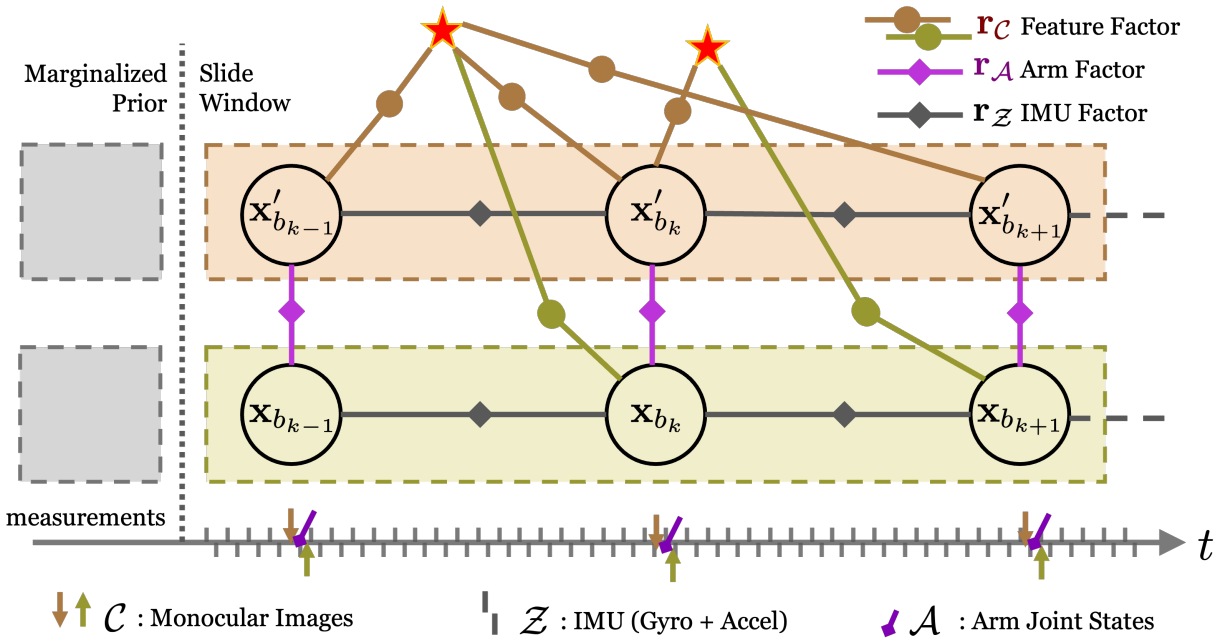


Figure 3.3: Illustration of the sliding window dual factor graph structure with the arm odometry factor coupling two states, with $\mathbf{x}_{b_k}$ as the primary state and $\mathbf{x}'_{b_k}$ as the corresponding secondary state at frame k [6]. The bottom timeline shows an approximated representation of measurement timings and sampling strategy; IMU data is sampled and pre-integrated between two frames, images are software synchronized, and arm joint states are sampled at each frame.

This process is depicted in Fig. 3.3 in the form of primary and secondary VIO. Since either VIO is merely mirroring each other, we will denote $\mathbf{x}_{b_k}$ as the primary VIO state and $\mathbf{x}'_{b_k}$ as the secondary one at the frame k. For instance, from the EE's standpoint, its

primary VIO state (i.e. $\mathbf{x}_{b_k}$) is $\mathbf{x}^w_{b_E}(k)$, while its complementary one ($\mathbf{x}'_{b_k}$) is $\mathbf{x}^w_{b_B}(k)$. This relative designation also applies to the geometric pose $\chi \in \mathbf{SE}(3)$, i.e. the complementary VIO pose is $\chi'_{b_k} = \chi^w_{b_E}(k)$ if its primary VIO pose is $\chi_{b_k} = \chi^w_{b_B}(k)$.

Let us define $\mathcal{K} := \{0, .., n\}$ as the set of integer indices for all the frames from $t_{k=0}$ to $t_{k=n}$ within the horizon. At each frame $k \in \mathcal{K}$ (k is also the discrete-time index for estimation within the horizon), the set of measurements from arm odometry is denoted by $\mathcal{A}_k := \{a_0, ..., a_n\}$, and that for the IMU is denoted by $\mathcal{Z}_k := \{z_0, ..., z_n\}$ between two consecutive frames $t \in \{t_k, t_{k+1}\}$. Similarly, $\mathcal{C}_k$ denotes the set of all the image measurements associated with features that had occurred more than once until frame k. We denote $\zeta_{l,j}$ as an element of $\mathcal{C}_k$, i.e. $\zeta_{l,j} \in \mathcal{C}_k$ ($l \in \{0, ..., m\}$, $j \in \mathcal{K}$) to associate it explicitly with l-th feature in j-th camera frame that occurs more than once, starting from initial appearance.

Each VIO operates over the optimization variable $\mathbf{X}$ for an independent receding horizon sliding window. Thus, $\mathbf{X}$ contains all state vectors $\mathbf{x}_{b_k}$'s ($k \in \mathcal{K}$) in the horizon. Besides, it also includes $\mathbf{p}^b_c$ and $\mathbf{q}^b_c$ which denote the relative position and the relative orientation, respectively, between the camera frame and the corresponding body frame. For computational efficiency, $\mathbf{X}$ also contains m inverse feature depth (denoted by λ_i, $i \in \{0, 1, ...m\}$) computed from all image measurements $\mathcal{C}_k$. Namely, the sliding window operates its optimization over

$$\mathbf{X} = \left[\mathbf{x}_{b_0}, \ldots, \mathbf{x}_{b_n}, \mathbf{p}^b_c, \mathbf{q}^b_c, \lambda_0, \ldots, \lambda_m\right] \tag{3.1}$$

In detail, the state at each frame is a vector

$$\mathbf{x}_{b_k} = \mathrm{col}\left(\mathbf{p}^w_{b_k}, \mathbf{v}^w_{b_k}, \mathbf{q}^w_{b_k}, \mathbf{b}_a, \mathbf{b}_g\right), \quad k \in \mathcal{K} \tag{3.2}$$

where, $\mathbf{p}^w_{b_k} \in \mathbb{R}^3$ denotes position, $\mathbf{v}^w_{b_k} \in \mathbb{R}^3$ is velocity, $\mathbf{q}^w_{b_k} \in \mathbb{Q} \subset \mathbb{R}^4$ is quaternion for orientation, and $\mathbf{b}_a, \mathbf{b}_g \in \mathbb{R}^3$ denote biases of accelerometer and gyroscope, respectively. Note that χ, $\mathbf{X}$ and $\mathbf{x}$ all convey pose information while taking different mathematical forms for different purposes.

The proposed factor graph can be solved by maximizing the posterior estimate of $\mathbf{X}$ using optimization solvers (e.g. the Ceres solver [35]) with the objective function formulated based on BA (similar to [18, 83]) as

$$\min_{\mathbf{X}} \left\{ \begin{array}{l} \|\mathbf{r}_p - H_p\mathbf{X}\|^2 + \sum\limits_{k\in\mathcal{K}} \sum\limits_{\zeta_{l,j}\in\mathcal{C}_k} \rho\left(\|\mathbf{r}_{\mathcal{C}}(\zeta_{l,j}, \mathbf{X})\|^2_{P_{\zeta_{l,j}}}\right) \\ + \sum\limits_{k\in\mathcal{K}} \|\mathbf{r}_{\mathcal{Z}}(z_k, \mathbf{X})\|^2_{P_{z_k}} + \sum\limits_{k\in\mathcal{K}} \|\mathbf{r}_{\mathcal{A}}(a_k(\hat{\mathbf{x}}'_k), \mathbf{X})\|^2_{P_{a_k}} \end{array} \right\} \tag{3.3}$$

where $\mathbf{r}_p$ and H_p in the first term come from the marginalization prior. $\mathbf{r}_{\mathcal{C}}$, $\mathbf{r}_{\mathcal{Z}}$ and $\mathbf{r}_{\mathcal{A}}$ denote the residuals for image, IMU, and arm odometry data, respectively. $P_{\zeta_{l,j}}, P_{z_k}, P_{a_k}$ are the corresponding covariance matrices used to compute the weighted information matrix for each residual (See [18] for more details). The last term plays a key role in our approach by coupling two VIOs through arm odometry (See Fig. 3.3), which is explained in more detail below in subsection 3.3.2.

Solving Eq. (3.3) has $O(n^2)$ space complexity and $O(n^3)$ time complexity with n states in the sliding window. Hence, for M VIO modules in the system, classical tightly coupled VIOs have space and time complexity of $O(M^2n^2)$ and $O(M^3n^3)$, respectively. In comparison, the proposed method treats each VIO as a distributed agent coupled through geometric constraints. Thus, the resulting space and time complexity would be $O(Mn^2)$ and $O(Mn^3)$, respectively.

3.3.2 Incorporation of Arm Odometry Factor

The arm odometry provides an anchor point to minimize catastrophic excitation from the current VIO without sacrificing significant computing resources. It is computed based on the previously optimized complementary VIO state $\left\{\hat{\mathbf{x}}'_{b_0}, \dots \hat{\mathbf{x}}'_{b_n}\right\}$, where the hat notation is used to denote the solution that achieves the optimization problem (3.3).

Manipulator Kinematics

FK of a manipulator maps its N DOF configuration space (which is a subset of $\mathbb{S}^1 \times \cdots \times \mathbb{S}^1 \equiv \mathbb{T}^N$, a.k.a. N-Torus) to the EE pose $\chi^s_t \in \mathbf{SE}(3)$. Using the Lie group theory for $\mathbf{SE}(3)$ [54], we can express FK using a twist coordinate $\boldsymbol{\xi}_i \in \mathbb{R}^6$ for each link, which consists of the axis $\boldsymbol{\omega}_i \in \mathbb{R}^3$ and a point $\mathbf{q}_i \in \mathbb{R}^3$ through it. For group operation, $\boldsymbol{\xi}_i$ can be written with the hat notation as $(\boldsymbol{\xi}_i)^\wedge \in \mathfrak{se}(3)$, i.e. Lie algebra of $\mathbf{SE}(3)$;

$$\boldsymbol{\xi}_i = \begin{bmatrix} -\boldsymbol{\omega}_i \times \mathbf{q}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = \begin{bmatrix} \mathbf{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix}, \quad (\boldsymbol{\xi}_i)^\wedge = \begin{bmatrix} \boldsymbol{\omega}_i & \mathbf{v}_i \\ 0 & 0 \end{bmatrix} \tag{3.4}$$

Accordingly, the FK ($g_{st}(\boldsymbol{\theta}) : \boldsymbol{\theta} \in \mathbb{T}^N \to \chi^s_t \in \mathbf{SE}(3)$) in Lie group can be defined with the product of exponential formula with each twist $(\boldsymbol{\xi}_i)^\wedge \in \mathfrak{se}(3)$ and corresponding angle $\theta_i \in [0, 2\pi]$, w.r.t. zero configuration $g_{st}(\mathbf{0})$ [54] as

$$g_{st}(\boldsymbol{\theta}) = e^{(\boldsymbol{\xi}_1)^\wedge \theta_1} e^{(\boldsymbol{\xi}_2)^\wedge \theta_2} \dots e^{(\boldsymbol{\xi}_N)^\wedge \theta_N}\ g_{st}(\mathbf{0}) \tag{3.5}$$

Arm Odometry

At every k-th frame for the estimated state $\hat{\mathbf{x}}_{b_k}$, its complementary state $\hat{\mathbf{x}}'_{b_k}$ is firstly contracted onto the pose manifold $\hat{\chi}'_{b_k} \in \mathbf{SE}(3)$. The pose evolves through FK $g_{st}(\boldsymbol{\theta})$ to the current odometry frame. The resulting estimated pose incorporating arm odometry $\tilde{\chi}_{b_k} \in \mathbf{SE}(3)$ is lastly lifted onto the local parameter space $a_k = (\tilde{\mathbf{p}}_{b_k}, \tilde{\mathbf{q}}_{b_k}) \in \mathbb{R}^3 \times \mathbb{Q}$. The tilde denotes the estimated variable after incorporating arm odometry. Specifically, for the manipulator illustrated in Fig. 3.2, we can compute the odometry state (i.e. the pose) over $\mathbf{SE}(3)$ for each of the EE and the base VIO's as

$$\begin{aligned} \tilde{\chi}^w_{b_E}(k) &= \hat{\chi}^w_{b_B}(k)\, T_s^{b_B}\, g_{st}(\boldsymbol{\theta})\, T^t_{b_E} \\ \tilde{\chi}^w_{b_B}(k) &= \hat{\chi}^w_{b_E}(k)\, T_t^{b_E}\, g^{-1}_{st}(\boldsymbol{\theta})\, T^s_{b_B} \end{aligned} \tag{3.6}$$

where $T^\bullet_\star$ is the fixed coordinate transform between $\bullet$ and $\star$.

Residual of Arm Odometry

The residual is then computed separately in translation and rotation between the current state and the arm odometry state as

$$\mathbf{r}_{\mathcal{A}}(a_k(\hat{\mathbf{x}}'_k), \mathbf{X}) = \begin{bmatrix} \delta\mathbf{p}_k \\ \delta\mathbf{q}_k \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{p}}^w_{b_k} - \mathbf{p}^w_{b_k} \\ \left(\mathbf{q}^w_{b_k}\right)^{-1} \otimes \tilde{\mathbf{q}}^w_{b_k} \end{bmatrix} \tag{3.7}$$

where $\otimes$ represent quaternion product [90]. The uncertainties at the EE are more prone to disturbances, while the mobile platform is much more stable. Hence, the arm odometry factor imposes heavier weight for the translation residuals at the EE (with $P^{b_E}_{a_k} = \text{diag}\left(\frac{1}{250} I_2, \frac{1}{200}, \frac{1}{10} I_4\right)$ where I_k is a $k \times k$ identity matrix). Conversely, weaker factors are applied in reverse (i.e. those at the base) with $P^{b_B}_{a_k} = \text{diag}\left(\frac{1}{10} I_2, \frac{1}{20}, \frac{1}{10} I_4\right)$ to minimize the impact of noise, disturbance, and kinematic modeling uncertainties from the arm. These values for P_{a_k} are manually tuned w.r.t. the image factor $P_{\zeta_{l,j}} = \frac{1}{520} I_2$.

3.4 Implementation

3.4.1 System Architecture

The detailed implementation of the proposed dual-VIO can be seen from the Figure 3.4. The VINS-Mono codebase is re-architected to be thread-safe and a manager block is introduced to manage the multi-threading of the two estimators for concurrent VIO estimation.

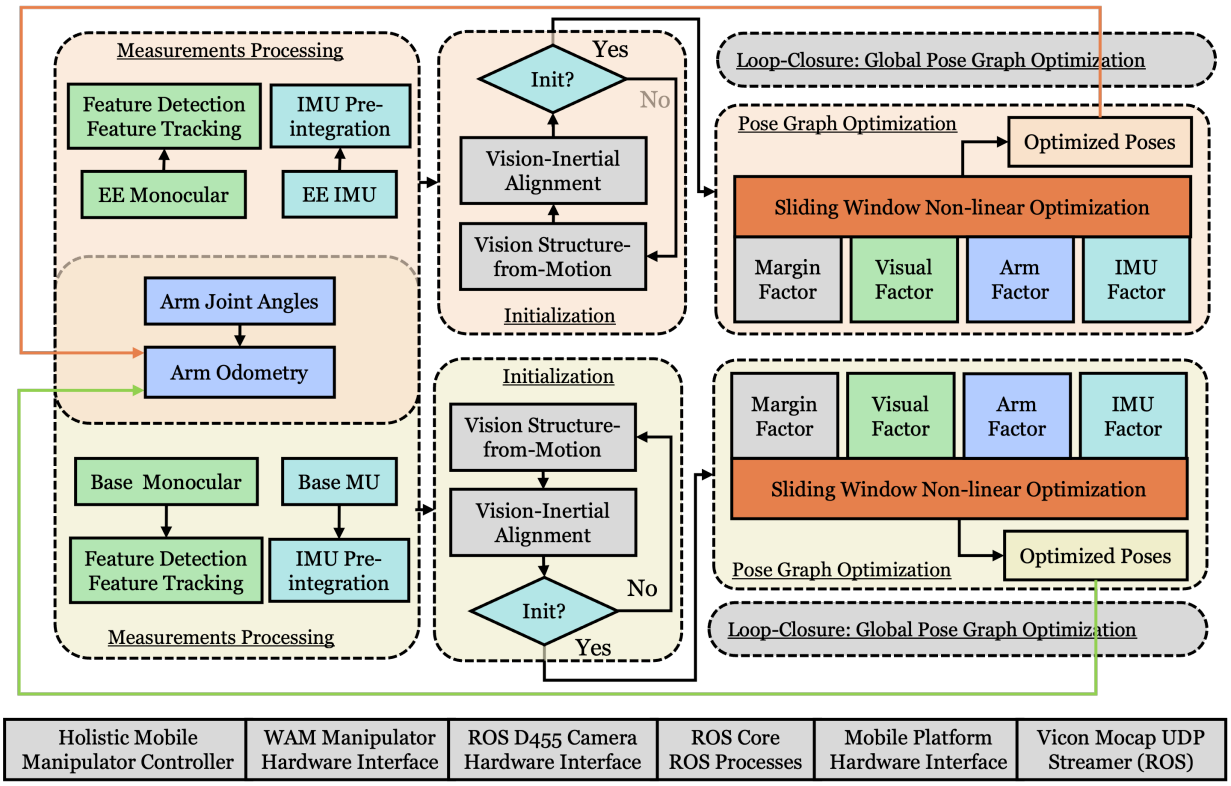


Figure 3.4: The block diagram illustrates the modified pipeline for the proposed dual-VIO with arm odometry. The blue blocks are the innovation parameters, differentiating the implementation difference of baseline and proposed tightly-coupled VIO.

As demonstrated in Figure 3.4[2], real-time performance is upheld through the concurrent operation of dual VIO. This approach involves the simultaneous publication of dual

[2]See detailed implementation in `https://github.com/UW-Advanced-Robotics-Lab/dual-VIO-MM`

odometry and loop-closure trajectories in real time. The baseline dual VIO employs two identical VIO systems operating independently with distinct calibration parameters. In contrast, our proposed solution establishes a tight coupling of the opponent's state as an anchoring point from one system to the other via the arm odometry, facilitating joint optimization of the state in parallel. Furthermore, the arm odometry can be activated or deactivated by passing a flagging argument, allowing for seamless switching between the baseline and proposed methods and providing a consistent and fair evaluation through scripting.

3.4.2 Computational Advantage

As stated at the end of **Section 3.3.1** (Formulation of Proposed Dual VIOs) before Eq. (3.3), the proposed factor graph is solved by maximizing the posterior estimate of $\mathbf{X}$ using optimization solvers (e.g. the Ceres solver [35]) with the objective function formulated based on BA (similar to [18,83]). The linear solver is based on dense Cholesky factorization, where this method has $O(n^2)$ space complexity and $O(n^3)$ time complexity (where the n is the number of states in the sliding window), utilizing the Schur complement trick[3] as discussed in Ceres Solver[4].

To rigorously explain the complexity, please allow us derive the following:

Recall **Section 3.3.1**, the sliding window operates its optimization over

$$\mathbf{X} = \left[\overbrace{\mathbf{x}_{b_0}, \ldots, \mathbf{x}_{b_n}}^{states}, \overbrace{\mathbf{p}_c^b, \mathbf{q}_c^b}^{extrinsic}, \overbrace{\lambda_0, \ldots, \lambda_m}^{measurements} \right] \tag{3.8}$$

The optimizing state at each frame is a vector

$$\mathbf{x}_{b_k} = \text{col}\left(\mathbf{p}_{b_k}^w, \mathbf{v}_{b_k}^w, \mathbf{q}_{b_k}^w, \mathbf{b}_a, \mathbf{b}_g\right), \quad k \in \mathcal{K} \tag{3.9}$$

where, $\mathbf{p}_{b_k}^w \in \mathbb{R}^3$ denotes position, $\mathbf{v}_{b_k}^w \in \mathbb{R}^3$ is velocity, $\mathbf{q}_{b_k}^w \in \mathbb{Q} \subset \mathbb{R}^4$ is quaternion for orientation, and $\mathbf{b}_a, \mathbf{b}_g \in \mathbb{R}^3$ denote biases of accelerometer and gyroscope, respectively.

Hence, the resultant camera block for a single monocular-VIO is in the size of $c = (3+3+4+3+3) = 16$, with a point block in the size of $s = 1$ represented by an inverse

[3]D. C. Brown, A solution to the general problem of multiple station analytical stereo triangulation, Technical Report 43, Patrick Airforce Base, Florida, 1958.

[4]Ceres documentation: `http://ceres-solver.org/nnls_solving.html`

distance. For simplicity, we drop the extrinsic parameters $(\mathbf{p}_c^b, \mathbf{q}_c^b) \in \mathbb{R}^3 \times \mathbb{R}^4$, as they are fixed during the optimization. They can be used in future work on run-time temporal calibration.

Recall, The objective function in Eq. (3.3), the fundamental computational cost derives from solving a linear least square problem of the form without the residual terms:

$$\min_{\mathbf{X}} \left\{ \|\mathbf{r}_p - H_p \mathbf{X}\|^2 \right\} \tag{3.10}$$

Conveniently, Eq. (3.10) is equivalent to solving the normal equation:

$$H_p \mathbf{X} = \mathbf{r}_p \tag{3.11}$$

The resultant complexity of inversing a matrix becomes:

$$H_p \in \mathbb{R}^{(nc+ms)\times(nc+ms)} = H_p \in \mathbb{R}^{(16n+m)\times(16n+m)} \tag{3.12}$$

In practice, we used $m = 1000$ features over a $n = 10$ slide window. This leaves us with an unmanageable optimization for a pure problem in Eq. (3.11), as the space complexity would be $O((16n + m)^2)$ and the time complexity would be $O((16n + m)^3)$.

By block partition $\mathbf{X} = \begin{bmatrix} \Delta x & \Delta\lambda \end{bmatrix}^T$ and $\mathbf{r}_p = \begin{bmatrix} \alpha & \beta \end{bmatrix}^T$, the Eq. (3.11) can be restated as the block structure linear system

$$\begin{bmatrix} B & E \\ E^T & C \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{3.13}$$

where $B \in \mathbb{R}^{nc\times nc}$ is a block sparse matrix with n blocks of size $c \times c$, and $C \in \mathbb{R}^{ms\times ms}$ is a block diagonal matrix with m blocks of size $s \times s$. The $E \in \mathbb{R}^{nc\times ms}$ is a general block sparse matrix.

As the $C \in \mathbb{R}^{m\times m}$ is a diagonal matrix, this implies the inverse of C is cheap with the computation complexity $O(m)$ and the space complexity of $O(m)$ optimally. Thus, it is cheap to eliminate $\Delta\lambda$ with $\Delta\lambda = C^{-1}(\beta - E^T\Delta x)$, resulting:

$$\left[B - EC^{-1}E^T\right] \Delta x = \alpha - EC^{-1}\beta \tag{3.14}$$

The matrix $S = \left[B - EC^{-1}E^T\right]$ is the Schur complement of C in H_p. Hence, the complexity will be inversing the $S \in \mathbb{R}^{nc\times nc}$. Since, in practice, $m \propto n$ scales with n states

in the system, with an example of $(nc = 16 \times 10) \ll (m = 1000)$, thus solving (3.14) is much cheaper than the full problem of (3.11).

Hence, with dense Cholesky factorization, the large bundle adjustment problems can be reduced to (3.14), this leaves us a space and time complexity of $O(n^2)$ and $O(n^3)$, dominated by n number of states in the sliding window $\mathbf{X}$.

Chapter 4

Experiment

4.1 Experimental Setup

The proposed dual-VIO is implemented based on VINS-Fusion under the ROS environment. Two commercially available robotic cameras (D455 from Intel) are configured at the EE and the mobile base as shown in Fig. 3.1. Each camera is enabled only with one global shutter RGB camera (with a resolution of 1280×720 pixels at 30 [Hz]) and one IMU at 200 [Hz]. Both depth modules are disabled due to their interference with each other and the Vicon MoCap system.

The IMU noise parameters were calibrated offline through the AVar [1] based on 5-hour ROS bag files. One collection is performed on a stationary table. Additional characteristics are collected while both cameras are on the powered robot to characterize the combined noise effects from the nominal noise in the system. The on-robot noise parameters are used for the VIO systems.

The intrinsic parameters of each camera are calibrated offline independently with the omnidirectional camera projection model and the radial-tangential distortion model option available through Kalibr [91] over a few minutes of the collection. The collection was performed with individual cameras moving around by hand, with the calibration board in view. Next, the camera-IMU extrinsic parameters are calibrated with its stationary IMU parameters.

We performed direct comparisons against the VINS-Fusion through the ablation study of our proposed method to evaluate the effectiveness of our approach. The baseline data is the dual-VIO with two VINS-Fusion concurrently. All other hyper-parameters are kept consistent throughout the ablation study.

4.1.1 Hardware

Our experimental hardware carries two on-board computers: Jetson Orin (WAM-PC) and Adlink MXE-211 (Summit-PC). The Summit-PC is an intel x86-based computer with the Pre-Emptive Real-Time (PREEMPT-RT) kernel that hosts the ROS master and controls the mobile base through Controller Area Network (CAN). The WAM-PC is an Advanced RISC Machine (ARM) based computer with a standard Linux kernel at Ubuntu 20 (not strictly real-time) that controls the WAM through CAN (with a net-dev support). Both camera modules are connected to the WAM-PC with soft-synchronization, while the GT Vicon MoCap data is published over the network through User Datagram Protocol (UDP) packets. All datasets are collected on WAM-PC through ROS network as '`rosbag`' files.

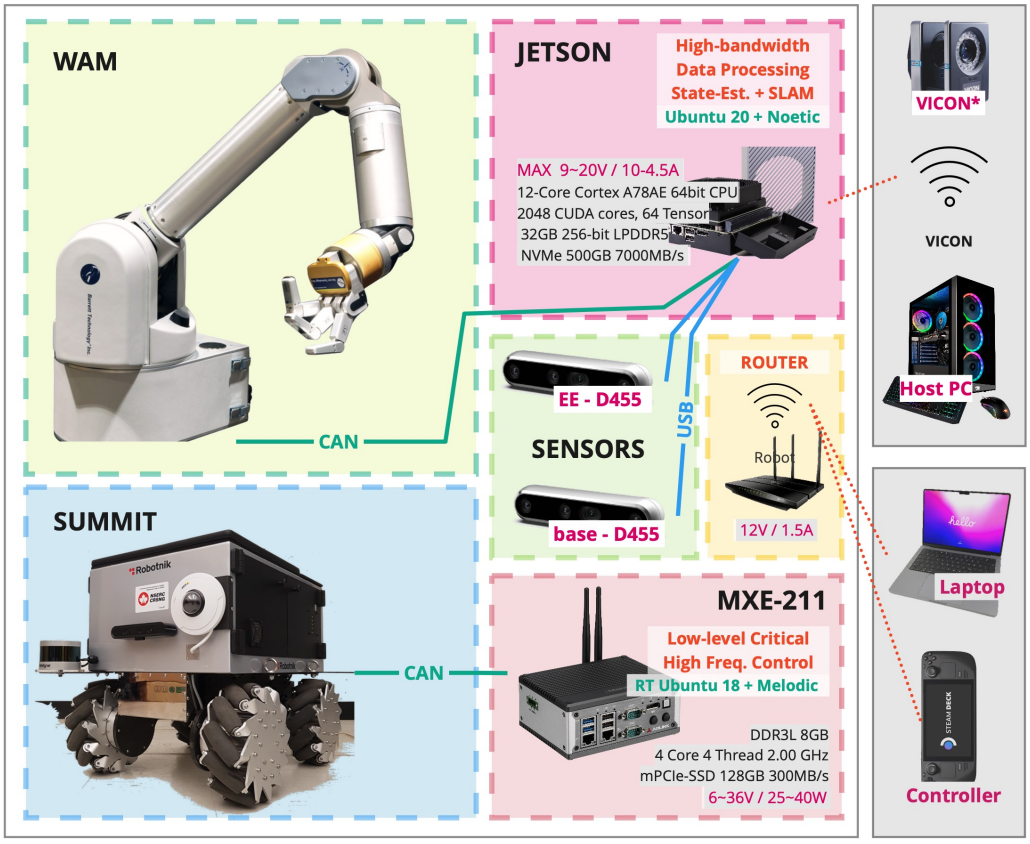


Figure 4.1: Hardware configuration diagram

4.1.2 Environment

The experiment was conducted under a well-lighted and cluttered indoor facility, which involves repetitive objects, subtle uneven terrain, low-feature ceiling, and reflective glass windows as shown in Fig. 4.2.

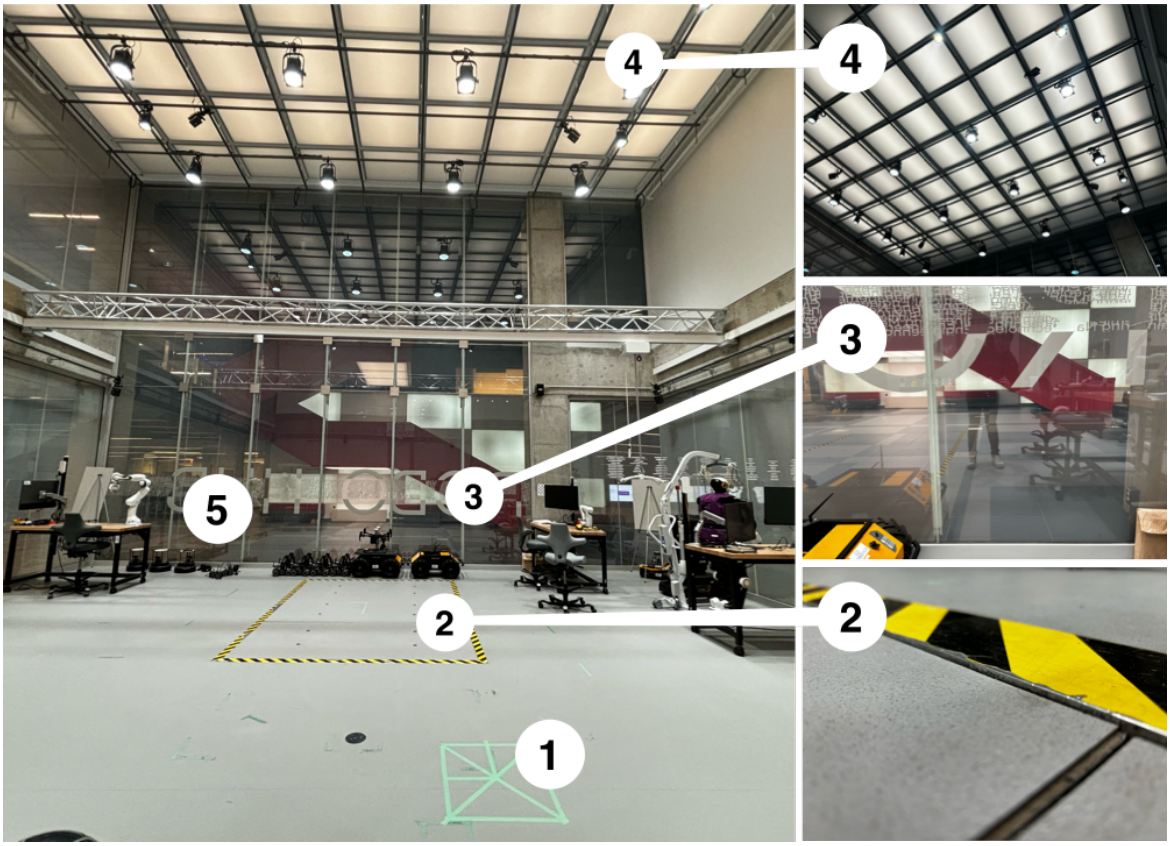
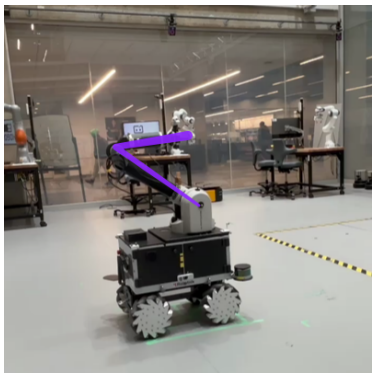


Figure 4.2: The environment of a cluttered facility for experiments; ② uneven terrain , ③ reflective glass window, ④ low-feature illuminated ceiling, and ⑤ repetitive robots. The green masking-tape (①) on the floor helps operators to locate the robot starting position without collision during the experiment [6].

4.1.3 Choreography

The choreography of the mobile manipulator consists of 42 combinations of locomotion and arm maneuvers involving six arm motions (as depicted in Figure 4.3) and seven base motions (as depicted in Figure 4.4 with detailed configurations listed in Table 4.1).



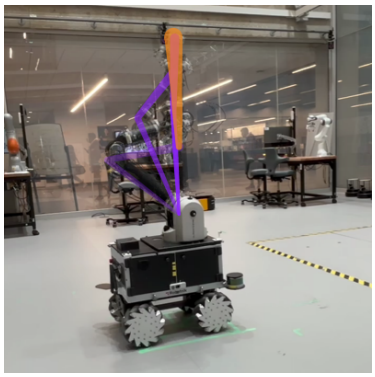
(a) Home (H)



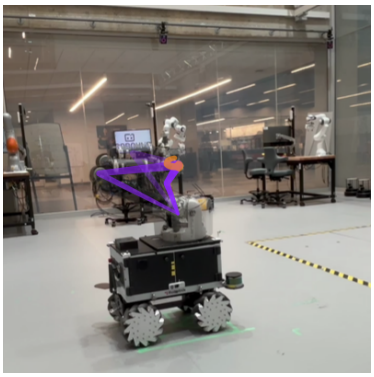
(b) Down (D)



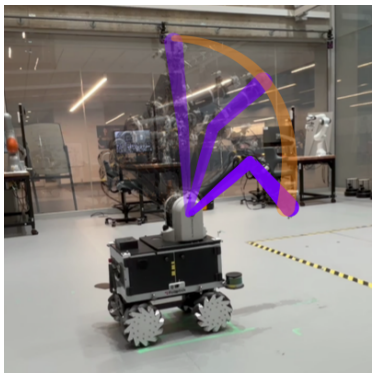
(c) Extended (E)



(d) Up (U)



(e) Left-And-Right (LR)

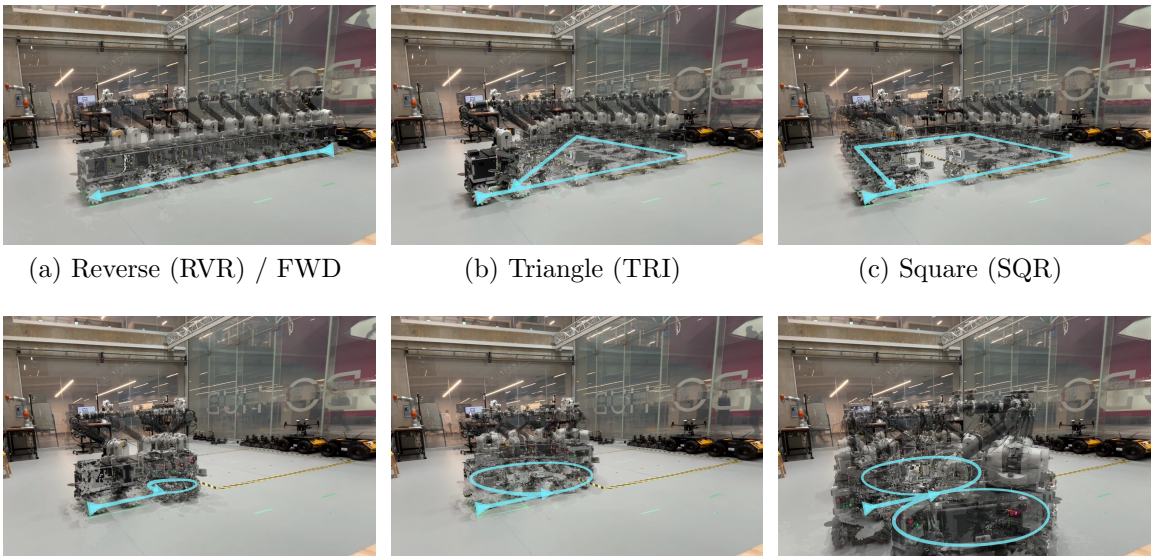


(f) Up-And-Down (UD)

Figure 4.3: A collection of six different arm motions with base fixed. Figure 4.3(a) showcases the home position of the arm without motion; Figure 4.3(b), 4.3(c), and 4.3(d) demonstrate static arm motion by moving the arm to the desired configuration [i.e. Down (D), Extended (E), Up (U)] initially and retracting back to the Home (H) [4.3(a)] at the end of the base movements. Figure 4.3(e) and 4.3(f) describe dynamic oscillating motions by sweeping the arm Left-And-Right (LR) and Up-And-Down (UD), respectively [6].

Figure 4.4 illustrates the seven base maneuvers include three degenerative motions [i.e., Forward (FWD), Reverse (RVR), Spin (SPI))] and four intricate non-holonomic movements

[i.e., Triangle (TRI), Square (SQR), Circle (CIR), and Waggle-Bee-Dance (BEE)].



(a) Reverse (RVR) / FWD (b) Triangle (TRI) (c) Square (SQR)

(d) Spin (SPI) (e) Circle (CIR) (f) Waggle-Bee-Dance (BEE)

Figure 4.4: A collection of seven base motions with arm fixed at the Home (H) position. In particular 4.4(a) showcases both reverse and forward motions, while 4.4(b) and 4.4(c) showcases Triangle (TRI) and Square (SQR) motions. Figure 4.4(d), 4.4(e), and 4.4(f) illustrates Spin (SPI), Circle (CIR), Waggle-Bee-Dance (BEE) motions, respectively.

Table 4.1 lists each configuration, with v_x and v_y denoting linear speeds, ω_z the angular rate, and t_{base} and t_{EE} as time duration for the base and the EE, respectively.

Table 4.1: Choreography Configuration for Seven Base Movements [6]

		Forward FWD	Reverse RVR	Spin SPI	Triangle TRI	Square SQR	Circle CIR	Bee BEE
v_x	m/s	0.2	0.2	0	$\pm 0.3, 0$	$0.3, 0$	0.3	0.3
v_y	m/s	0	0	0	$\pm 0.3, 0$	$\pm 0.3, 0$	0	0
ω_z	rad/s	0	0	0.2	0	0	0.4	± 0.4
t_{base}	s	20	20	30	30	40	32	50
t_{EE}	s	30	30	40	40	50	42	60

4.1.4 Dataset

The proposed system is evaluated through the custom dataset collected with our mobile manipulator platform in the '`rosbag`' form. The dataset comprises two sets of monocular RGB global-shutter cameras and IMU data, which were gathered through six primary arm motions (in Figure 4.3) and seven basic base motions (in Figure 4.4). The dataset also includes Vicon MoCap data, which serves as ground truth (as illustrated in Fig. 3.1).

4.1.5 Alignment

Initially, the Vicon MoCap is aligned when the mobile manipulator is at its zero-configuration as depicted in Figure 4.5.

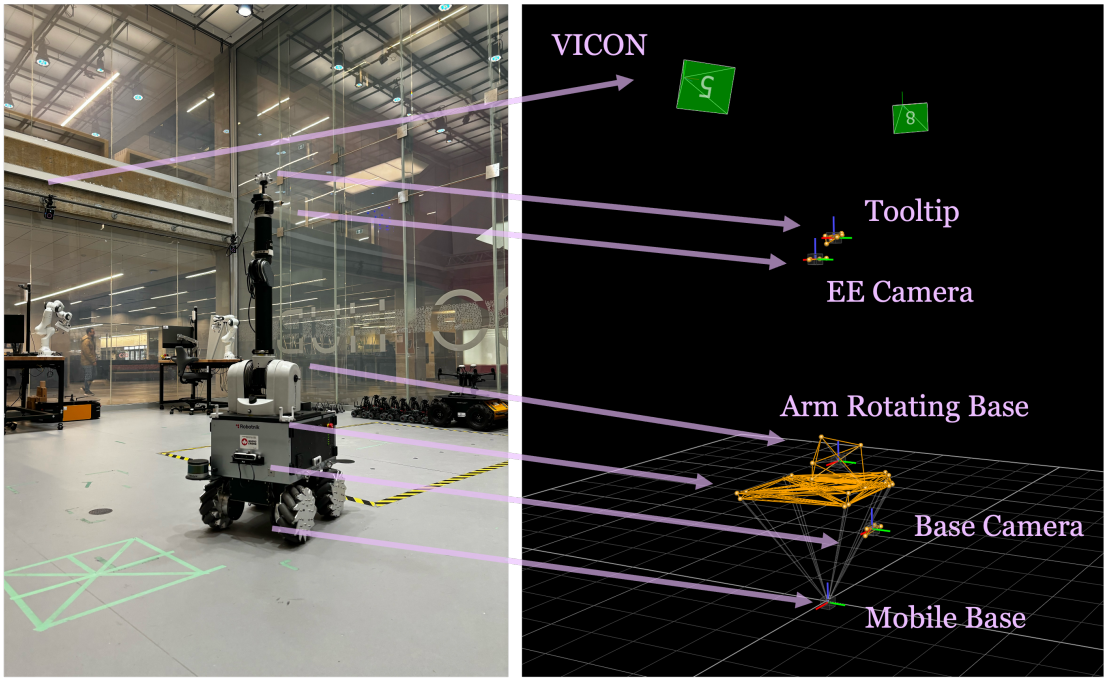


Figure 4.5: Vicon MoCap initial alignment

At the run-time, the GT datasets are offset absolutely after the initialization of each VIO during the algorithm's runtime. This can allow us to directly compare the VIO and corresponding GT from the initialization point [92]. Such methodology ensures an equitable and dependable comparison of the proposed algorithm relative to the baseline.

4.1.6 Calibration Configs

Recall Section 2.3, the IMU and D455 cameras are characterized with Allan Variance [1] and calibrated with Kalibr respectively. The finalized calibration parameters used for VINS-Mono are tabulated in Table 4.2 [1]. To note, the calibration parameters are kept consistent when comparing the baseline and the proposed method for all orchestrated motions.

Table 4.2: Calibration configs used for the mobile base and EE D455

		D455 Mobile Base	D455 End-Effector
Image	Width	1280	1280
	Height	720	720
Distortion	xi	0.1637964957	0.251411578
	k1	0.0194718746	0.067578804
	k2	0.0497254353	0.092162723
	p1	-0.001814677	-0.00431335
	p2	0.0024918334	0.002341003
Projection	gamma1	733.7232928	798.1967750
	gamma2	731.5284724	796.5242037
	u0	643.9825144	642.3813841
	v0	362.6415373	361.5809470
Accelerometer	noise (discrete)	0.283	0.0558
	random walk	0.00212	0.0000525
Gyroscope	noise (discrete)	0.006463	0.00229
	random walk	0.0000267	0.00000196
Gravity		9.8	9.8
Sync	time delta	0.000347597	0.000223268
Tracker	max feature	150	150
	min feat. dist.	30	30
Solver	max solver time	0.04	0.04
	max solver itr.	8	8

[1] Tabulated value may not be complete and truncated for paper display, please refer to the code base for details `https://github.com/UW-Advanced-Robotics-Lab/dual-VIO-MM`

4.1.7 Performance Metric

Quantitative metrics are essential to evaluate the proposal system and the baseline. In geometric control, a combined performance index of translation and rotation on a smooth manifold $\mathbf{SE}(3)$ is often used, with the Frobenius norm, recall eq. (2.61):

$$\begin{aligned} \because \quad & \Psi = \chi_{\hat{\mathbf{q}}_{b_k}^w}^T \, \chi_{\mathbf{q}_k^{\mathrm{GT}}} \in \mathbf{SE}(3) \\ \therefore \quad & \mathbf{e}_{\mathbf{SE}(3)} = \left\| \chi_{\hat{\mathbf{q}}_{b_k}^w}^T \, \chi_{\mathbf{q}_k^{\mathrm{GT}}} - \mathbf{I} \right\|_{\mathrm{F}} \end{aligned} \tag{4.1}$$

By definition, utilizing the homogeneous representation of $\chi \equiv \begin{bmatrix} \boldsymbol{R} & \mathbf{p} \\ 0 & 1 \end{bmatrix} \in \mathbf{SE}(3)$, we can obtain the simplification of:

$$\mathbf{e}_{\mathbf{SE}(3)} = \mathrm{Tr}\left(\mathbf{I} - \boldsymbol{R}_{\hat{\mathbf{q}}_{b_k}^w}^T \, \boldsymbol{R}_{\mathbf{q}_k^{\mathrm{GT}}} \right) + \frac{1}{2} (\mathbf{p} - \mathbf{p}_d)^T (\mathbf{p} - \mathbf{p}_d) \tag{4.2}$$

The orientation error $\mathbf{e}_q$ will always be bounded in group representations, while the translation error $\mathbf{e}_p$ can be unbounded in cartesian space. Hence, when comparing performance in Absolute Trajectory Error (ATE) in a global frame, one should compare orientation and translation independently. Equation (4.2) stated that the combined metric $\mathbf{e}_{\mathbf{SE}(3)} \in \mathbf{SE}(3)$ is essentially equivalent to the summation of decoupled metrics $\mathbf{e}_{\mathbf{p}} + \mathbf{e}_{\mathbf{q}}$.

Specifically, translation error $\mathbf{e}_{\mathbf{p}}$ is evaluated in Euclidean space using the l_2 norm, while rotation error $\mathbf{e}_{\mathbf{q}}$ is assessed by the Frobenius norm on the rotation group $\mathbf{SO}(3)$ (a smooth manifold) towards identity as

$$\mathbf{e}_{\mathbf{p}} = \left\| \hat{\mathbf{p}}_{b_k}^w - \mathbf{p}_{b_k}^{\mathrm{GT}} \right\|_2 \qquad \mathbf{e}_{\mathbf{q}} = \left\| \boldsymbol{R}_{\hat{\mathbf{q}}_{b_k}^w}^T \, \boldsymbol{R}_{\mathbf{q}_k^{\mathrm{GT}}} - \mathbf{I} \right\|_{\mathrm{F}} \tag{4.3}$$

where $\mathbf{p}_{b_k}^{\mathrm{GT}}$ and $\boldsymbol{R}_{\mathbf{q}_k^{\mathrm{GT}}}$ denote the position vector and the rotation matrix, respectively, obtained from the GT data, and $\boldsymbol{R}_{\hat{\mathbf{q}}_{b_k}^w}$ denotes the rotation matrix converted from the estimated quaternion $\hat{\mathbf{q}}_{b_k}^w$.

Decoupling the single scalar metrics over $\mathbf{SE}(3)$ to two separate indices over $\mathbf{SO}(3) \times \mathbb{R}^3$ has the following benefits:

- Allow meaningful comparison and analysis in physical representation when considering their magnitudes and effects on the overall transformation. The combined metrics do not provide much insight into the reliability of the system.

- Rotations and translations have different sensitivities to errors and noise in the measurement process. Separating the two allows one to understand how errors in each component affect the overall transformation independently.
- Often, $\mathbf{SO}(3) \times \mathbb{R}^3$ provides a better interpretability
- Due to different sensitivity and scales, decoupling two provides more reliable insights into the robustness and stability of the system. As the VIO inevitably will drift over time, it is important to separate them for the metric. Hence, decoupling two can provide a meaningful benchmark comparison with other works.

For additional performance metrics, motivated readers may refer to the tutorial papers on quantitative trajectory evaluation for VIO [Z.Zhang][2]. In visual SLAM and VIO, ATE has been most widely used to evaluate algorithms. Some works also show Relative Trajectory Error (RTE) metrics, which are less time-sensitive. However, this metric is more subjective, as one has to define the step size. Some works also show evaluation in separate translations x, y, z and rotation in r, p, y.

To conclude, since we are interested in VIO accuracy with dynamic locomotions, the mean of the ATE is employed as a performance metric. We computed ATE for translation error ($\mathbf{e_p}$) and rotation error ($\mathbf{e_q}$) separately so that they can provide direct insights for us to understand the overall performance of the odometry.

[2][Z.Zhang] Z. Zhang and D. Scaramuzza, "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry" in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid: IEEE, Oct. 2018, pp. 7244-7251. DOI: 10.1109/IROS.2018.8593941.

4.2 Results for Dynamic Arm Motions (LR and UD)

Among six arm motions by seven base motions, we are particularly interested in two dynamic arm motions [Left-And-Right (LR) in fig. 4.3(e) and Up-And-Down (UD) in fig. 4.3(f)] with seven base motions, as they represent the most intricate dynamic locomotions with both systems having independent motions composing a complex choreography.

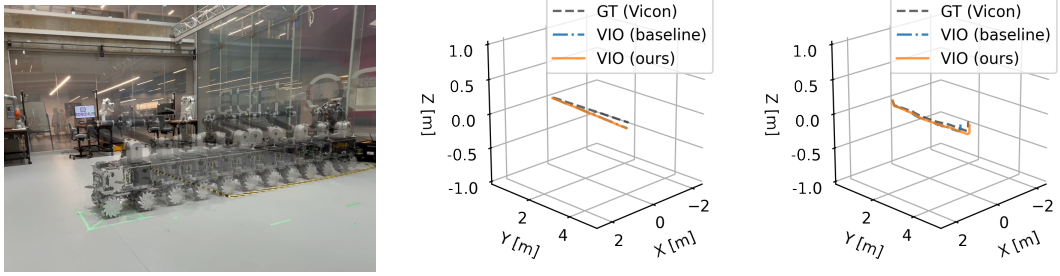
In this section, we will showcase the demonstrations and trajectories in Section 4.2.1 and Section 4.2.2 respectively for Left-And-Right (LR) and Up-And-Down (UD) arm motions. In each section, there are seven demonstration subfigures for seven different base maneuvers and one summary figure comparing the performance of our proposed method with the baseline. Each demonstration figure is composed of three subfigures. The first column in each subfigure is a time-lapse image generated by super-imposing the sampled frames per 2 [s] from the recorded video. The second and third columns are the 3D trajectories plot for the base and EE, correspondingly.

To note, there are three trajectories in each trajectory plot; the black dashed line indicates GT from Vicon MoCap, the blue dash-dotted line indicates the baseline with two VINS-Fusion in parallel, and the solid orange line showcases our proposed method with tightly-coupled dual-VIO methods through arm odometry.

Lastly, we will conclude the experiment with tabulated metrics and discussions in Section 4.2.3.

4.2.1 Demonstrations and Trajectories (Left-And-Right (LR))

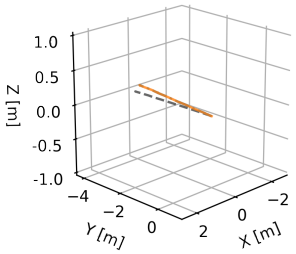
The individual performance results are indicated in subfigures in Figure 4.4 below for the left-right arm motion. At the same time, the overall metric is depicted in Figure 4.5.



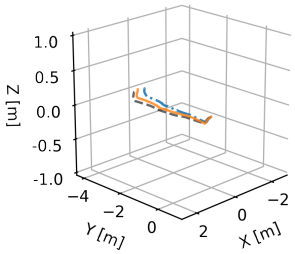
(a) FWD LR Video (b) FWD Base (c) FWD LR EE



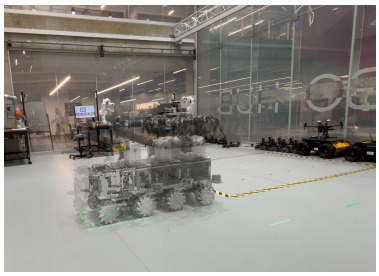
(d) RVR LR Video



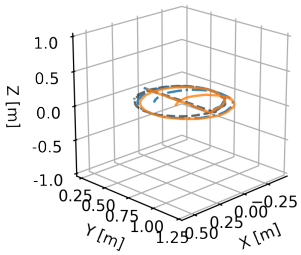
(e) RVR Base



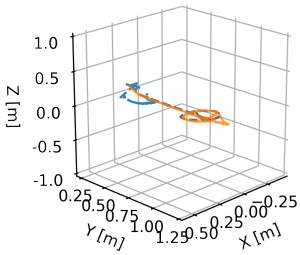
(f) RVR LR EE



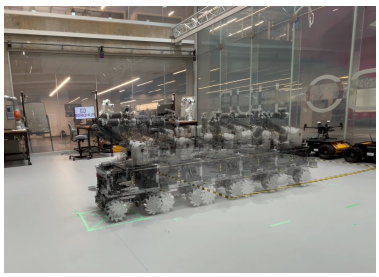
(g) SPI LR Video



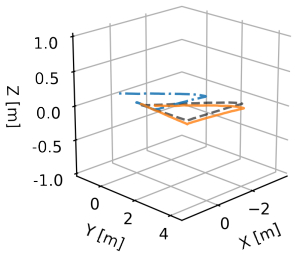
(h) SPI Base



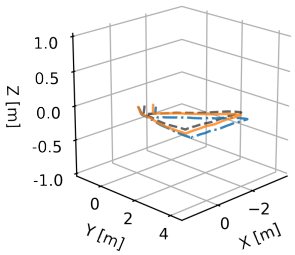
(i) SPI LR EE



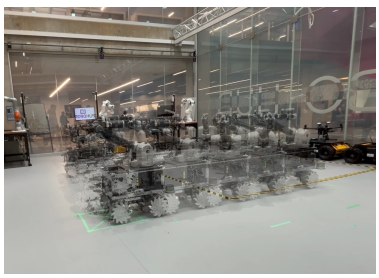
(j) TRI LR Video



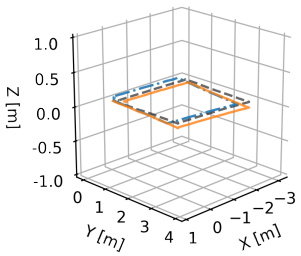
(k) TRI Base



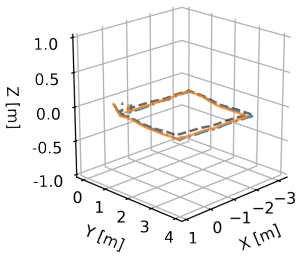
(l) TRI LR EE



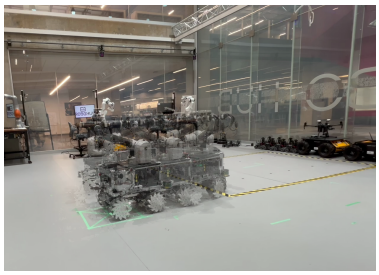
(m) SQR LR Video



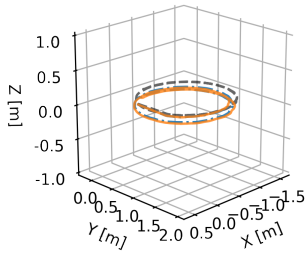
(n) SQR Base



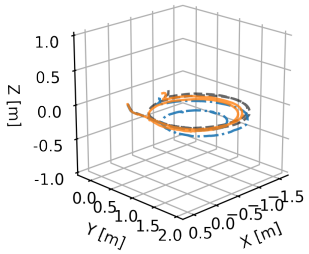
(o) SQR LR EE



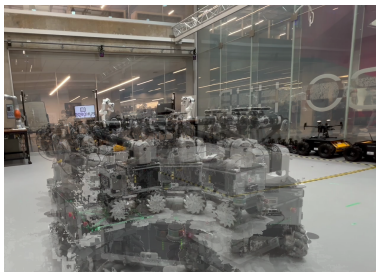
(p) CIR LR Video



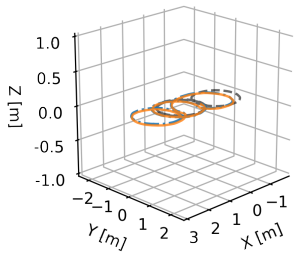
(q) CIR Base



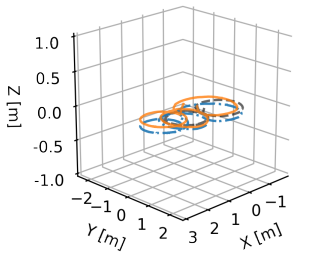
(r) CIR LR EE



(s) BEE LR Video



(t) BEE Base



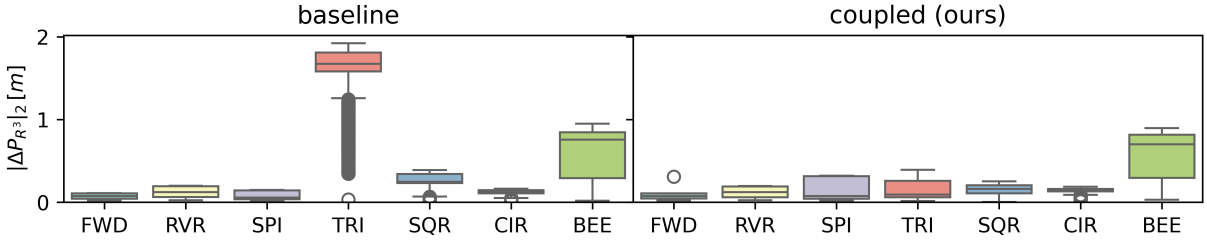
(u) BEE LR EE

Figure 4.4: A collection of 7 time-lapsed images in the first columns (a,d,g,j,m,p,s) and 14 trajectory plots in the second and third columns for dual-VIO (without loop-closure) in seven different base motions (i.e., FWD, RVR, SPI, TRI, SQR, CIR, and BEE) with arm sweeping Left-And-Right (LR) [6]

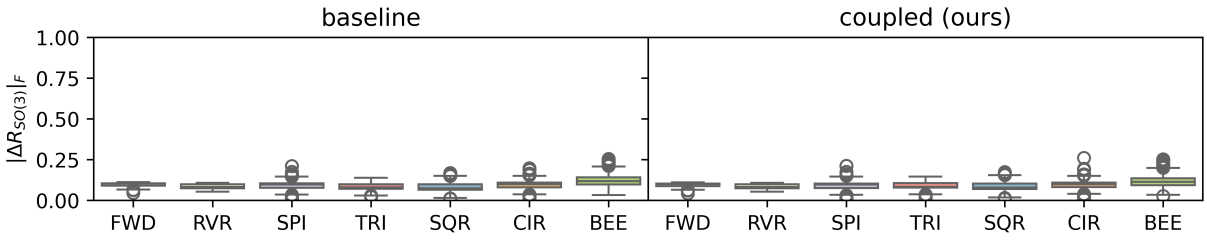
As depicted in Figure 4.4, our proposed method showed more consistent and accurate trajectories in contrast to the baseline method without the arm odometry in both EE and the base VIO.

In particular, the EE motions are much more consistent than the baseline method in all base motions. The improvements in the EE are much more significant. Specifically, the EE trajectories with more complexed base motions, i.e., SPI, TRI, CIR, and BEE as shown in subfigures 4.5(i), 4.5(l), 4.4(r), and 4.4(u) respecitvely.

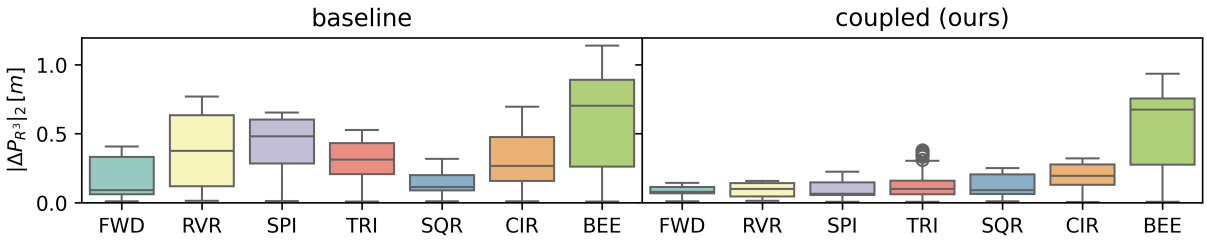
Meanwhile, the base trajectories showed slightly worsened behaviours in SPI and SQR (as depicted in fig. 4.5(h) and 4.5(n)). However, in the case of delayed and inaccurate initialization of the base VIO as depicted in the fig. 4.5(k) for TRI base motion, the base VIO performed much better than the baseline due to the early correction from the EE VIO.



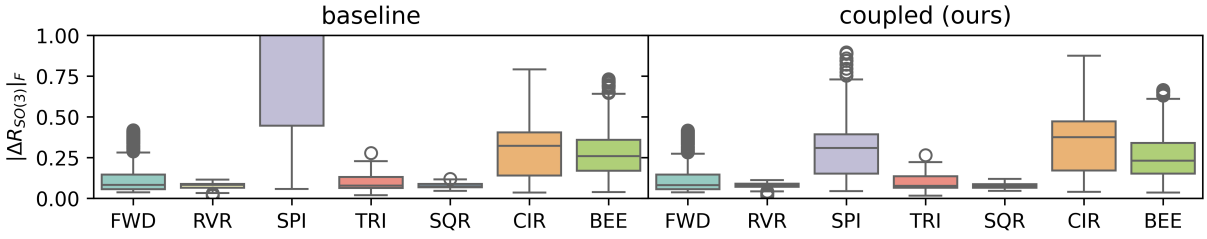
(a) Base translation ATE metrics with LR arm motion



(b) Base orientation ATE metrics with LR arm motion



(c) EE translation ATE metrics with LR arm motion



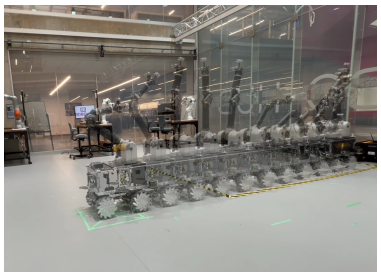
(d) EE orientation ATE metrics with LR arm motion

Figure 4.5: Boxplots for ATE errors in terms of translation and orientation components for both EE and Base VIOs (without loop closure) with seven different base maneuvers while arm waving Left-And-Right (LR) [6]

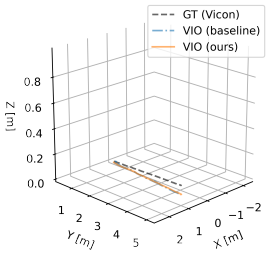
In Left-And-Right (LR) arm motions, our proposed methods showed superior performance from the above trajectories visually in terms of translation. These improvements can also be seen in the decoupled metrics as indicated in the box plots Figure 4.5, which provides a summary metric of our method compared to the baseline. Both fig. 4.5(a) and fig. 4.5(c) showed significant improvements in both mean translation error and variance. The orientation is, however, insignificant in contrast to the baseline model as indicated in both fig. 4.5(b) and fig. 4.5(d). For degenerative pure base rotation (SPI) motion, we see significant improvements and stability in both base and EE VIO in terms of absolute translation and orientation.

4.2.2 Demonstrations and Trajectories (Up-And-Down (UD))

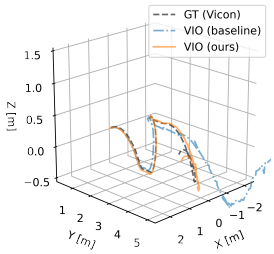
Similarly, for UD arm motions, the individual performance results are indicated in subfigures in Figure 4.4. At the same time, the overall metric is depicted in Figure 4.5.



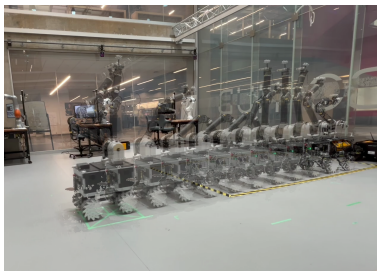
(a) FWD UD Video



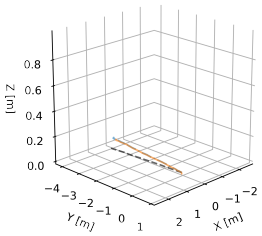
(b) FWD Base



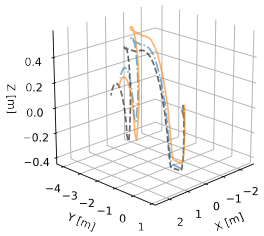
(c) FWD UD EE



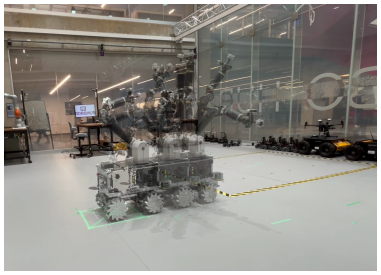
(d) RVR UD Video



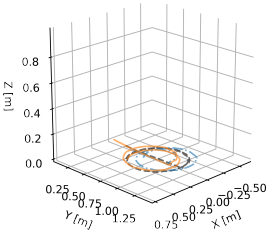
(e) RVR Base



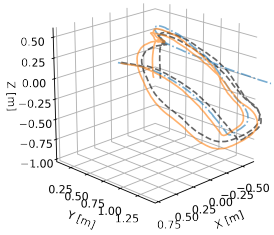
(f) RVR UD EE



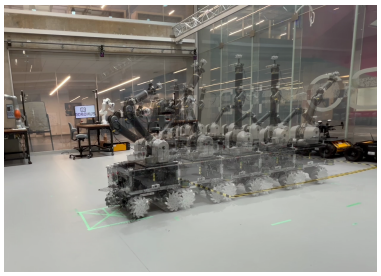
(g) SPI UD Video



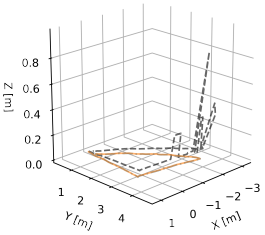
(h) SPI Base



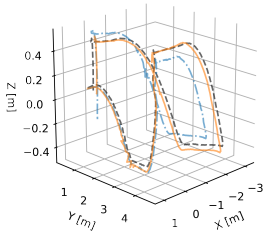
(i) SPI UD EE



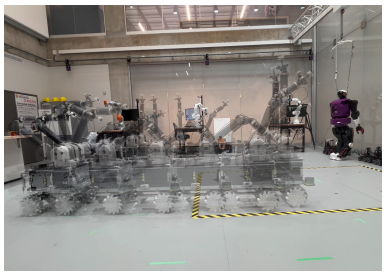
(j) TRI UD Video



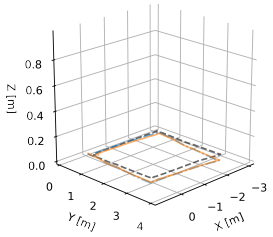
(k) TRI Base



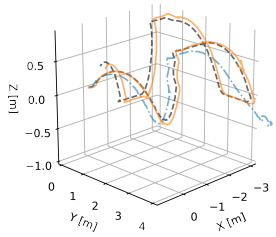
(l) TRI UD EE



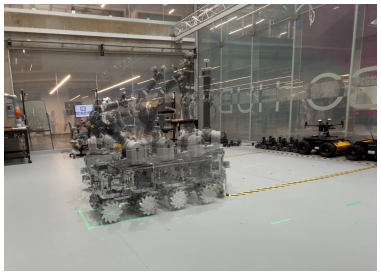
(m) SQR UD Video



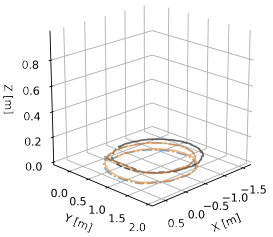
(n) SQR Base



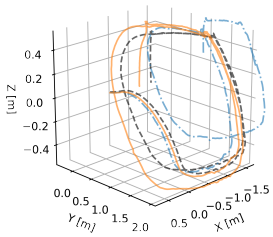
(o) SQR UD EE



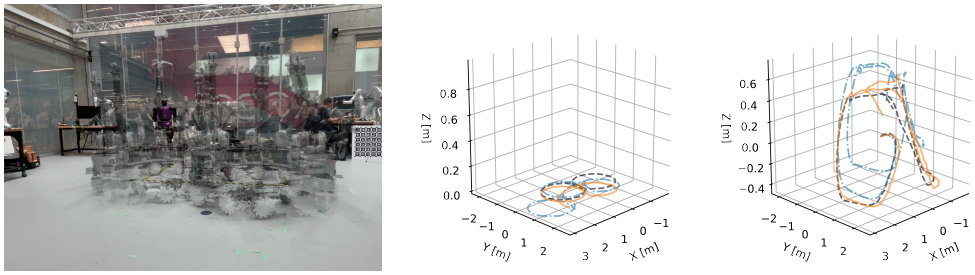
(p) CIR UD Video



(q) CIR Base



(r) CIR UD EE



(s) BEE UD Video (t) BEE Base (u) BEE UD EE

Figure 4.4: A time-lapsed image (a) and trajectory plots (b,c) for dual-VIO (without loop-closure) in Waggle-Bee-Dance (BEE) base motion with arm sweeping Up-And-Down (UD) [6]

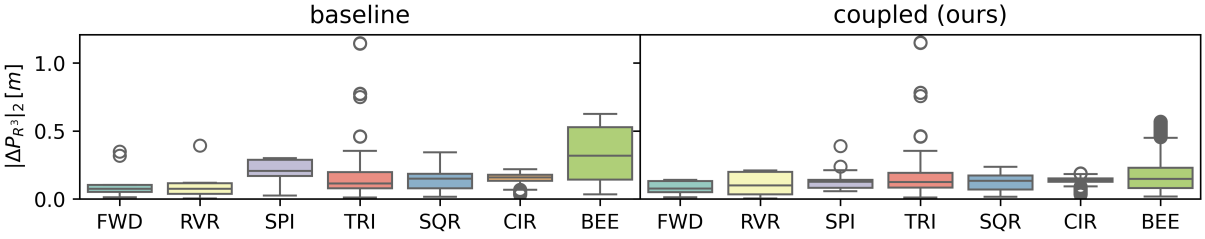
From Figure 4.6(a) to Figure 4.4(s), our proposed method provides more consistent and accurate trajectories in contrast to the baseline method without the arm odometry in both EE and the base VIO.

In particular, our proposed method showed superior performance at the EE for all base motions. In particular, the significance and consistency of the proposed methods can be visually observed in FWD, SPI, TRI, SQR, CIR and BEE trajectories as depicted in figures 4.6(c), 4.5(i), 4.5(l), 4.5(o), 4.5(r), and 4.4(u) respectively.

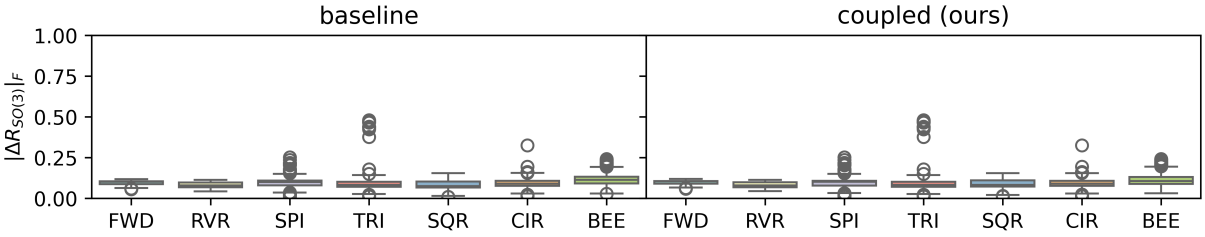
Among all seven base motions, the SPI and SQR motions appear to be most challenging for EE VIO to provide stable numerical results throughout the motions for the baseline method, resulting a diverging estimation, while the base VIO had a much more stable estimation throughout all motions. By fusing the state of the base VIO through arm trajectories, our method provides a stable estimation at the EE VIO. This further highlights the robustness of the proposed method without the loss of the EE VIO upon abnormal disturbances.

The improvements in EE VIO did not sacrifice the base estimation throughout all seven dynamic locomotions, rather an improved estimation for the SPI and BEE base VIO estimation. This further supports the benefits of this new tight-coupling approach.

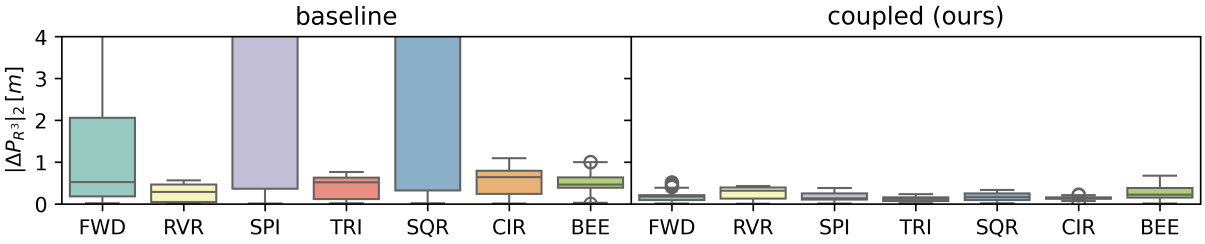
As demonstrated in all seven trajectories, our methods show much more stable behaviour and perform better, as depicted in the box plots from Figure 4.5. Our approach exhibits greater resilience to disturbances, maintaining stability and convergence even when subjected to significant perturbations. In contrast, the baseline model is more susceptible to such disturbance, leading to diverging estimation.



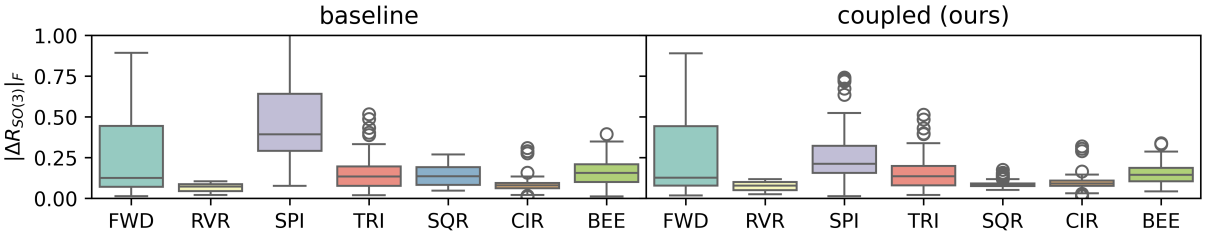
(a) Base translation ATE metrics with UD arm motion



(b) Base orientation ATE metrics with UD arm motion



(c) EE translation ATE metrics with UD arm motion



(d) EE orientation ATE metrics with UD arm motion

Figure 4.5: Boxplots for ATE errors in terms of translation and orientation components for both EE and Base VIOs (without loop closure) with seven different base maneuvers while arm waving Up-And-Down (UD) [6]

4.2.3 Performance Results and Discussion (LR and UD)

As shown in boxplots from fig. 4.5 and fig. 4.5 earlier in this section, both the translation error and variance are significantly lower and more consistent for both VIO modules at the base and especially at the EE. However, the improvements in orientation are not obvious in fig. 4.5(d) and fig. 4.5(d). Hence, the tabulated mean ATE metrics are summarized in Table 4.3 and Table 4.4 for translation and orientation error, respectively, with bolded numbers to highlight our methods with minimized errors.

Table 4.3: Mean ATE in translation $\mathbf{e_p}$ at VIOs with dynamic locomotion

	Base VIO [m] ($\Delta\mathbf{p}_{R3}$)				EE VIO [m] ($\Delta\mathbf{p}_{R3}$)			
	Basline (VINS)		Ours (coupled)		Basline (VINS)		Ours (coupled)	
	LR	UD	LR	UD	LR	UD	LR	UD
FWD	0.071	0.076	0.075	0.086	0.179	1.966	**0.089**	**0.188**
RVR	0.120	0.073	**0.119**	0.110	0.389	0.277	**0.094**	**0.264**
SPI	0.076	0.196	0.146	**0.121**	0.447	4.884	**0.094**	**0.173**
TRI	1.582	0.146	**0.149**	0.150	0.309	0.402	**0.142**	**0.121**
SQR	0.264	0.134	**0.150**	**0.127**	0.147	45.691	**0.125**	**0.168**
CIR	0.115	0.147	0.134	**0.132**	0.324	0.560	**0.197**	**0.137**
BEE	0.579	0.347	**0.554**	**0.199**	0.586	0.502	**0.537**	**0.277**

Table 4.4: Mean ATE in orientation $\mathbf{e_q}$ at VIOs with dynamic locomotion

	Base VIO ($\Delta R_{\mathbf{SO}(3)}$)				EE VIO ($\Delta R_{\mathbf{SO}(3)}$)			
	Basline (VINS)		Ours (coupled)		Basline (VINS)		Ours (coupled)	
	LR	UD	LR	UD	LR	UD	LR	UD
FWD	0.133	0.241	**0.133**	0.243	0.096	0.095	**0.094**	0.097
RVR	0.076	0.067	**0.076**	0.075	0.084	0.081	**0.084**	**0.081**
SPI	1.131	0.444	**0.288**	**0.241**	0.090	0.094	**0.090**	**0.094**
TRI	0.096	0.142	**0.096**	0.145	0.079	0.210	0.086	**0.210**
SQR	0.076	0.138	**0.075**	**0.083**	0.078	0.079	0.082	0.085
CIR	0.304	0.077	0.357	0.092	0.096	0.091	**0.095**	**0.091**
BEE	0.277	0.157	**0.255**	**0.145**	0.119	0.113	**0.114**	**0.110**

As both Table 4.3 and Table 4.4 represented, the ATE metrics of the proposed method are much smaller and more consistent than the baseline data (as highlighted in bold) for

intricate dynamic locomotions involving complex base maneuvers (SPI, TRI, SQR, CIR, BEE) and active arm motions (LR, UD).

In particular, our proposed methods showed superior performance in EE VIO estimations for all intricate locomotions. Despite these improvements came with little worsened estimations in the base VIO in some scenarios, the base VIO performance is more consistent regardless of the base motions.

4.3 Additional Results for Static Arm Motions

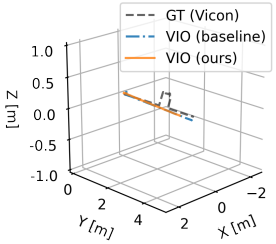
The rest of the locomotion (i.e. four static arm motions by seven base movements) is summarized in this section with each arm motion per subsection, i.e., Home (H), Down (D), Extended (E), and Up (U) positions.

4.3.1 Demonstrations and Trajectories (Home (H))

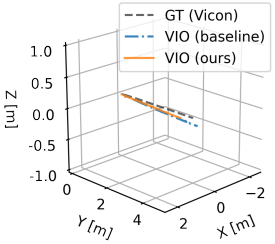
From Figure 4.5, the improvements in both base and EE VIO trajectories are insignificant.



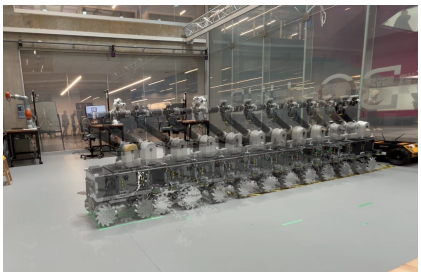
(a) FWD Home (H) Video



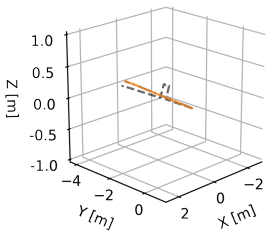
(b) FWD Base



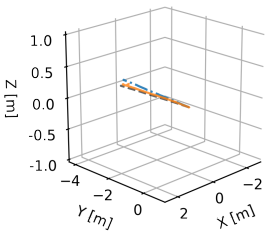
(c) FWD H EE



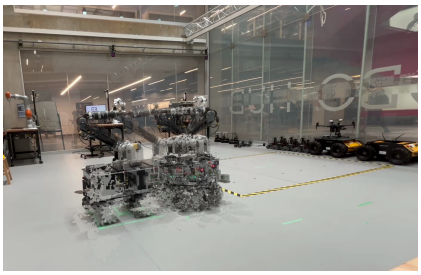
(d) RVR H Video



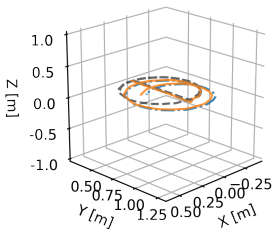
(e) RVR Base



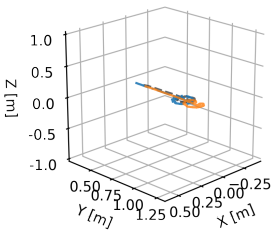
(f) RVR H EE



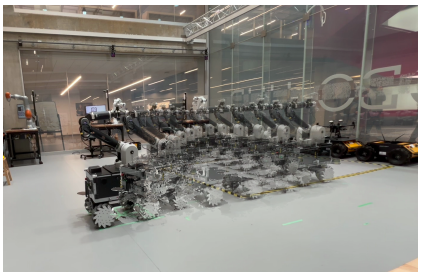
(g) SPI H Video



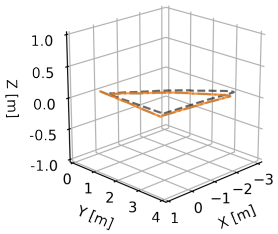
(h) SPI Base



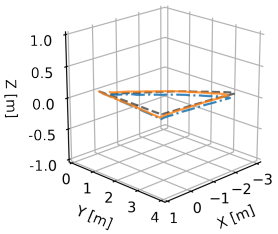
(i) SPI H EE



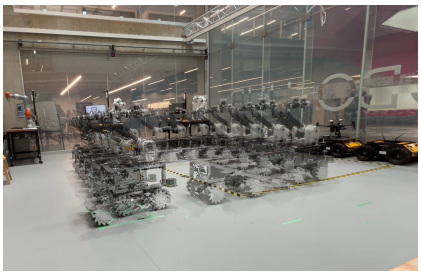
(j) TRI H Video



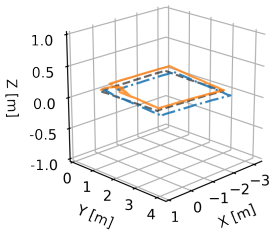
(k) TRI Base



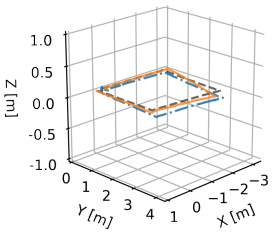
(l) TRI H EE



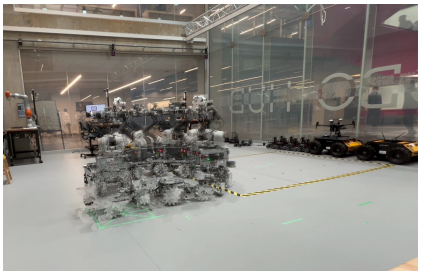
(m) SQR H Video



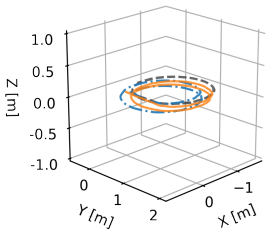
(n) SQR Base



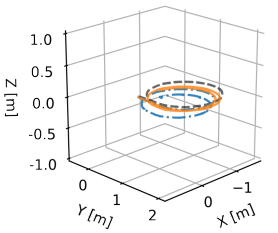
(o) SQR H EE



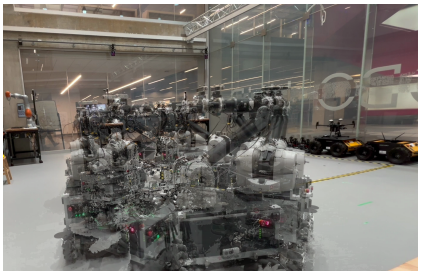
(p) CIR H Video



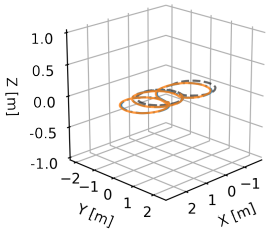
(q) CIR Base



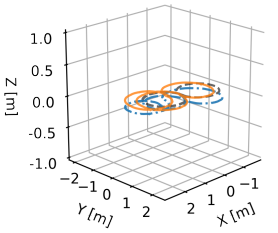
(r) CIR H EE



(s) BEE H Video



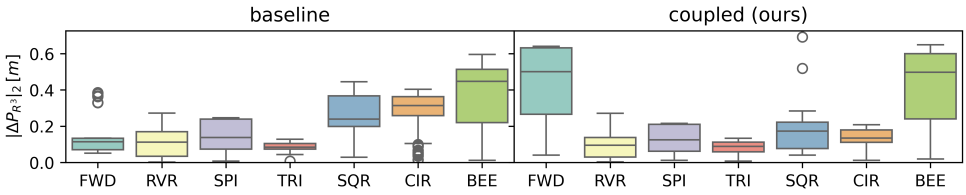
(t) BEE Base



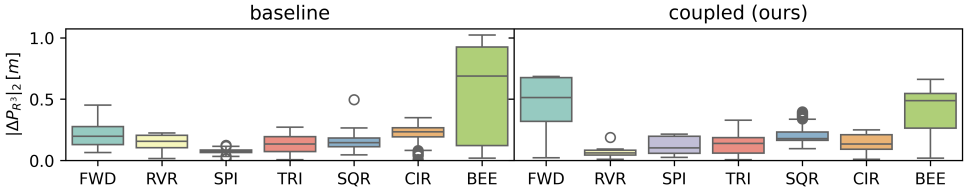
(u) BEE H EE

Figure 4.5: A time-lapsed image (a) and trajectory plots (b,c) for dual-VIO (without loop-closure) in Waggle-Bee-Dance (BEE) base motion with arm at Home (H) position

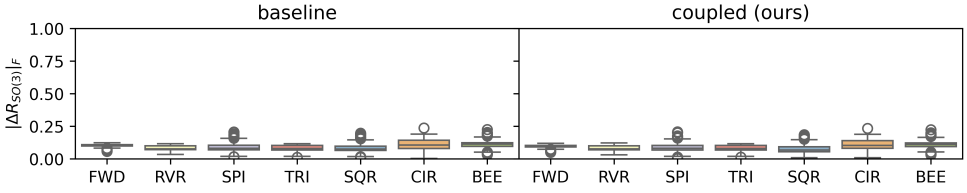
However, our approach provides better trajectories in both base and EE VIO for SQR and CIR base motions, as fig. 4.6 indicates slight improvements. Overall, the improvements are less significant and even worsen performance in the simplest motions as FWD. This is as expected due to modelling errors in arm kinematics because our arm kinematics and camera extrinsic are not calibrated through eye-hand calibration. Camera-arm extrinsic is calibrated through offsetting constants obtained during Vicon MoCap manually.



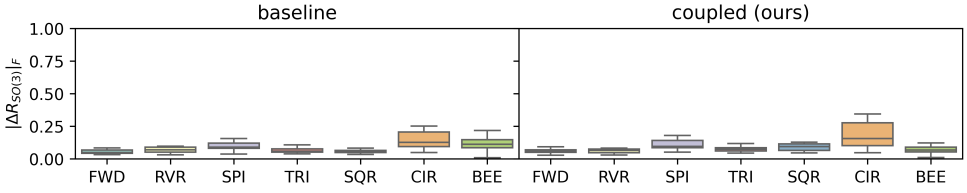
(a) Base translation metrics



(b) EE translation metrics



(c) Base orientation metrics

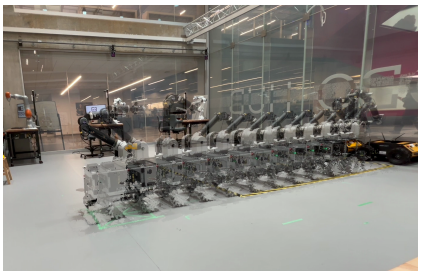


(d) EE orientation metrics

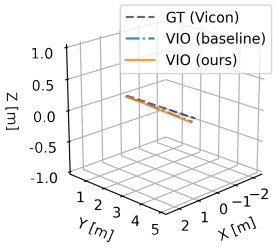
Figure 4.6: Boxplots for ATE errors for both base and EE VIOs (without loop closure) in terms of translation and orientation components at the Home (H) configuration.

4.3.2 Demonstrations and Trajectories (Down (D))

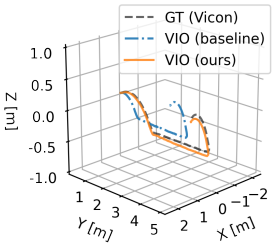
For the Down (D) arm position with seven different base motions, our proposed methods outperform the baseline methods in EE VIO particularly.



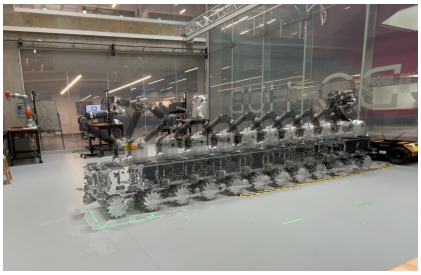
(a) FWD Down (D) Video



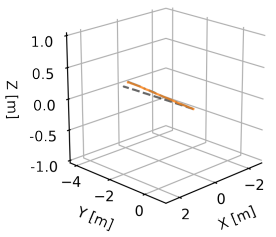
(b) FWD Base



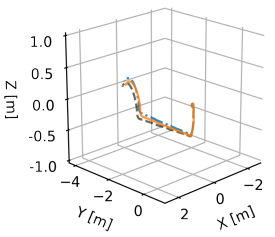
(c) FWD D EE



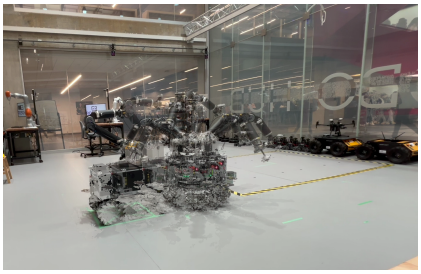
(d) RVR D Video



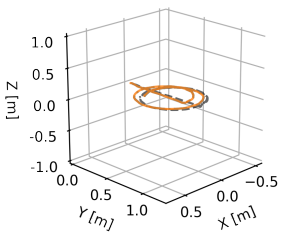
(e) RVR Base



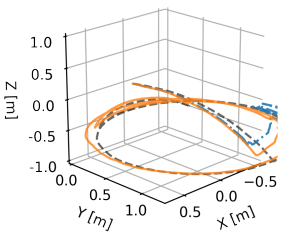
(f) RVR D EE



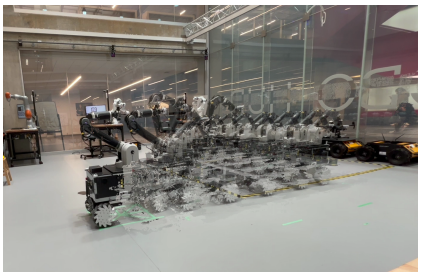
(g) SPI D Video



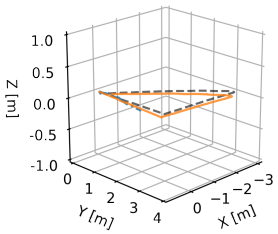
(h) SPI Base



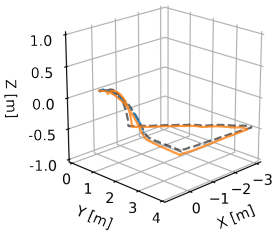
(i) SPI D EE



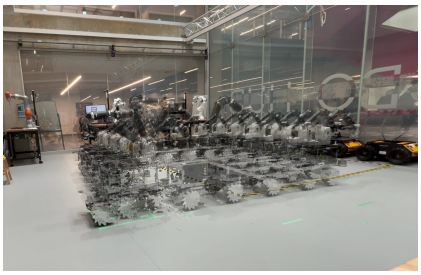
(j) TRI D Video



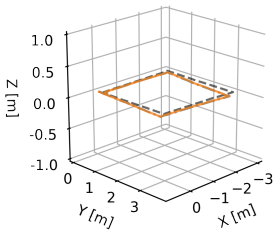
(k) TRI Base



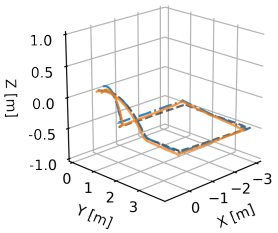
(l) TRI D EE



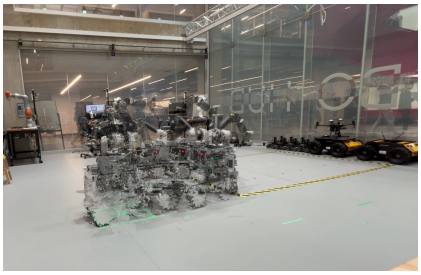
(m) SQR D Video



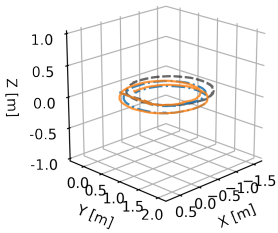
(n) SQR Base



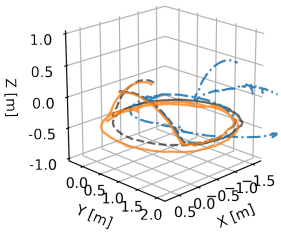
(o) SQR D EE



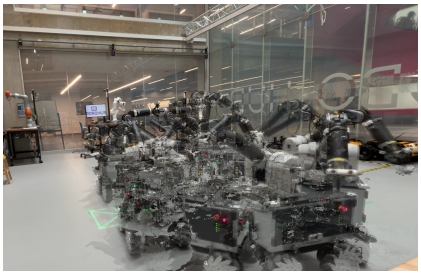
(p) CIR D Video



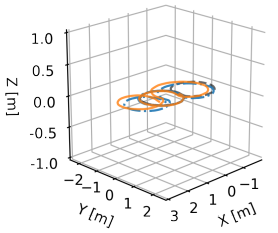
(q) CIR Base



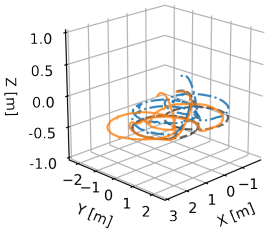
(r) CIR D EE



(s) BEE D Video



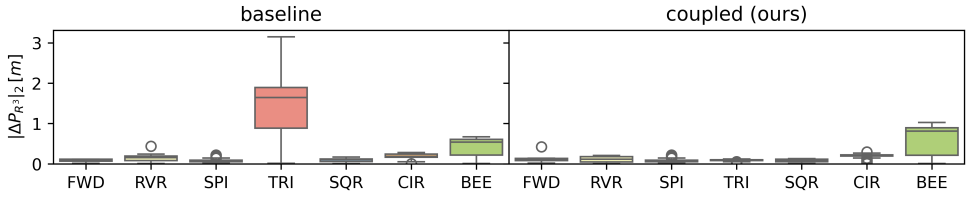
(t) BEE Base



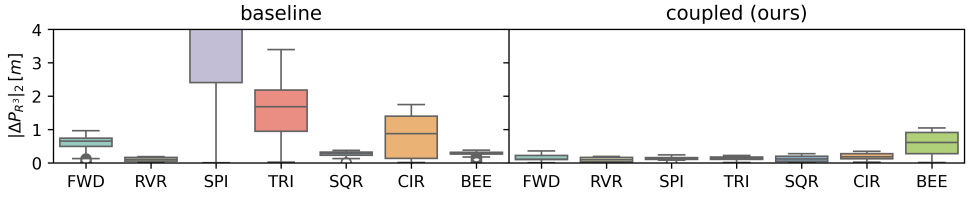
(u) BEE D EE

Figure 4.6: A time-lapsed image (a) and trajectory plots (b,c) for dual-VIO (without loop-closure) in Waggle-Bee-Dance (BEE) base motion with arm at Down (D) position

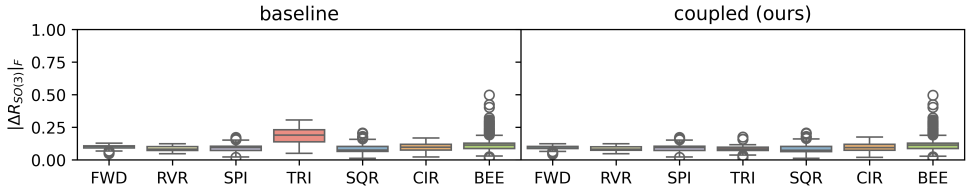
As illustrated in Figure 4.7 below, the performance improvements in EE are significant, while the baseline method suffers stability issues at the EE VIO.



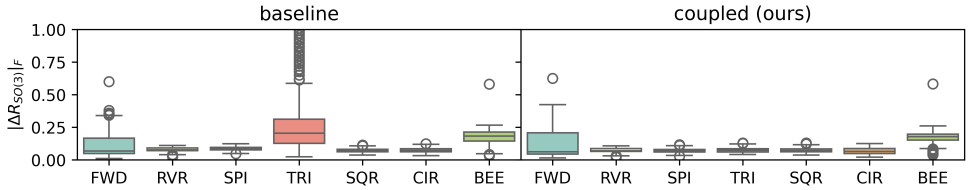
(a) Base translation metrics



(b) EE translation metrics



(c) Base orientation metrics

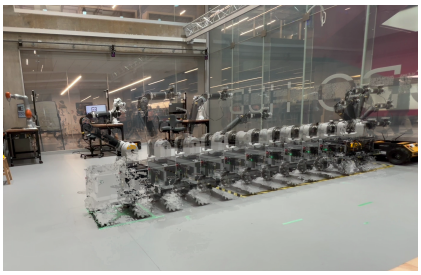


(d) EE orientation metrics

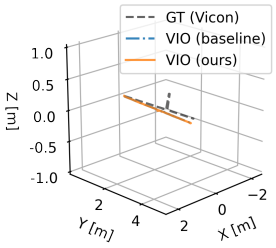
Figure 4.7: Boxplots for ATE errors for both base and EE VIOs (without loop closure) in terms of translation and orientation components at the Down (D) configuration.

4.3.3 Demonstrations and Trajectories (Extended (E))

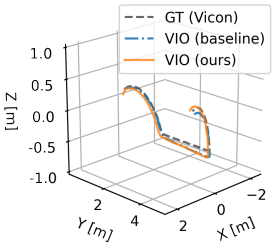
For the Extended (E) arm position with seven different base motions, our proposed methods outperform the baseline methods in EE VIO particularly.



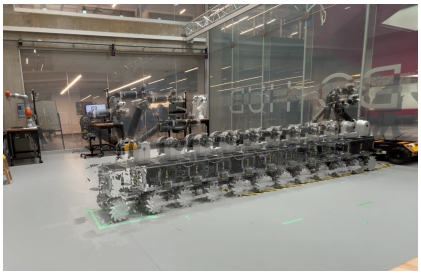
(a) FWD Extended (E) Video



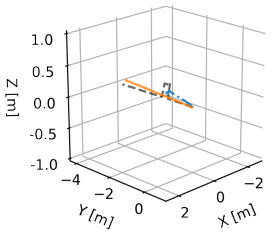
(b) FWD Base



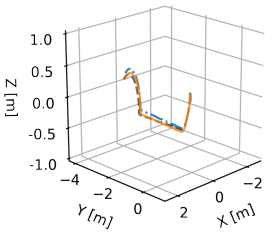
(c) FWD E EE



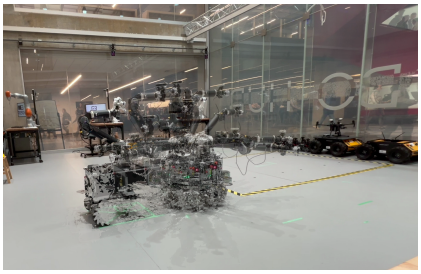
(d) RVR E Video



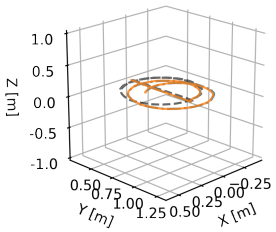
(e) RVR Base



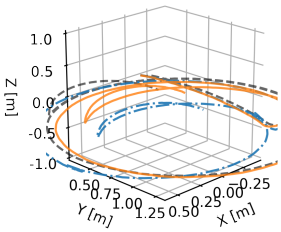
(f) RVR E EE



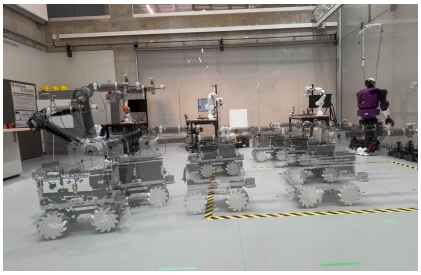
(g) SPI E Video



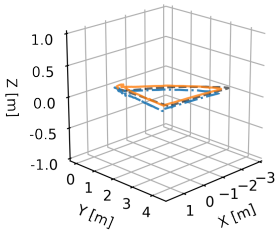
(h) SPI Base



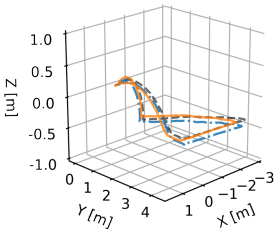
(i) SPI E EE



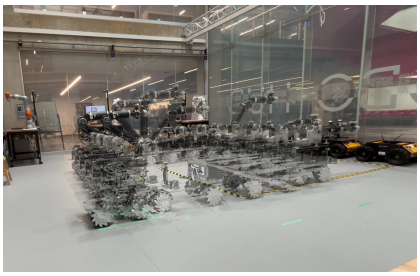
(j) TRI E Video



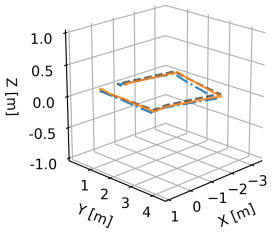
(k) TRI Base



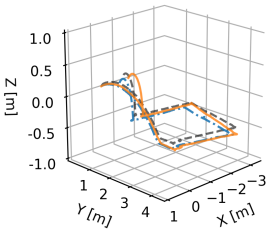
(l) TRI E EE



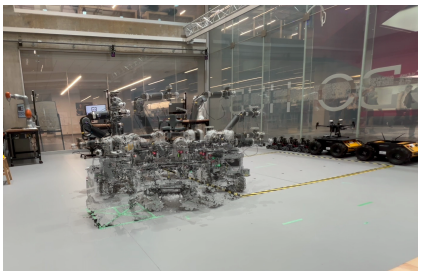
(m) SQR E Video



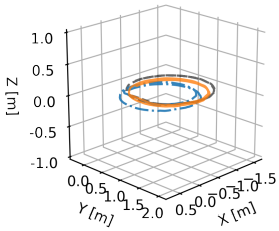
(n) SQR Base



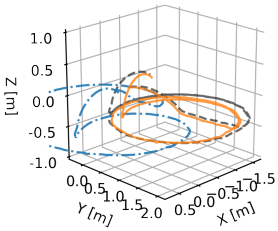
(o) SQR E EE



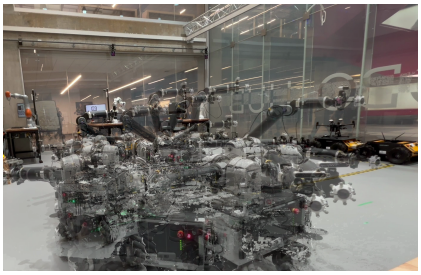
(p) CIR E Video



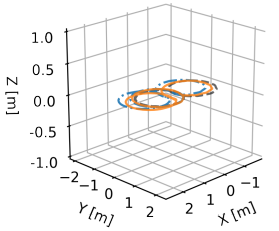
(q) CIR Base



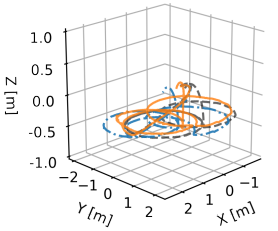
(r) CIR E EE



(s) BEE E Video



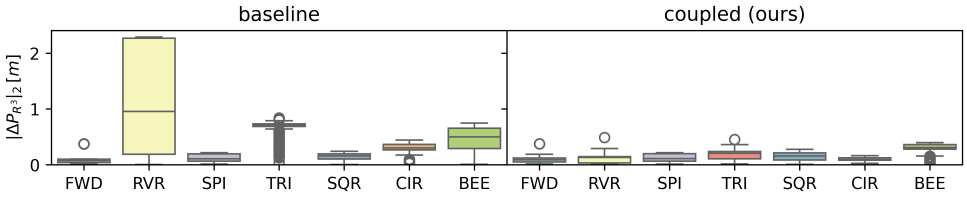
(t) BEE Base



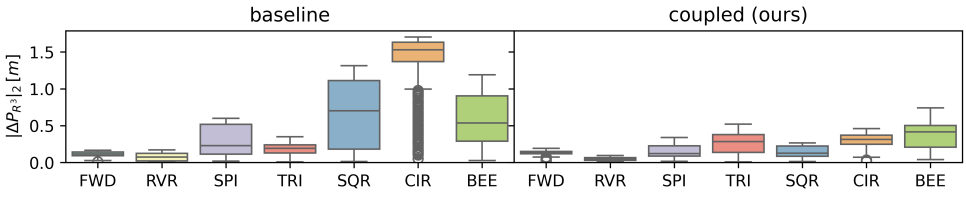
(u) BEE E EE

Figure 4.7: A time-lapsed image (a) and trajectory plots (b,c) for dual-VIO (without loop-closure) in Waggle-Bee-Dance (BEE) base motion with arm at Extended (E) position

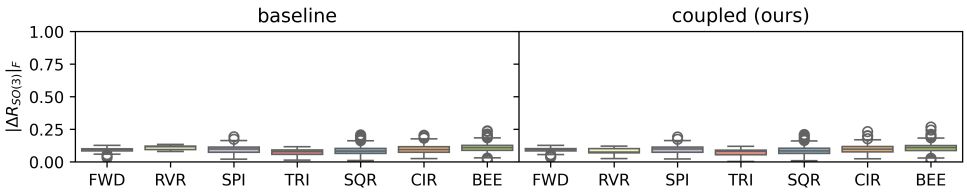
As illustrated in Figure 4.8 below, the performance improvements in EE are significant, while the baseline method suffers scaling issues at the EE VIO, resulting in larger trajectory errors.



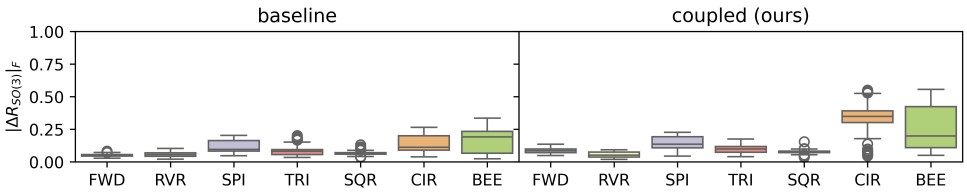
(a) Base translation metrics



(b) EE translation metrics



(c) Base orientation metrics

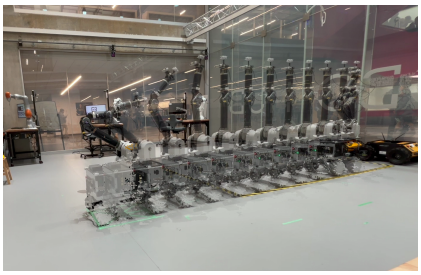


(d) EE orientation metrics

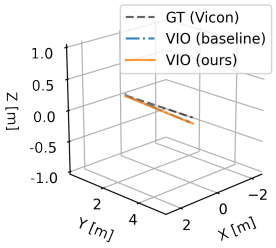
Figure 4.8: Boxplots for ATE errors for both base and EE VIOs (without loop closure) in terms of translation and orientation components at the Extended (E) configuration.

4.3.4 Demonstrations and Trajectories (Up (U))

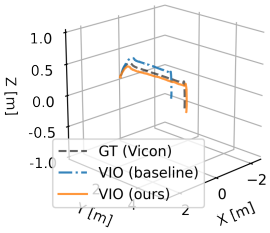
For the Up (U) arm position with seven different base motions, our proposed methods outperform the baseline methods in EE VIO particularly.



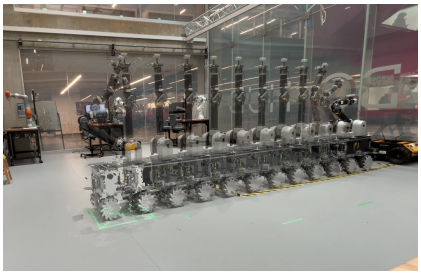
(a) FWD Up (U) Video



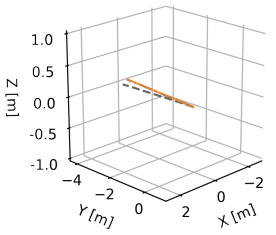
(b) FWD Base



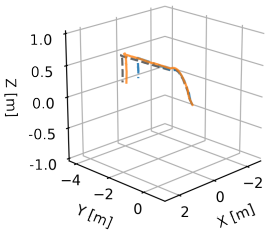
(c) FWD U EE



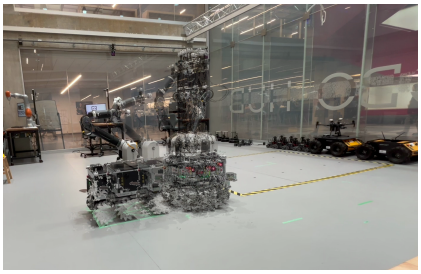
(d) RVR U Video



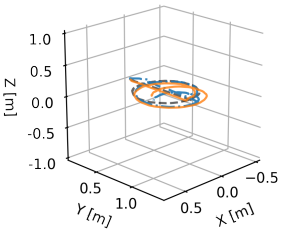
(e) RVR Base



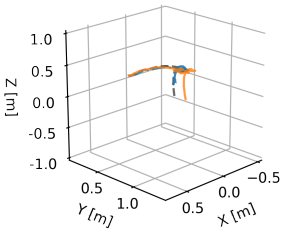
(f) RVR U EE



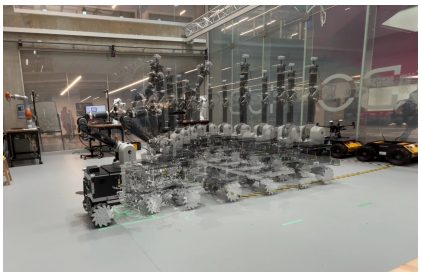
(g) SPI U Video



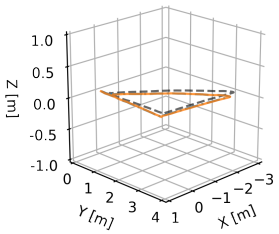
(h) SPI Base



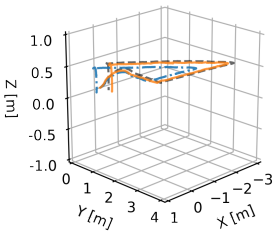
(i) SPI U EE



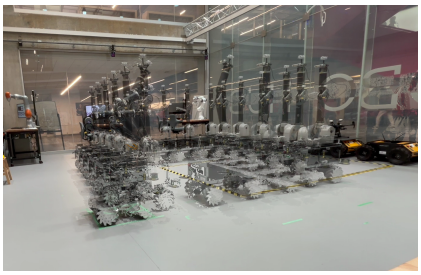
(j) TRI U Video



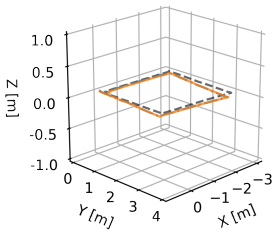
(k) TRI Base



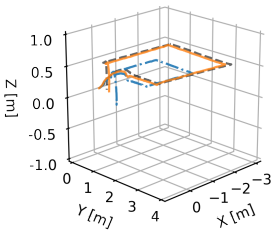
(l) TRI U EE



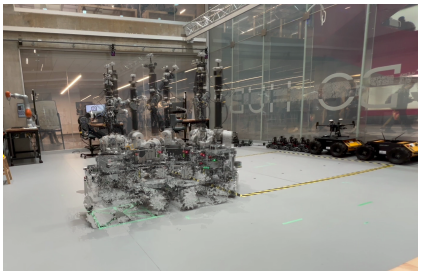
(m) SQR U Video



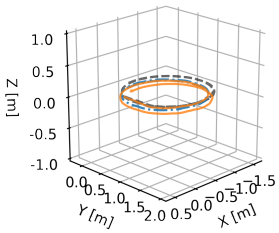
(n) SQR Base



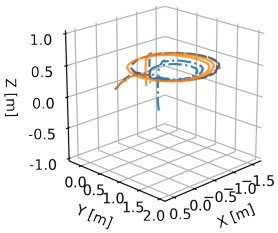
(o) SQR U EE



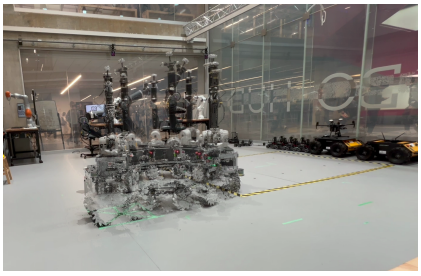
(p) CIR U Video



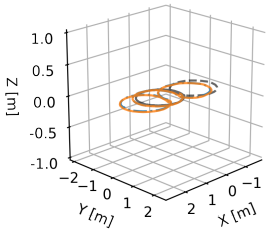
(q) CIR Base



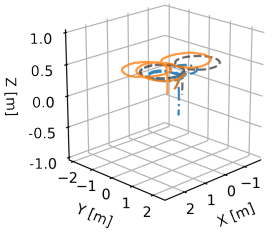
(r) CIR U EE



(s) BEE U Video



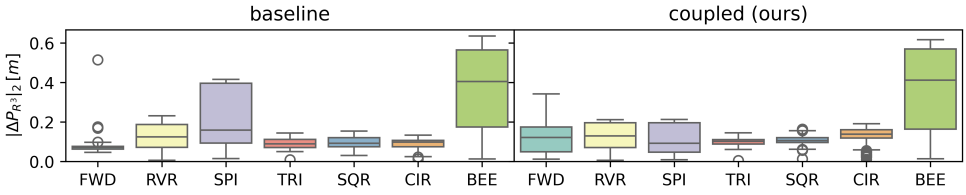
(t) BEE Base



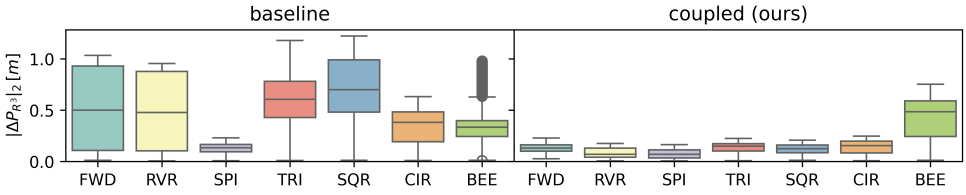
(u) BEE U EE

Figure 4.8: A time-lapsed image (a) and trajectory plots (b,c) for dual-VIO (without loop-closure) in Waggle-Bee-Dance (BEE) base motion with the arm at Up (U) position

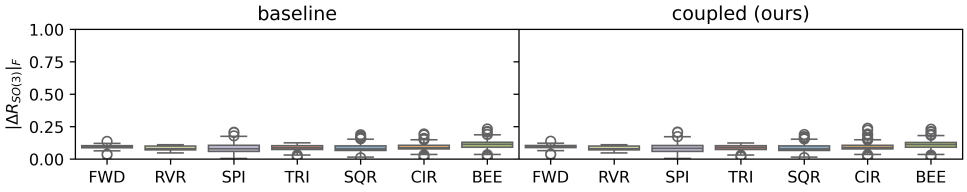
As illustrated in Figure 4.9 below, the performance improvements in EE are significant, while the baseline method suffers stability issues at the EE VIO.



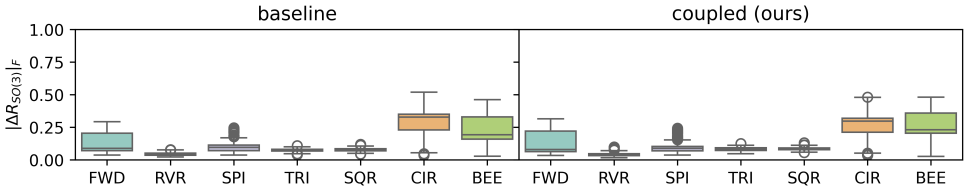
(a) Base translation metrics



(b) EE translation metrics



(c) Base orientation metrics



(d) EE orientation metrics

Figure 4.9: Boxplots for ATE errors for both base and EE VIOs (without loop closure) in terms of translation and orientation components at the Up (U) configuration.

4.3.5 Performance Discussion on Static Arm Motions

Overall, our performance in static arm motions also suggests that our proposed method significantly improves the accuracy and consistency of the VIO at the EE, particularly when the locomotion becomes more complex. This should come as no surprise; the EE suffers from significant vibrations and excitation when the mobile base crosses the uneven terrain due to the imperfect ground levelling in the testing facility and stiff omni-wheels. By imposing the base VIO onto the EE VIO pose graph optimization, the resultant pose estimation has been made substantially better, thereby producing robust odometry with minimal consistent ATE.

Meanwhile, the performance of the base odometry appears generally similar to that of the baseline; however, it showed more consistency throughout different base motions. This is because our methods can suppress significant odometry errors from the ground impacts or visual ambiguities through the regulation of the complementary VIO estimates and thus can prevent early divergence of the VIOs upon encountering uncertainties.

Chapter 5

Conclusion and Future Work

In conclusion, we have proposed and validated a new tightly coupled dual-VIO scheme for robust dual-state estimation of a mobile manipulator operating under dynamic locomotion. It moderates both VIO modules (the mobile base and the EE) through additional arm odometry factors via arm forward kinematics. This fusion enables the handling of degenerate modes through arbitration, thereby bringing the best out of each system: stable odometry from the base and dexterous FOV from the EE.

Our approach adopts jointly optimizing two VIO modules without maintaining a single large factor graph, allowing real-time capability for multi-VIO fusion for kinematically constrained systems with aggressive dynamic locomotion. This work can be essential for active-SLAM, where the EE VIO can focus on active exploration for mapping, while the base VIO concentrates on exploitation for re-localization as a part of the future works.

The current implementation has three major uncertainties in the arm: unobservable modes in modelling, inaccurate zeroing, and inaccurate eye-in-hand pose between the camera and the robot. These factors limit the accurate estimation of orientation and translation for basic motions. Therefore, there is a need for an online calibration to improve the performance further. As another future work, the proposed dual-VIO may be used in stereo mode to further improve the accuracy with the help of a high bandwidth communication channel. Furthermore, we plan to utilize mutual features between two cameras to support online temporal calibration on the fly, realizing fault tolerance and self-reinitialization capability.

References

[1] "Allan Variance ROS," https://github.com/ori-drs/allan_variance_ros, Dec. 2023.

[2] "Office Room from GrabCAD," https://grabcad.com/library/office-room-8.

[3] F. Dellaert, "Factor Graphs: Exploiting Structure in Robotics," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 4, no. 1, pp. 141–166, May 2021.

[4] "Ethz-asl/kalibr: The Kalibr visual-inertial calibration toolbox," https://github.com/ethz-asl/kalibr.

[5] "Introducing the Intel® RealSense Depth Camera D455," https://www.intelrealsense.com/depth-camera-d455/.

[6] J. Xu and S. Jeon, "A New Tightly-Coupled Dual-VIO for a Mobile Manipulator With Dynamic Locomotion," *IEEE/ASME Transactions on Mechatronics*, pp. 1–9, 2024.

[7] K. Jung, Y. Kim, H. Lim, and H. Myung, "ALVIO: Adaptive Line and Point Feature-based Visual Inertial Odometry for Robust Localization in Indoor Environments," 2020.

[8] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 1689–1696.

[9] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. Rome, Italy: IEEE, Apr. 2007, pp. 3565–3572.

[10] Gary Bradski, "The OpenCV Library," https://github.com/opencv/opencv, Jan. 2008.

[11] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[12] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust Visual Inertial Odometry Using a Direct EKF-Based Approach," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. ETH-Zürich, 2015.

[13] Y. Alkendi, L. Seneviratne, and Y. Zweiri, "State of the Art in Vision-Based Localization Techniques for Autonomous Navigation Systems," *IEEE Access*, vol. 9, pp. 76 847–76 874, 2021.

[14] M. Servières, V. Renaudin, A. Dupuis, and N. Antigny, "Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking," *Journal of Sensors*, vol. 2021, p. e2054828, Feb. 2021.

[15] G. Jia, X. Li, D. Zhang, W. Xu, H. Lv, Y. Shi, and M. Cai, "Visual-SLAM Classical Framework and Key Techniques: A Review," *Sensors*, vol. 22, no. 12, p. 4582, Jun. 2022.

[16] W. Chen, G. Shang, A. Ji, C. Zhou, X. Wang, C. Xu, Z. Li, and K. Hu, "An Overview on Visual SLAM: From Tradition to Semantic," *Remote Sensing*, vol. 14, no. 13, p. 3010, Jun. 2022.

[17] J. Jeon, S. Jung, E. Lee, D. Choi, and H. Myung, "Run Your Visual-Inertial Odometry on NVIDIA Jetson: Benchmark Tests on a Micro Aerial Vehicle," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5332–5339, Jul. 2021.

[18] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[19] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 2320–2327.

[20] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[21] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, Jun. 2008.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 2564–2571.

[23] R. Li, S. Wang, and D. Gu, "Ongoing Evolution of Visual SLAM from Geometry to Deep Learning: Challenges and Opportunities," *Cogn Comput*, vol. 10, no. 6, pp. 875–889, Dec. 2018.

[24] T. Qin, T. Chen, Y. Chen, and Q. Su, "AVP-SLAM: Semantic Visual Mapping and Localization for Autonomous Vehicles in the Parking Lot," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 5939–5945.

[25] H. Strasdat, J. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, Feb. 2012.

[26] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "PL-VIO: Tightly-Coupled Monocular Visual–Inertial Odometry Using Point and Line Features," *Sensors (Basel)*, vol. 18, no. 4, p. 1159, Apr. 2018.

[27] I. Asl Sabbaghian Hokmabadi, M. Ai, and N. El-Sheimy, "Shaped-Based Tightly Coupled IMU/Camera Object-Level SLAM," *Sensors*, vol. 23, no. 18, p. 7958, Jan. 2023.

[28] Z. Song, X. Zhang, T. Li, S. Zhang, Y. Wang, and J. Yuan, "IR-VIO: Illumination-Robust Visual-Inertial Odometry Based on Adaptive Weighting Algorithm With Two-Layer Confidence Maximization," *IEEE/ASME Trans. Mechatron.*, vol. 28, no. 4, pp. 1920–1929, Aug. 2023.

[29] F. Zheng, W. Lin, and L. Sun, "RL-VIO: A Robust and Lightweight Visual-inertial Odometry in Dynamic Environments," in *2023 42nd Chinese Control Conference (CCC)*. Tianjin, China: IEEE, Jul. 2023, pp. 4243–4248.

[30] B. P. W. Babu, D. Cyganski, J. Duckworth, and S. Kim, "Detection and Resolution of Motion Conflict in Visual Inertial Odometry," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, May 2018, pp. 996–1002.

[31] B. Yi, C. Jin, L. Wang, G. Shi, V. Ila, and I. R. Manchester, "PEBO-SLAM: Observer design for visual inertial SLAM with convergence guarantees," Jun. 2023.

[32] Y. Ning, "A Comprehensive Introduction of Visual-Inertial Navigation," Jun. 2023.

[33] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent Robotics and Autonomous Agents. Cambridge, Mass: MIT Press, 2005.

[34] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Initialization-Free Monocular Visual-Inertial State Estimation with Application to Autonomous MAVs," in *Experimental Robotics: The 14th International Symposium on Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, M. A. Hsieh, O. Khatib, and V. Kumar, Eds. Cham: Springer International Publishing, 2016, pp. 211–227.

[35] Agarwal, Sameer and Mierle, Keir and The Ceres Solver Team, "Ceres Solver," https://github.com/ceres-solver/ceres-solver, Oct. 2023.

[36] A. R. Memon, H. Wang, and A. Hussain, "Loop closure detection using supervised and unsupervised deep neural networks for monocular SLAM systems," *Robotics and Autonomous Systems*, vol. 126, p. 103470, Apr. 2020.

[37] Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M. R. Oswald, and M. Pollefeys, "NICE-SLAM: Neural Implicit Scalable Encoding for SLAM," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Apr. 2022, pp. 12 786–12 796.

[38] P.-Y. Lajoie, S. Hu, G. Beltrame, and L. Carlone, "Modeling Perceptual Aliasing in SLAM via Discrete–Continuous Graphical Models," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1232–1239, Apr. 2019.

[39] T. H. Nguyen, T.-M. Nguyen, and L. Xie, "Flexible and Resource-Efficient Multi-Robot Collaborative Visual-Inertial-Range Localization," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 928–935, Apr. 2022.

[40] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, Mar. 2015.

[41] D. Wu, X. Zhong, X. Peng, H. Hu, and Q. Liu, "Multimodal Information Fusion for High-Robustness and Low-Drift State Estimation of UGVs in Diverse Scenes," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–15, 2022.

[42] M. F. Ahmed, K. Masood, V. Fremont, and I. Fantoni, "Active SLAM: A Review on Last Decade," *Sensors*, vol. 23, no. 19, p. 8097, Jan. 2023.

[43] I. Lluvia, E. Lazkano, and A. Ansuategi, "Active Mapping and Robot Exploration: A Survey," *Sensors*, vol. 21, no. 7, p. 2445, Apr. 2021.

[44] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[45] F. Dellaert and M. Kaess, "Factor Graphs for Robot Perception," *FNT in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[46] M. Lourakis and A. Argyros, "Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment?" in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1.* Beijing, China: IEEE, 2005, pp. 1526–1531 Vol. 2.

[47] D. Wisth, M. Camurri, and M. Fallon, "Robust Legged Robot State Estimation Using Factor Graph Optimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4507–4514, Oct. 2019.

[48] D. Sharafutdinov, M. Griguletskii, P. Kopanev, M. Kurenkov, G. Ferrer, A. Burkov, A. Gonnochenko, and D. Tsetserukou, "Comparison of modern open-source Visual SLAM approaches," *J Intell Robot Syst*, vol. 107, no. 3, p. 43, Mar. 2023.

[49] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual–Inertial Odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[50] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, Jun. 2004.

[51] C. Mei and P. Rives, "Single View Point Omnidirectional Camera Calibration from Planar Grids," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Apr. 2007, pp. 3945–3950.

[52] L. W. Tu, *An Introduction to Manifolds*, ser. Universitext. New York, NY: Springer New York, 2011.

[53] J. M. Lee, *Introduction to Smooth Manifolds*, ser. Graduate Texts in Mathematics. New York, NY: Springer, 2012, vol. 218.

[54] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. CRC Press, 1994.

[55] S. Teng, D. Chen, W. Clark, and M. Ghaffari, "An Error-State Model Predictive Control on Connected Matrix Lie Groups for Legged Robot Control," Jan. 2023.

[56] J. Solà, J. Deray, and D. Atchuthan, "A micro Lie theory for state estimation in robotics," Dec. 2021.

[57] T. Lee, "Geometric tracking control of the attitude dynamics of a rigid body on SO(3)," in *Proceedings of the 2011 American Control Conference*, Jun. 2011, pp. 1200–1205.

[58] D. E. Chang, K. S. Phogat, and J. Choi, "Model Predictive Tracking Control for Invariant Systems on Matrix Lie Groups via Stable Embedding into Euclidean Spaces," *IEEE Trans. Automat. Contr.*, vol. 65, no. 7, pp. 3191–3198, Jul. 2020.

[59] F. Bullo and R. M. Murray, "Tracking for fully actuated mechanical systems: A geometric framework," *Automatica*, vol. 35, no. 1, pp. 17–34, Jan. 1999.

[60] ——, "Proportional Derivative (PD) Control on the Euclidean Group," California Institute of Technology, Report or Paper, Jan. 1995.

[61] "Multi-Camera configurations with the Intel® RealSense LiDAR Camera L515," https://dev.intelrealsense.com/docs/lidar-camera-l515-multi-camera-setup.

[62] D. Schubert, N. Demmel, L. von Stumberg, V. Usenko, and D. Cremers, "Rolling-Shutter Modelling for Direct Visual-Inertial Odometry," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Macau, China: IEEE, Nov. 2019, pp. 2462–2469.

[63] C. Pu, C. Yang, J. Pu, and R. B. Fisher, "A general mobile manipulator automation framework for flexible tasks in controlled environments," *Advanced Engineering Informatics*, vol. 57, p. 102062, Aug. 2023.

[64] W. Chen, X. Wang, S. Gao, G. Shang, C. Zhou, Z. Li, C. Xu, and K. Hu, "Overview of Multi-Robot Collaborative SLAM from the Perspective of Data Fusion," *Machines*, vol. 11, no. 6, Jun. 2023.

[65] K. He, P. Simini, W. P. Chan, D. Kulić, E. Croft, and A. Cosgun, "On-The-Go Robot-to-Human Handovers with a Mobile Manipulator," in *2022 31st IEEE International*

Conference on Robot and Human Interactive Communication (RO-MAN), Aug. 2022, pp. 729–734.

[66] T. Sandy, L. Stadelmann, S. Kerscher, and J. Buchli, "ConFusion: Sensor Fusion for Complex Robotic Systems Using Nonlinear Optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1093–1100, Apr. 2019.

[67] P. Mäkinen, P. Mustalahti, S. Launis, and J. Mattila, "Redundancy-Based Visual Tool Center Point Pose Estimation for Long-Reach Manipulators," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Jul. 2020, pp. 1387–1393.

[68] B. Ubezio, S. Sharma, G. Van Der Meer, and M. Taragna, "Kalman Filter Based Sensor Fusion for a Mobile Manipulator," in *Volume 5A: 43rd Mechanisms and Robotics Conference*. Anaheim, California, USA: American Society of Mechanical Engineers, Aug. 2019, p. V05AT07A043.

[69] M. Luo, E. Li, R. Guo, J. Liu, and Z. Liang, "End-Effector Pose Estimation in Complex Environments Using Complementary Enhancement and Adaptive Fusion of Multisensor," *Journal of Sensors*, vol. 2021, pp. 1–18, Apr. 2021.

[70] S. Aguilera-Marinovic, M. Torres-Torriti, and F. Auat-Cheein, "General Dynamic Model for Skid-Steer Mobile Manipulators With Wheel–Ground Interactions," *IEEE/ASME Trans. Mechatron.*, vol. 22, no. 1, pp. 433–444, Feb. 2017.

[71] H.-W. Chae, J.-H. Choi, and J.-B. Song, "Robust and Autonomous Stereo Visual-Inertial Navigation for Non-Holonomic Mobile Robots," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9613–9623, Sep. 2020.

[72] S. Cao, X. Lu, and S. Shen, "GVINS: Tightly Coupled GNSS–Visual–Inertial Fusion for Smooth and Consistent State Estimation," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2004–2021, Aug. 2022.

[73] Y. Yang, P. Geneva, X. Zuo, and G. Huang, "Online Self-Calibration for Visual-Inertial Navigation: Models, Analysis, and Degeneracy," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3479–3498, Oct. 2023.

[74] J. Vihonen, J. Honkakorpi, J. Tuominen, J. Mattila, and A. Visa, "Linear Accelerometers and Rate Gyros for Rotary Joint Angle Estimation of Heavy-Duty Mobile Manipulators Using Forward Kinematic Modeling," *IEEE/ASME Trans. Mechatron.*, vol. 21, no. 3, pp. 1765–1774, Jun. 2016.

[75] T. Zhang, Y. Tan, Z. Lei, and X. Chen, "Robust Visual Odometry On SE3: Design and Verification," *IEEE/ASME Trans. Mechatron.*, vol. 28, no. 4, pp. 1807–1816, Aug. 2023.

[76] C. Zhang, Z. Yang, H. Zhuo, L. Liao, X. Yang, T. Zhu, and G. Li, "A Lightweight and Drift-Free Fusion Strategy for Drone Autonomous and Safe Navigation," *Drones*, vol. 7, no. 1, p. 34, Jan. 2023.

[77] Q. Xu, Z. Gao, C. Yang, and J. Lv, "High-Accuracy Positioning in GNSS-Blocked Areas by Using the MSCKF-Based SF-RTK/IMU/Camera Tight Integration," *Remote Sensing*, vol. 15, no. 12, p. 3005, Jun. 2023.

[78] Y. Cao and G. Beltrame, "VIR-SLAM: Visual, inertial, and ranging SLAM for single and multi-robot systems," *Auton Robot*, vol. 45, no. 6, pp. 905–917, Sep. 2021.

[79] H. Li and J. Stueckler, "Online Calibration of a Single-Track Ground Vehicle Dynamics Model by Tight Fusion with Visual-Inertial Odometry," Sep. 2023.

[80] Z. Zhu, Y. Kaizu, K. Furuhashi, and K. Imou, "Visual-Inertial RGB-D SLAM With Encoders for a Differential Wheeled Robot," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5360–5371, Mar. 2022.

[81] S. Zhu, Z. Tang, M. Yang, E. Learned-Miller, and D. Kim, "Event Camera-based Visual Odometry for Dynamic Motion Tracking of a Legged Robot Using Adaptive Time Surface," May 2023.

[82] Y. Kim, B. Yu, E. M. Lee, J.-h. Kim, H.-w. Park, and H. Myung, "STEP: State Estimator for Legged Robots Using a Preintegrated Foot Velocity Factor," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4456–4463, Apr. 2022.

[83] J. Kang, H.-B. Kim, K. H. Choi, and K.-S. Kim, "External Force Estimation of Legged Robots via a Factor Graph Framework with a Disturbance Observer," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 12 120–12 126.

[84] M. G. Miiller, F. Steidle, M. J. Schuster, P. Lutz, M. Maier, S. Stoneman, T. Tomic, and W. Sturzl, "Robust Visual-Inertial State Estimation with Multiple Odometries and Efficient Mapping on an MAV with Ultra-Wide FOV Stereo Vision," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid: IEEE, Oct. 2018, pp. 3701–3708.

[85] L. Morra, A. Biondo, N. Poerio, and F. Lamberti, "MIXO: Mixture Of Experts-Based Visual Odometry for Multicamera Autonomous Systems," *IEEE Trans. Consumer Electron.*, vol. 69, no. 3, pp. 261–270, Aug. 2023.

[86] C. Wang, Z. Cao, J. Li, J. Yu, and S. Wang, "Hierarchical Distribution-based Tightly-Coupled LiDAR Inertial Odometry," *IEEE Trans. Intell. Veh.*, pp. 1–13, 2023.

[87] C. Zhang, Y. Liu, F. Wang, Y. Xia, and W. Zhang, "VINS-MKF: A Tightly-Coupled Multi-Keyframe Visual-Inertial Odometry for Accurate and Robust State Estimation," *Sensors*, vol. 18, no. 11, p. 4036, Nov. 2018.

[88] M. Abate, A. Schwartz, X. I. Wong, W. Luo, R. Littman, M. Klinger, L. Kuhnert, D. Blue, and L. Carlone, "Multi-Camera Visual-Inertial Simultaneous Localization and Mapping for Autonomous Valet Parking," Apr. 2023.

[89] L. Zhang, D. Wisth, M. Camurri, and M. Fallon, "Balancing the Budget: Feature Selection and Tracking for Multi-Camera Visual-Inertial Odometry," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1182–1189, Apr. 2022.

[90] A. Bani Younes and D. Mortari, "Derivation of All Attitude Error Governing Equations for Attitude Filtering and Control," *Sensors (Basel)*, vol. 19, no. 21, p. 4682, Oct. 2019.

[91] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden: IEEE, May 2016, pp. 4304–4311.

[92] Z. Zhang and D. Scaramuzza, "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid: IEEE, Oct. 2018, pp. 7244–7251.